Instructional design and development methods prescribe how specifications for advanced instructional products, such as training simulators, should be designed. In practice, however, the design process is disturbed by many 'pragmatic' factors, such as conflicting constraints, interference from management, personnel changes in design teams and technological progress leading to new possibilities. Prescriptive models recognise the iterative nature of the design process, but they do not prescribe when and how iteration should take place. The empirical studies in this thesis regard the design of training simulator specifications. They confirm that iteration is both unavoidable and difficult to deal with. Developing support for iteration requires insight into the different reasons for iteration and insight into when iteration is or is not desirable. The empirical studies show that novice instructional designers can be adequately supported by providing systematic design methods in combination with specific measures that help them deal with iteration.

Iteration in instructional design

Daniëlle M. L. Verstegen

Iteration in instructional design:

An empirical study on the specification of training simulators

Daniëlle M. L. Verstegen

IVLOS series

The IVLOS series is published by the IVLOS Institute of Education of Utrecht University (Instituut voor Lerarenopleiding, Onderwijsontwikkeling en Studievaardigheden). The purpose of this series is to disseminate the results of research that can contribute to the improvement of education.

Other publications in this series are:
J. W. F. van Tartwijk. Docentgedrag in beeld. De interpersoonlijke betekenis van non-verbaal gedrag van docenten in de klas.
W. Veen. Computer ondersteunde docenten. De rol van de docent bij invoering van computers in de lespraktijk.
W. F. Admiraal. Reacties van docenten op aandachteisende situaties in de klas.
I. F. Wijgh. Gespreksvaardigheid in de vreemde taal, begripsbepaling en toetsing.
M. Dolk. Onmiddellijk onderwijsgedrag. Over denken en handelen van leraren in onmiddellijke onderwijssituaties.
H. Oost. De kwaliteit van probleemstellingen in dissertaties. Een evaluatie van de wijze waarop vormtechnische aspecten van probleemstellingen worden uitgewerkt.
I. Huibregtse. Effecten en didactiek van tweetalig onderwijs in Nederland.
P. den Brok. Teaching and student outcomes: a study on teachers' thoughts and actions from an interpersonal and a learning activities perspective.
B. Koster. Lerarenopleiders onder de loep. De ontwikkeling van een beroepsprofiel voor lerarenopleiders en het effect van het kennisnemen daarvan op hun zelfdiagnose.
I. van den Berg. Peer assessment in universitair onderwijs. Een onderzoek naar bruikbare ontwerpen.
C. Driessen. Analyzing textbook tasks and the professional development of foreign language teachers.
D.M.L. Verstegen. Iteration in instructional design: An empirical study on the specification of training simulators.

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG
Verstegen, D. M. L.
Iteration in instructional design: An empirical study on the specification of training simulators.
Iteratie tijdens het ontwerpen van onderwijs: Een empirische studie van de specificatie van trainingssimulatoren.
Proefschrift Universiteit Utrecht. – Met literatuuropgave. – Met samenvatting in het Nederlands.
Key words: iterative design, iteration, instructional design process, support for iteration, specification of instructional products, user requirements for learning means, training simulators, novice designers.
Trefwoorden: iteratief ontwerpen, iteratie, onderwijsontwerpproces, ondersteuning voor iteratie, specificatie van leermiddelen, gebruikerseisen voor leermiddelen, trainingssimulatoren, onervaren ontwerpers.
ISBN: 9064642885
Print: Ponsen & Looijen BV
Lay-out: Daniëlle Verstegen
Cover design: Nieke Janssen/Koos Wolff

Iteration in instructional design:
An empirical study on the specification of training simulators

Iteratie tijdens het ontwerpen van onderwijs:
Een empirische studie van de specificatie van trainingssimulatoren

(met een samenvatting in het Nederlands)

Proefschrift ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de Rector Magnificus, Prof. dr. W.H. Gispen, ingevolge het besluit van het College voor Promoties in het openbaar te verdedigen op vrijdag 30 januari 2004 des ochtends te 10.30 uur

door

Daniëlle Maria Ludovica Verstegen

geboren op 27 mei 1968, te Nuth

Promotor: Prof. dr. A. Pilot, IVLOS, Universiteit Utrecht.

Copromotor: Dr. Y.F. Barnard, EURISCO International.

Het in dit proefschrift beschreven onderzoek werd uitgevoerd bij TNO Technische Menskunde in Soesterberg.

The research reported in this dissertation was carried out at TNO Human Factors, Soesterberg, the Netherlands.

Foreword

Writing a Ph.D. thesis is, just like designing specifications for instructional products, an iterative process. Most of the time I could handle this idea very well. I saw my research going into notes, my notes into reports, and my reports into drafts of the various chapters. I could see how my ideas became clearer and the text got better. There were also, of course, many moments that I was just fed up with the whole thing. Luckily, my supervisors and colleagues were around to cheer me up. And when that did not help, there was always the threat that I would have to take up my fair share of the housework if I gave up. Many people have helped me to get through all the necessary iterations, and I would like to mention some of them explicitly.

A part of the work for this thesis was done in the context of the European research project MASTER (EUCLID 11.1). My masters of those days -Johan Riemersma, John van Rooij, Yvonne Barnard, Karel van den Bosch and Jan Moraal- stood by the cradle of my career as a researcher. The remaining part of my research was funded within the background research program of TNO Human Factors. I am grateful that I was given this chance to further develop my research skills and to apply the newly developed knowledge in projects for the Royal Netherlands Army. I would like to thank all my colleagues for the interesting discussions, the encouragement and the more practical arrangements that have made it possible for me to do research and to report it in this thesis.

Working on a Ph.D. in a work setting, and not as a Ph.D. student, has two big advantages. One is that you get to choose your subject and the way that you execute your research yourself. In my case this has led to a Ph.D. thesis with an uncommon and not very fashionable subject: iteration in instructional design. Not an easy subject to do research on, as I have learned the hard way, but a subject that I still find both important and fascinating. The second big advantage is that I got to choose my own supervisors, Albert Pilot and Yvonne Barnard. I have never regretted my choice. Yvonne's first, and possibly largest, achievement was that she convinced me to believe in this undertaking and to give up my holidays and weekends for it. She has been my sounding board and sparring partner, from the other side of the desk in the first years and from the other side of Europe later on. Albert patiently waited until I chose to show up again, sometimes not for six months or so, but then we took up our discussions where we had left of. He always managed to make time for long discussions and timely words of encouragement. I have tortured both Albert and Yvonne with numerous drafts, especially during the last months. Their comments have definitely helped me to further improve with each iteration cycle. The same is true for the comments of Johan Riemersma, Arja Veerman, Cathy van de Laak, Liam Fitzpatrick, Ineke Buijs and Jur Keessen who all read drafts and helped me to shave of the last rough edges.

The studies that are reported could not have been executed without the help of Martin van Schaik, Stijn Steutels en Sietske Zagers. Martin accompanied me on trips to distant islands and was my solid rock during the sometimes chaotic preparations for empirical studies. Stijn implemented the prototype tools, and Sietske helped to organise the second study. For the organisation and execution of the first experimental study I got a lot of help from the university lecturers, Jos Jaspers and Jerry Andriessen. I would also like to thank the participants in all studies: students, colleagues and target users.

A special word of thanks to Koos Wolff for making most of the figures, to Nieke Janssen for making the design of the cover, to Walter van Dijk for the photographs and to Dianne van den

Hoek for her secretarial and emotional support. I would like to thank TriaM Europe and Arnold Dekker for their help during the implementation of the tool, and EURISCO International for lending me a quiet office to write the first versions of my theory chapter and the conclusions.

Finally, of course, a few words for Liam and Gerrit. This thesis would never have been finished without their patience and continuous encouragement, and Liam's knowledge of Word.

Daniëlle Verstegen

Table of contents

Table of Figures

Table of Tables

Chapter 1
Introduction

The general problem addressed in this thesis is: how can instructional designers be supported during the iterative design process in order to design effective instructional products? To answer this question I have done research in the area of the design of specifications for training simulators. In Chapter 1, I describe the characteristics of this research area, and define the research questions to be addressed in this thesis in more detail.

1.1 The focus of this thesis

The operation and maintenance of technically advanced systems, such as cars, aeroplanes, power plants, computers, and radar systems, are critical and complex tasks that are not easy to learn. A large amount of training is necessary to reach adequate performance. Sometimes operation and maintenance tasks are taught with the operational system under supervision, e.g., learner drivers practise in a real car in the normal traffic environment with a driving instructor. In many cases, however, other ways of learning these tasks are more effective and efficient.

Instructional products are made to help a learner acquire some knowledge or skills (Merrill, 2001). In learning processes the learner interacts with a number of instructional products and other human beings (teachers, coaches, other learners, etc.). Technological progress has enabled the development of advanced instructional products, such as Computer Based Training (CBT), E-learning packages, simulations and simulators, that are used on their own or in combination with more traditional instructional products, such as text books, assignments, or lesson plans for teachers in the classroom. Technically advanced instructional products offer many opportunities to improve learning, but they are also expensive. Within the limited budgets available, the optimal choice and efficient use of instructional products is important and this puts more pressure on the design of such products. There is a tendency to give much attention to technical aspects. This is understandable since advanced instructional products contain advanced technology that has to be carefully designed and constructed. The key question, however, is not what is technologically possible or most advanced, but what can optimise the learning process.

The design of courses and/or specifications for the required instructional products is often referred to as instructional design (see § 3.1 for more precise definitions), and people who have specialised in this task are called instructional designers. Instructional design is the process of deciding what methods of instruction are best for bringing about desired changes in the learner's knowledge and skills for a specific course content and a specific learner population (Reigeluth, 1983). Thus, ideally, the production of instructional products should be based on what learners need to learn (Analysis), and how they can learn it most effectively and efficiently (Design), as depicted in Figure 1.1. Depending on the model that is used, these phases are seen as sequential, cyclic or overlapping (see § 3.3). In practice, this process is disturbed by many 'practical' factors, such as conflicting constraints, interference from management, personnel changes and technological progress leading to new possibilities. Reacting to these disturbances can make the process chaotic, but not reacting to them will certainly lead to solutions that are out of date or too expensive, and to instructional products that do not fulfil the requirements or will not be accepted in the organisation. Therefore, instructional design is an iterative process.

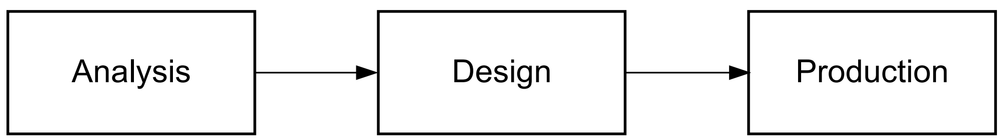


Figure 1.1: Simplified view on the production of instructional products.

The challenge for instructional designers is to design the best solution from an educational point of view, while at the same time taking other factors into account and reacting promptly to changing conditions. Instructional design and development methods recognise the iterative nature of the design process. They do not, however, explain when and how iteration should take place. This has led to the formulation of the general problem addressed in this thesis as:

How can instructional designers be supported during the iterative design process in order to design effective instructional products?

To answer this question I have done research in the area of the design of specifications for training simulators. In Section 1.2, I give a definition of training simulators and describe which kind of simulators have been the focus of my research. In Section 1.3, characteristics of training simulators are discussed. In Section 1.4, I define the research questions in more detail and, in Section 1.5, I give an overview of this thesis.

1.2 Training simulators

1.2.1 Definition of a training simulator

A simulator consists of a more or less realistic replication of an operational system and its environment, including the displays and controls that are available to the operators (Farmer, Jorna, Riemersma, van Rooij, and Moraal, 1999). More specifically, Gagné (1962) states that simulators:
- attempt to represent a real situation,
- provide certain controls over the situation, and
- deliberately omit certain aspects of the real operational situation.

A training simulator is a device that simulates certain aspects of the real system and/or its environment for the purpose of training (Riemersma et al., 1994). A training simulator is equipped with additional facilities to support training and instruction (Farmer et al., 1999).

These definitions include a broad range of devices ranging from almost exact copies of the operational system, e.g., a home trainer which is basically a bicycle with the wheels off the ground, to desk-top computer simulations without specific hardware, e.g., a flight simulator game where the user uses a keyboard and a joy-stick to control a simulated aircraft with simplified control panels and an outside view simulated on the screen. The area of my research was more limited and concerned simulators that are:
- designed for training, and not designed and used primarily for research, for designing operational systems or for leisure activities, and
- designed to practice tasks, and not to teach understanding of processes; i.e. not a simulation that allows learners to see how the electric current flows through an electric circuit, or a simulation of the effect of certain planning measures on the development of a town.

These training simulators fall under what is sometimes called 'procedural simulations' (Alessi and Trollip, 2001; Reigeluth and Schwartz, 1989): they are used to practice a sequence of actions to accomplish some goal. They are used in professional organisations, both military and civil, for task-oriented practice, for example to learn to drive a car or to fly an aircraft, to learn to operate a power plant, to use advanced communication devices, or to work in air traffic control (see also § 2.2.1 and § 2.3.1.1). In these settings the term trainees is used for the people who learn with a simulator (rather than students or learners), and the term instructors (rather than teachers) for those who conduct the training and support the learning process (see also § 1.5). Such 'procedural' simulators can still vary widely in complexity and the degree in which they simulate the real task environment: from a dome where trainee gunners use a simulated weapon to shoot with a laser beam at simplified projections of targets, to the simulation of an entire ship's bridge with all the real communication equipment or an aircraft cockpit with an advanced image system on a moving base. Some examples of training simulators are shown in Figure 1.2.

Figure 1.2: On the left: a flight simulator with 3D outside view.
On the right: a truck simulator on a moving base.
(Pictures W. van Dijk, Copyright 2003, TNO Human Factors, reprinted with permission).

1.2.2 Simulators as training devices

The effectiveness of a training simulator is determined by the training scenario, the underlying model (of the operational system and the task environment) and the instructional overlay (Reigeluth and Schwartz, 1989). The first two determine the degree of similarity between the practice situation and the real situation. The term fidelity refers to how closely the simulation imitates reality (Alessi and Trollip, 2001). Fidelity is usually the main cost driver: a small improvement can greatly increase the price of the simulator, for example when it requires a more advanced moving base or an expensive image system, but the returns in terms of learning and transfer can be disappointing (Moraal, 1985). It is not easy to determine when the extra investment will be worth it. Research has shown that more fidelity does not always improve training results (see, e.g., Alessi, 1988; Boer, 1991; Lintern and Garrison, 1992). To shed light on this issue researchers have distinguished different kinds of fidelity that influence the training value in different ways. Stanton (1996) makes a difference between the degree to which the simulator looks (physical fidelity) and acts (functional fidelity) like the operational system. Defined in this way, functional fidelity is vital for the transfer of the learned skills to the operational environment, whereas perfect physical fidelity is not required for those aspects that are not important for task execution. Ultimately, the value of a training device is determined by the degree of learning and transfer that occurs. Positive transfer will occur when the stimulus and response elements are the same in the simulated and real environments, i.e. when the trainees' behaviour in the simulated task is the same as the operational task under similar conditions (Korteling, van den Bosch, and van Emmerik, 1997).

Fidelity alone is not sufficient (Salas, Bowers, and Rhodenizer, 1998): in order to make simulators good training devices an instructional overlay, i.e. special facilities to support training and instruction is also required (Farmer et al., 1999; Flexman and Stark, 1987; Reigeluth and Schwartz, 1989; see also § 1.3.1).

1.3 Characteristics of training simulators

Simulators are used to substitute (parts of) training that would otherwise have to be done with the operational system. Each training session with the simulator consists of one or more exercises, usually called training scenarios in this context. In a course or curriculum, simulator training is usually combined with studying theory and practising with the operational system (called field training). Training sessions in simulators are preceded by, or alternated with, theory, either provided by the instructor in the classroom, or by self-study with books. Field training is done -for those aspects that cannot be covered with the simulator- in between simulator sessions, immediately after simulator training or 'on the job' with guidance from more experienced colleagues (see § 2.3.1.2).

1.3.1 Advantages of training simulators

Using simulators can be a solution when training with the operational system is impossible or too dangerous, e.g., to practise emergency procedures with aeroplane crews, or to practise how to deal with reactor problems in a nuclear power plant. Environmental restrictions can also make it impossible to practise some tasks in the real environment (see, e.g., Kennedy, 2001a). Practising with a simulator can also be cheaper. For pilot training, cost ratios ranging from 1:15 to more than 1:40 are mentioned (Strachan, 2000). An hour in a simulator for a P-3C Orion aircraft, for example, costs $119 compared to $2.903 per hour to fly the real thing (Kennedy, 2001b). The cost to operate a real tank is estimated at $75 per mile, compared to $2.50 per mile for a simulator (Kennedy, 2000). Another advantage is that the operational system remains available for normal operational use when simulators are used for training.

From an educational point of view, simulators have two other advantages. Firstly, they offer *full control over training content*: all trainees can practise the same tasks and part-tasks in all variants and appropriate (simulated) environmental conditions. Unlike, for example, driving lessons in a real car where trainees may not learn to drive in busy traffic or on snowy roads when those conditions do not occur during the lessons. Deviations from reality might even enhance training effectiveness: complex tasks can be practised in simplified environments, tasks can be broken up in parts, or task execution can be deliberately slowed down. On the other hand, the density of learning moments can be increased by practising difficult parts of the task more often, e.g., practising moving a ship out of and into a harbour many times, skipping the easier task of sailing at sea. The amount of training and the difficulty level of exercises can be controlled and increased gradually, and adapted to the performance of individual trainees. It is even possible to adapt the content of training to prepare trainees for specific tasks in the near future, such as a specific military mission.

Secondly, training simulators can incorporate an instructional overlay, i.e. a range of *facilities to support training and instruction* that are not available on the operational system, such as facilities to measure and register the trainees' performance, to provide demonstrations and (remedial) instruction, to provide augmented cues that speed up the learning process (and are gradually removed), to detect errors and provide guidance and feedback, and to adapt training to individual trainees. The simulator can take over parts of the instructor's task, e.g., provide some feedback automatically, and support the instructor in other tasks, e.g., by facilitating monitoring trainees and assessing their performance.

1.3.2 Disadvantages of training simulators

Training simulators are usually developed for highly specialised systems. They can often not be bought off-the-shelf, but have to be developed for one or a few customers. This means that customers will have to invest in carefully determining functional specifications, i.e. in specifying what the future simulator should be able to do. Even if they decide to buy an existing simulator, they will have to invest in a careful analysis of their own training needs to make sure that these can be covered with the simulator they intend to buy. In many cases, adaptations to the simulator are desirable because the entrance level of trainees or the division of tasks and responsibilities, and thus the training goals, differ from those that the off-the-shelf simulator was designed for. Those changes are often expensive or even impossible, because training simulators are specialised systems and often contain special hardware components. Making or buying a simulator is expensive. Thus, although training with simulators can in the long run be cheaper than training with the operational system, it requires a large investment -both in time and money- up front.

As described above, some aspects of the real operational situation are deliberately omitted in a simulator. This can have advantages for learning, but it also has risks for the transfer of training. Transfer is defined as: the extent in which trainees can apply the knowledge, skills and attitudes that they have learned during training in an effective way in their work setting (Veldhuis and Veerman, 2002). It reflects the change in the performance of tasks in the real task environment as a result of learning (Riemersma et al., 1994). Transfer can be quite good even with imperfect training environments. For example: instructors who train air defence gunners report several differences between their simulator and the real equipment: the hit rate is too high, the noise and smoke are absent during simulator training, etc. Still, more than 90% of trainees hit the target with their first shot on the real equipment (data from field study, see § 2.3.3.3). However, deviations that affect functional fidelity can have dramatic consequences. Van Breda and Boer (1988), for example, report a study where trainee tank drivers who had practised in a training simulator actually performed worse than trainees who had not practised at all due to problems with functional fidelity. For example: the behaviour of the vehicle (e.g., in curves), the visual system (e.g., the point of view) and the motion system (the movements that the trainee feels while driving) were not realistic.

1.3.3 Specific characteristics of training simulators

Some aspects are typical for training simulators, and not or much less so for other instructional products. The most important are:

A) Training simulators are technically complex systems
Simulators attempt to represent real, and often complex, situations and systems. Most of them contain specific hardware components that are also used in the operational system. Training scenarios should be based on realistic circumstances and representative of real-world conditions (Oser, Cannon-Bowers, Salas, and Dwyer, 1999). During the design phase these fidelity aspects take up a lot of attention. On the other hand, to achieve learning, structure and control over the training situation are required (Jeantheau, 1969; Oser et al., 1999).

B) Training simulators are expensive
Training simulators can be -and often are- expensive. In the field study prices between 0.5 and 35 Million Euro were reported (see § 2.3.1.3). The choice of instructional products for different parts of training is important, especially when expensive facilities like visual systems or moving bases are involved. On the other hand, the costs of training simulators are not large compared to the costs of the operational systems that trainees are trained to operate or maintain. Inadequate or incomplete training can lead to expensive damage and possibly even life-threatening situations.

C) Design of training simulators is a long, time-consuming and complex process
At the customers side different parties are involved in the design of specifications: instructors, managers, financial experts, etc. The personnel that is made responsible often has little or no expertise in this task (see also § 2.4.4 and § 3.2.1), and little knowledge about instructional design theory. Since simulators are usually technically complex systems, their production requires specialised personnel and is often contracted out to industry and not under the control of the customers. A prototyping approach is not feasible under these circumstances (see also § 3.3.5), which puts more pressure on the design phase.

D) Specific facilities
Two kinds of facilities are specific to training simulators: editing facilities for simulator training programmes and scenarios, and support for simulator instructors. These facilities play a large role in simulator training, and need to be specified beforehand since adding them later on is often expensive or technically impossible. For many other instructional products this is less true: a teacher can revise a classroom lesson without major costs, and even changes to CBT usually only involve software changes and no new hardware.

In the last decades the needs for simulator-based training changed and increased. Key factors were insufficient availability of both training grounds and budgets for training, the trend towards sub-contracting of training, and the rapid development of advanced technical systems demanding more complex and diverse training for complex tasks. There are more and more doubts whether the simulators in use brought enough value for money in training (Farmer at al, 1999). Van Rooij (1997) claims that the acquisition of training simulators is currently often a time-consuming, unstructured and technologically-oriented process that offers no guarantees for a valid, complete, consistent, and (cost) effective result. The solution for this problem is assumed to lie in designing better specifications, which made it an interesting domain for research concerning the general problem addressed in this thesis.

1.4 Research questions

In the context of training simulators the general problem addressed in this thesis can be further refined to:

How can the designers of training simulator specifications be supported during the iterative design process in order to design effective training simulators?

For my research I have used the MASTER method for the design of functional specifications for training simulators (Farmer et al., 1999, see also § 4.2). This method follows the Instructional System Design approach (see § 3.3.3) and claims that training simulator specifications, like specifications for other instructional products, should be based on an analysis of what is going to be trained with the simulator and how it is going to be trained (cf. Figure 1.1). The MASTER method consists of three main phases: Training Needs Analysis, Training Programme Design and Training Media Specification. For my research I have used the second phase, because in this phase the designers can exploit the potential advantages of simulators for training and instruction.

This thesis has two parts. The first part (Chapters 1 to 4) concerns the design of training simulators. More specifically, the research questions in this part are:

1. Which factors determine the (kind and level of) fidelity of training simulators?
2. Do training simulators have the necessary facilities to support training and instruction?
3. Which issues should be addressed during the design of functional specifications for training simulators from an educational point of view (product requirements)?
4. How are functional specifications for training simulators designed?
5. Which aspects of the design process need to be supported (process requirements)?
6. To what extent does the MASTER method fulfil the product requirements defined in Question 3 and the process requirements defined in Question 5?

The second part of this thesis (Chapters 5-10) focuses on the iterative aspect of the design process and on the usability of the design phase of the MASTER method: Training Programme Design (TPD), with the following research questions:

7. How can designers be supported during the iterative design of simulator training programmes?
8. How is iteration used in the design process?
9. Can novice designers design simulator training programmes with the MASTER-TPD method in an adequate way?
10. Do subjects use the different forms of support that are offered?

1.5 Overview of this thesis

The terminology in the field of instructional design has not yet been standardised. Different authors use different terms for the same concept or use the same terms in different ways. In this thesis I will use the terms training, trainees, and instructors in the context of simulator training. These terms may have the connotation of an instructor-led learning paradigm where trainees do not have much influence on their own learning process. This is, however, not intended here (although it is true that they are often used in this way, see Chapter 2). Thus, training goals can address required knowledge and attitudes, as well as procedures or skills that need to be practised. When I refer to learning processes in general or describe research conducted in different settings, I will use the more general terms learning, learner and teacher. Other terms will be defined where they are most relevant. Definitions of important terms can be found in the Glossary.

The research questions described in Section 1.4 are addressed as follows:
- research questions 1, 2 and 3 in Chapter 2,
- research questions 4 and 5 in Chapter 3,
- research question 6 in Chapter 4,
- research question 7 in Chapter 5,
- research questions 8 and 9 in Chapters 6, 7 and 9, and
- research question 10 in Chapter 8.

In Chapter 2, I look at how simulators are currently used for training by analysing data from 44 field visits to training simulators. Many interviewees are not satisfied with the fidelity of their simulator, although it is not always clear to what extent this limited fidelity affects learning and transfer. The extra opportunities for training and instruction that simulators can offer are not fully exploited, and the facilities to support instructors are often limited. To avoid such shortcomings in future, I define issues that should be addressed during the design of functional specifications for training simulators from an educational point of view (i.e. the product requirements).

In Chapter 3, I take a closer look at the process of designing functional specifications for training simulators. I compare the characteristics of this process with descriptive research about the process of instructional design in general and describe various types of prescriptive models that are available in the literature to support the instructional design process. Designing functional specifications for training simulators is a complex and inherently iterative process. I define which aspects of the design process need to be supported (process requirements) and conclude that none of the available models addresses all aspects.

In Chapter 4, I describe the MASTER method for the design of functional specifications for training simulators (Farmer et al., 1999). The MASTER method fulfils many of the requirements defined in Chapter 2 and 3, but it does not actively support iteration and it has not yet been used in a realistic setting.

In Chapter 5, I describe how the Training Program Design phase (TPD) of the MASTER method was implemented in an existing commercial package and how this prototype tool tries to support iterative design with process guidelines, hints, warnings and predefined design cycles. The prototype tool has been used in two studies to evaluate the usability of the MASTER-TPD method and the role of iteration in the design process. In both studies subjects used the MASTER-TPD method and prototype tool to design a draft training programme for a simulator.

In Chapter 6, I describe the design and most important results of the first evaluation study. This was an empirical study where subjects worked on their designs at home at their own pace over a three month period. I found no obvious correlation between the number of iterations and the quality of the resulting training programme designs. The time that subjects had dedicated to their design task had an important influence on the quality of the results.

In Chapter 7, I describe the results of a second study that was carried out in a controlled setting to in order to study iteration in more detail by introducing events that could trigger iteration during the design process.

In Chapter 8, I look into the results of both studies in more detail and discuss the usability of the MASTER method and the different forms of support that were available to subjects.

In many respects the situation during the two evaluation studies described in Chapters 6, 7 and 8 was similar to the situation that the MASTER method was developed for. There are, however, some differences that may have consequences for the validity of the results. Therefore, I briefly discuss the results of (less formal) studies with real target users in Chapter 9, and compare the results with those from the evaluation studies described before.

In Chapter 10, I summarise the answers to the research questions. I propose a list of events that can trigger iteration and measures to support an iterative design process. Finally, I discuss to what extent the results of my research can be generalised to other instructional design situations, and give some suggestions for further research.

Chapter 2
The use of training simulators in practice

In Chapter 2, I look at how simulators are currently used for training by analysing data of 44 field visits to training simulators. The research questions addressed in this chapter are:

1. Which factors determine the (kind and level of) fidelity of training simulators?
2. Do training simulators have the necessary facilities to support training and instruction?
3. Which issues should be addressed during the design of functional specifications for training simulators from an educational point of view (product requirements)?

2.1 Introduction

As discussed in Section 1.3.1, training with simulators is a solution when training with the operational system is not possible, too dangerous or too expensive. Full control over training conditions, the possibility to practice in abnormal, unusual and hazardous conditions and incorporation of specific instructional facilities can make simulators more effective for training than the operational systems themselves (Boer, 1991; Farmer, Jorna, Riemersma, van Rooij and Moraal, 1999; Flexman and Stark, 1987). In order to be good training devices simulators should adequately simulate the operational system and environments. They should also provide facilities to support training and instruction (see § 1.2.2), such as facilities to control and manipulate training conditions and to support the instructor or even to take over some of the instructor's tasks (see § 1.3.1).

Fidelity is usually the main cost driver: a small improvement can greatly increase the price of the simulator, for example when it requires a more advanced moving base or an expensive image system. How can we know beforehand what level and which kind of fidelity will be required for effective training? In a few domains research has been done that provides detailed information (see, e.g., Bone and Lintern, 1999; Kleiss, 1995; So, Chung, and Goonetilleke, 1999). In general, two aspects play a role: which tasks will be covered with the simulator, and which system features and functions are necessary for trainees to learn these tasks? Thus, from a theoretical viewpoint, the required level of fidelity depends on the training goals (Moraal, 1985). For those tasks and task aspects that should be covered with the training simulator, at least functional fidelity should be guaranteed. Physical fidelity is not required for those aspects that are not important for task execution (see § 1.2.2 for the definitions of physical and functional fidelity).

Limited functional fidelity restricts the range of tasks or task aspects that can be trained with the simulator. That means that trainees will not achieve all training goals at the end of simulator training. This is not necessarily a problem if these training goals can be, cheaper or more effectively, addressed elsewhere, by practising with another instructional product, during field training or 'on the job'. Several authors suggest that there is also a relationship between physical and/or functional fidelity and the place in the training pipeline (Alessi and Trollip, 2001; Boer, 1991; Reigeluth and Schwartz, 1989). A simplified, lower fidelity simulation may sometimes better serve beginning trainees. One reason is that high fidelity means higher complexity which taxes memory and other cognitive abilities. A second explanation is that proven instructional techniques which improve initial learning also tend to lower fidelity, e.g., practice on sub-tasks, providing help and adjusting the difficulty to the trainees' level (Alessi, 1988). For more experienced trainees, however, a high-fidelity simulation would be more suitable as it focuses on the transfer to the real task environment. Wager, Polkinghorne, and Powley (1992) propose a model for the selection of the required type of simulation based on the nature of training goal, the complexity of task and the experience of the trainee. Reigeluth and Schwartz (1989) also provide guidelines in this area.

In the context of simulator training the term training is used for practice and practice is defined as repeated attempts to perform a task, or more precisely, trainee activities executed in interaction with a training scenario intended to promote learning (Farmer et al., 1999). The term instruction is used in a broad sense for all activities executed to enhance training efficiency, including explanations, demonstrations, and feedback. Instruction can be provided by an instructor, fellow trainees or automatically by the training system.

In this chapter the results of a field study concerning 44 training simulators are presented, focusing on which factors determine the level of fidelity in practice and to what extent simulators provide facilities to support training and instruction. In Section 2.2 the method of the field study is described. Section 2.3 gives an overview of the most important results and in

Section 2.4 the answers to the research questions are discussed. The results of this field study are also reported in Verstegen, Barnard, and Pilot (2002). An earlier version covering only military training simulators appeared in Verstegen and Barnard (1998) and Verstegen, Barnard, and van Rooij (1999b).

2.2 Method

2.2.1 Field data

For the purpose of my research I performed a post-hoc analysis of data available from a field study of the MASTER project (Military Application of Simulation and Training concepts based on Empirical Research), in which I had participated. The MASTER project was executed within the European defence research programme EUCLID (European Co-operation for the Long term in Defence, RTP 11.1). TNO Human Factors was the main contractor of the MASTER consortium[1]. The MASTER field study consisted of visits to military training sites of the Army (12 cases), Navy (10 cases) and Air Force (22 cases) in the Netherlands, France, the United Kingdom and Spain. The simulators simulate vehicles, such as helicopters, fighter and transport aircraft, and tanks; communication and navigation systems, such as air traffic control, radar, and sonar; and weapon systems, such as air defence missiles and tank gunnery. The MASTER consortium kindly permitted the use of their raw data.

To make sure that the results would not only be valid for military settings, I also visited some simulators that are used in civil education were carried out. These training systems simulate vehicles: aeroplanes (1 case), ships (3 cases) and trucks (1 case), all in the Netherlands.

2.2.2 Methods of data collection

The different work packages within the MASTER project focused on different aspects of simulators. To cover all their interests the MASTER consortium (Riemersma et al., 1995) composed a 21-page questionnaire. It covers a wide range of topics, including training aspects, technical features, design aspects, and specific research interests. The questionnaire was used as a checklist to guide data collection. In most cases one or two members of the MASTER consortium visited the simulator training sites. These field visits lasted 0.5 to 1 day. Using the questionnaire, semi-structured interviews were conducted with personnel involved in simulator training, usually one or more simulator instructors or training staff. In some cases, the questionnaire was sent by mail and completed by the interviewees themselves. In total, about ten researchers were involved in the MASTER field orientation (including the author of this thesis). For each training simulator, results were summarised in reports of two to five pages which can be found in Carver, McGuinness, and van den Bosch (1996), and van Rooij et al. (1996). Closer inspection resulted in 39 reports of field visits with data concerning the research questions addressed in this chapter. Not all the reports are equally complete and detailed. For some issues information is not in all 39 cases available. Where possible, I have contacted consortium members for complementary information.

[1] The partners in the MASTER project were TNO Human Factors (main contractor), National Aerospace Laboratory (NLR), SAINSEL, Thomson-CSF, Laboratoire PRODEC (Université de Rouen), Dassault Aviation DGT/DSA, Centre d'Etude et de Recherches de Médecine Aerospatiale (CERMA), DGA/CELAR, Industrieanlagen Betriebsgesellschaft (IABG-ORA + IABG-WVO), Atlas Elektronik GmbH, CAE Electronics GmbH, DASA/Dornier Luftfahrt GmbH (EJ 30 +VFB 200), Eurocopter Deutschland, ESG Elektroniksystem und Logistik GmbH, TZN Forschungs- und Entwicklungszentrum Unterlüss GmbH, Defence Research Agency (DRA), British Aerospace (BAe), Link Miles, Hughes Rediffusion Simulation, Roke Manor Research Ltd., Marconi Simulation, Army Personnel Research Establishment and Logica.

For the additional five field visits to simulators used in civil education in the Netherlands the questionnaire was translated into Dutch and shortened, focusing on aspects relevant to my research questions. For the data collection, a similar procedure was followed: two TNO staff members (among which always the author of this thesis) visited the sites and conducted interviews with the training co-ordinator and/or simulator instructors. This resulted in data regarding 44 simulator training sites in total.

2.2.3 Research questions

In the context of this thesis, I was particularly interested in those aspects that make simulators good training devices. This led to the following research questions:

1. Which factors determine the (kind and level of) fidelity of training simulators?
2. Do simulators have the necessary facilities to support training and instruction?
3. Which issues should be addressed during the design of functional specifications for training simulators from an educational point of view?

2.3 Results

I will discuss the results grouped around 4 main topics:

§ 2.3.1 Training simulators
What kind of tasks is trained with simulators? What is the place of simulator training in the course? How were the simulators developed and/or purchased? What are the limitations of the present simulators?

§ 2.3.2 Training programmes
How is simulator training structured and organised? What is the structure of a training session? Who is responsible for the construction of training programmes and scenarios? How is feedback provided? Which performance data are registered and how?

§ 2.3.3 Trainees
How are trainees selected? How is the performance of trainees evaluated? Does simulator training prepare trainees adequately for their jobs (transfer/retention)?

§ 2.3.4 Instructors
How are instructors recruited and trained? What are the tasks of the instructor? What kind of instructor support does the simulator provide? What is the interviewees' opinion about the simulator?

Each section ends with a discussion regarding the section's topic. The research questions are addressed in the conclusions in Section 2.4. I will differentiate between simulators used in military and civil education only when the results show striking differences.

2.3.1 Training simulators

2.3.1.1 Tasks trained with simulators

The visited simulators are used to train operator tasks. They fall under what is sometimes called 'procedural simulations' (Alessi and Trollip, 2001; Reigeluth and Schwartz, 1989): they are used to teach a sequence of actions to accomplish some goal, such as driving a truck, flying an aircraft, sailing a ship, firing missiles, operating radar and/or sonar equipment, or operating systems for air traffic control. In some cases, tactical and team skills are included, but these are not the main focus of training. Depending on the task, simulators provide training for individual trainees, crews or both (see Table 2.1).

Table 2.1: Individual trainees or crew training

Type of trainees	Number of simulators (out of 44)
Only individual trainees	21
Mainly individual trainees, possibly also crews	2
Only crews	9
Mainly crews, possibly also individual trainees	5
Both crews and individual trainees	7

The simulators differ in complexity and level of physical fidelity: from a dome where trainee gunners use a simulated weapon to shoot with a laser beam at simplified projections of flying targets, to the simulation of an entire ship's bridge with all the real communication, radar and sonar equipment or a cockpit with an advanced image system and a moving base. They also vary in age: some have been in use since the mid seventies, others are relatively new; one was even still a prototype during the field study.

Almost all the simulators are primarily used for initial training of trainees who are new to the tasks to be learned, although two thirds of them are also used for refresher training and/or advanced courses (see Table 2.2). In two simulators only refresher training is executed; both are flight simulators that are used to keep up the required number of flight hours or refresh the skills of pilots who have not flown for a while. Simulators are used to train a wide variety of trainees from low educational levels up to trainees with academic degrees.

Table 2.2: Type of training

Type of training	Number of simulators (out of 44)
Mainly initial training, but also refresher courses	31
Only initial training	11
Only refresher courses	2

2.3.1.2 Place of simulator training in the course

For initial training, simulator sessions are combined with studying theory and practising with the real system (field training). In about half of the training programmes field training is an explicit part of the training programme, either alternated with or scheduled immediately after simulator training (see Table 2.3). In the other cases field training is part of other, advanced courses, or is the responsibility of the operational units that trainees are assigned to after completion of training.

Table 2.3: Simulator practice and practice on the real system

Simulator practice /practice on real system	Number of simulators (out of 44)
Simulator practice alternated with field training	6
Simulator practice followed by field training	14
Practice with real system not part of this training programme	17
No information about practice with real system	7

Simulators are used for the purpose of practice only. Prerequisite knowledge and skills are taught elsewhere, e.g., by instructors in the classroom or by studying books or syllabi (see Table 2.4). Theory lessons are given before, or alternated with, simulator practice. Three training programmes also include CBT. In one of the training programmes for gunners, for example, a CBT package is used to train the identification of targets before practising with the gunnery simulator. In four other cases the interviewees said that they would like to develop and use CBT in the future. During classroom lectures slides, cardboard replica's, or mock-ups may be used. Sometimes, the instructors use the simulator to provide demonstrations of correct or incorrect task behaviour. None of the simulators is equipped with other facilities to support learning of background theory.

Table 2.4: Learning prerequisite knowledge and skills

Prerequisite knowledge and skills	Number of simulators (out of 42[a])
Classroom lectures by instructor (can be combined with books/syllabi/CBT)	25
Only self-study	2
Sometimes theory lessons (depending on the entrance level of trainees)	5
Little or no theory study	3
No information available about theory study	7

[a] The two simulators that are only used for refresher training are not taken into account here.

2.3.1.3 Development and/or purchase

In most cases there is little or no information regarding the development and purchase of the training simulator (see Table 2.5). In about a third of the cases the price of the simulator is explicitly reported, ranging from about 0.5 Million Euro to 35 Million Euro. In most cases the original specifications and documentation regarding the way simulators have been developed or purchased are not available to the interviewees. There are some exceptions: in one military case and all but one of the five cases where simulators are used for civil education, the interviewees had been heavily involved in the design process themselves. In these cases, simulator specifications have been specified by a project team, consisting of instructors, domain experts and technical experts, including people from outside, for example, from universities or research institutes. In five other cases the interviewees knew that, before they started on the job, other instructors had been involved during the purchase of the simulator, but they had no further information or documentation.

Table 2.5: Purchase and/or development of simulators

Information about purchase/development	Number of simulators (out of 44)
Information available, interviewees were involved themselves	5[a]
Little or no information, but (other) instructors were involved	5
Interviewees have no information about purchase/development	19
No data regarding purchase/development in field study reports	15

[a] Four of the five are simulators used in civil education.

In three cases the interviewees claimed that the specifications had been based on the results of a training needs analysis, i.e. on training goals. Two of these were simulators used in regular civil school education. The instructors from these schools explained that they were used to systematic lesson design based on training goals for the rest of their curriculum and that they were obliged to do this in order to be recognised by and get funding from the government. In most cases, however, there are no indications that specific analysis or design models have been used to specify simulator specifications. Sometimes it is obvious that the acquisition had been entirely system-driven: a helicopter simulator was developed out of the pieces that remained after a crash; and several trainers were delivered together with the real weapon system without separate training requirements ever being written.

2.3.1.4 Reported limitations of the simulators

When asked for critical remarks or possible improvements of the simulator more than two thirds of the interviewees mention aspects related to fidelity (see Table 2.6). They were not satisfied with the image and/or motion system, the simulation model or the range of events or threats that could be simulated. It is often not clear whether these limitations affect learning and transfer or not. The consequences can be quite serious. Instructors of a military ship simulator claim that the fact that the simulator is not fully representative of the ship-borne equipment severely limits the effectiveness of training. In fact, they say, it is difficult for trainees to achieve the required level of skill at the end of training and it takes one to two years on the job to attain proficiency. In other cases, however, learning and transfer do not seem to be affected. In one of the helicopter simulators, for example, the instruments react too slowly on movements, but this does not hamper the learning process and, according to the instructors, only experienced pilots returning for refresher training are aware of the difference. One of the gunnery simulators has a black and white image system only. The instructors ask for a better image system, but at a different moment in the interview they also claim that coloured images are not required for the training of procedures, the purpose for which the simulator is presently used.

The fact that some events or environment characteristics cannot be simulated is often due to technical limitations. Often the system's rigid architecture, poor scenario editing facilities, and/or limited databases do not allow instructors to improve training programmes and to adapt to new developments in operational tasks. In one of the simulators for air defence gunners, for example, only targets with constant speeds can be simulated and this leads to unrealistic flight paths for the more modern targets. The instructors of a tank gunnery simulator complain about the fact that trainees in different cabins have to follow the same lesson and that the sequence of scenarios cannot be changed during a training session. The instructors of one of the helicopter simulators complain about the fact that some malfunctions are not included in the database. Other interviewees report limited facilities to make and/or adapt training scenarios (13 cases), insufficient support for the instructor on the Instructor Operator Station (IOS) (15 cases) and insufficient facilities for data registration (16 cases), amongst others (see Table 2.6).

Table 2.6: Reported limitations of the training simulators[a]

Reported limitations	Subtotal	Number of simulators (out of 44)
Fidelity problems		
- Limitations of vision and/or motion system	21	34
- Limitations of simulation model	7	
- Some events/threats cannot be simulated	16	
Facilities to construct and/or adapt scenarios and programmes		
- Available facilities not good or difficult to use	10	13
- No facilities available	1	
- Databases incomplete	5	
Registration and storage of performance data		
- Available facilities are not good	10	16
- No facilities available	6	
Facilities for the instructor on the IOS		
- Facilities for monitoring absent or not good	9	15
- IOS difficult to use (ergonomics)	4	
- Facilities to adapt scenarios or sequence during training absent or not good	2	
- Unspecified complaints	4	
Workload of instructors too high (during scenario execution)		6
Instruction/training for instructors should be improved		5
Technical limitations		4
Limited training capacity		2
Simulator does not simulate other actors in the scenario		1
No documentation available		1
No printing facilities		1

[a] Most interviewees mentioned several limitations of their simulator.

Many interviewees indicate that the range of tasks that can be covered is limited by the simulator's capacities. For example: low altitude flying cannot be practised in a simulator without an advanced image system; and practising dog fighting is not possible without a 360 degrees field of view. It is not clear, though, whether these tasks were initially supposed to be trained with the simulator or not. In one case interviewees even indicated that the limited fidelity had an advantage for learning: in a flight simulator without motion system the trainee pilots do not get the feedback from the movements of the plane. Thus, they are forced to pay close attention to their instruments, and this is, according to the instructors, a valuable training objective in itself.

The interviewees in civil education have fewer complaints. They seem to have a different view on limitations to fidelity: all of them mention differences between the simulators and the real systems that, in their opinion, do not affect training and do not have to be improved. None of the ship simulators, for example, has a moving base. Moving the projected 'out of the window' view brings along the illusion of ship movement, and this is good enough for training purposes. The simulation model of the flight simulator is also not perfect; but the school has started a research project to find out whether improvements would benefit training, before spending any money on it.

2.3.1.5 Discussion

The visited simulators are used to practice a wide variety of mainly procedural tasks. The opportunities to incorporate facilities for studying theory are hardly exploited. Facilities for demonstration, pre-programmed explanations or entire CBT modules could be used not only to learn prerequisite knowledge and skills beforehand, but also to provide individualised instruction on a just-in-time basis during simulator training. The facilities to support instructors to implement and deliver training often seem to be poor. Many interviewees are not satisfied with the facilities to construct training scenarios and programmes and limited facilities to support instructors during training (see also § 2.3.4.3) and the facilities for data registration and storage (see also § 2.3.2.5).

The lack of information about the specification and purchase of training simulators can mean two things: the specification process was not documented or the documentation got lost (which is not unlikely given the high personnel turnover in military settings). It is possible that a thorough analysis of training needs has been done but the results got lost, but it is equally possible that specifications have been written on an ad-hoc basis or not at all. The latter would also explain the lack of integration between simulator training and other forms of (computer-based) instruction. In only three cases there was evidence of the use of systematic analysis and design models to specify the simulator specifications. Two of these were simulators used in regular civil school education, where instructors have experience with systematic methods and are obliged to design their courses in a systematic way.

The simulators' fidelity is clearly a big issue: the vast majority of interviewees are critical about differences between the simulator and the real system, although it is less clear whether or how these differences affect learning and transfer. Because the original specifications are not available it is difficult to judge whether the limited training coverage of many simulators has been an intentional decision or not. For reasons of cost-effectiveness the designers may have decided not to include expensive features and to cover some training goals during training with the real system or on the job. It is equally possible, though, that field training has been designed later, only to amend the simulators' training limitations or, even worse still, that some tasks are not practised at all. The fact that almost all the interviewees responsible for civil education had personally been involved in specifying the training simulators may explain why they had fewer complaints.

2.3.2 Training programmes

2.3.2.1 Structure of training programmes

The reports of the field visits do not give detailed information about the structure and content of the training programmes. Trainees work through a number of exercises, usually called training scenarios in this context, that address a sequence of (part)tasks and gradually increase in difficulty level. In some cases the whole programme is pre-specified: trainees go through a standard set of scenarios in a fixed order. In other cases the instructor picks scenarios from the database during the training sessions. In two cases no scenarios are used (see Table 2.8). In a tank simulator, for example, the instructor chooses a terrain database and tells the trainee which route to take on the fly.

Sometimes the amount of practice is individualised: trainees continue practising until they have mastered a certain task, before they are allowed to go to the next topic (cf. mastery learning, see, e.g., Block, 1971; Patrick, 1992). Often, however, all trainees get the same amount of practice because the duration of a course and the time available for simulator training are fixed. Some instructors try to help weaker trainees by allowing them more practice time at the cost of other trainees, or by allowing them to practice during lunch breaks or in their own time. However, the most common and usually the only form of individualisation is the ad-hoc adaptation of scenarios by instructors, either just before or during training (see Table 2.7). Instructors can adapt training by setting parameters, changing weather conditions, slowing down or speeding up the scenario, introducing or removing events, adding or removing targets or repositioning them in place or time, etc.

Table 2.7: Adaptation of training scenarios

Opportunities to adapt scenarios	possible	not possible	no information
Adapt scenarios before training	30	3	11
Adapt scenarios during training	29	5	10

The content of refresher training can be less structured. Sometimes, it is determined by the trainees themselves or by their commanders. Experienced pilots, for example, can ask to train specific procedures or situations that they have difficulty with; and in a simulator for tactical decision making, staff teams prepare and execute their own exercises. In other cases, more structured refresher training programmes are used.

2.3.2.2 Structure of training sessions

The duration of training sessions is often predetermined, e.g., one lesson hour. Training sessions typically start with a briefing by the instructor to activate prior knowledge and prepare for training activities (Farmer et al., 1999). During the briefing the instructor explains the goal of the training scenarios and -if necessary- provides additional instruction. After the briefing trainees work through one or more training scenarios. Scenarios can be as short as a couple of minutes for simple procedures. In simulators for command and control procedures, where the timing of events is closer to the real world tempo, sessions usually contain one longer scenario. In one case such a session could even last several days. The session usually ends with a debriefing to consolidate learning and to evaluate the session in terms of training goals (Farmer et al., 1999). In six cases, no real instructors are present. In these cases the simulator training personnel is only responsible for scenario management, and not for any instructional aspects, such as briefing and debriefing.

2.3.2.3 Construction of training programmes and scenarios

Usually, a set of training scenarios is defined and implemented before the training programme starts (see Table 2.8). In some cases, a basic set of scenarios has been developed by the manufacturer and delivered with the training simulator. Otherwise, scenarios have been developed by instructors. Scenarios are often linked to training goals, although it is not always clear whether they have actually been developed on the basis of these training goals.

Table 2.8: Definition and implementation of scenarios

Ways to make new training scenarios	Subtotal	Number of simulators (out of 44)
Scenarios defined and implemented beforehand		
- New scenarios can be implemented and added to database	29[a]	41
- Unknown whether new scenarios can be added to database	12	
Scenarios constructed just before exercise starts		1
No scenarios used		2

[a] In 11 of these 29 cases interviewees are not satisfied with the available facilities.

Extending and revising the set of scenarios is usually possible (see Table 2.8). It is the responsibility of the instructors, a higher officer or a committee responsible for the content of training. However, in more than a third of the cases where new scenarios can be implemented, the interviewees are not satisfied with the editing facilities and/or the available databases. In two cases available editing facilities are never used.

2.3.2.4 Feedback

During training sessions trainees get intrinsic feedback on their performance from the behaviour of the training system. for example, the image of the outside world changes when a steering movement is made, or a target disappears after it has been hit. Due to differences between the simulator and the real system this intrinsic feedback is not always complete or correct, e.g., motion cues are lacking in flight simulators without motion system, and visual information is less detailed when the image system does not have a high resolution. Additional, extrinsic feedback is provided in different ways (see Table 2.9):

A) Automatic feedback during exercise
There are only three examples of automatic feedback during exercises. One of the gunnery simulators provides feedback when the trainee does not follow the correct procedure; and two civil ship simulators warn of calamities, e.g., a sound signal when there is collision danger or when the ship is about to hit the bottom. In three other cases the interviewees expressed a wish for automatic feedback on errors.

B) Verbal feedback by instructor during exercise
In most cases, the instructors provide feedback during the exercises, directly or through headphones. Feedback is provided immediately after errors or at the end of a (sub)task or scenario phase. Often the training scenario can be paused to provide feedback.

C) Automatic feedback/scores after exercises
There are only four cases of automatic feedback provided after exercises. In a truck simulator performance criteria have been specified for each scenario, e.g., concerning the position on the road and gear changing. Scores on these criteria are provided after the scenario has been completed. Similar automatic scoring on performance criteria was available in a gunnery trainer and a tank simulator. The fourth case concerns a simulator for firing air defence missiles that provides feedback about the flight path after firing, but this facility is rarely used in practice because the feedback is not reliable.

D) Debriefing at the end of the training session
Debriefings are very common, but vary widely in length and method. They can be quite short when extensive feedback was provided during the exercises, sometimes not more than giving a score. They can also be more elaborate, including replay of (parts of) the scenario, group discussions, videos, demonstrations and remedial instruction. Debriefings are usually provided by instructors. During refresher training debriefing they can be the responsibility of the team commander, especially when existing teams practice together.

Table 2.9: Extrinsic feedback

Ways to provide feedback	yes	no	no information
Automatic feedback during exercise	3	41	-
Verbal feedback by instructor during exercise	32	1[a]	11
Automatic feedback/score after exercise	4	40	-
Verbal debriefing after training session	36	1[a]	7

[a] One gunnery simulator provides automatic feedback when the procedure is not folowed correctly; as this simulator is used as procedure trainer only, this is the only form of feedback. This is the only case without verbal feedback.

2.3.2.5 Registration of performance data

Many simulators have some facilities to register and/or print training data, although these can be quite limited, e.g., only an overview of the flown trajectory in a flight simulator, or a log of the system's control settings. Printing facilities are not always available (see Table 2.10). Registered performance data are used by the instructors to support performance evaluation, i.e. for the purpose of feedback and assessment. Replaying (parts of) scenarios is used during the debriefing to illustrate correct and/or incorrect performance. Facilities for long term storage are only available in one simulator used for civil education. In three other civil schools performance sheets are filled in by the instructors and stored manually.

Table 2.10: Facilities to register/store performance data

Type of facilities	yes	no	no information
Facilities for data registration (any kind)	33	5	6
Store and replay (part of) scenarios	21	15	8
Printing facilities	16	8	20
Storing automatically provided scores/hit rates	4	34	6
Facilities for long term storage	1	37	6

2.3.2.6 Discussion

The information about the structure of training programmes is not very detailed, but in general it seems to be quite straightforward: trainees work through a number of scenarios which address a sequence of (part)tasks, and increase gradually in difficulty level. There is no evidence of elaborate training needs analyses or more complex training schemes. The possibilities to adapt training to the trainees' performance are limited: sometimes the amount of practice is adapted, but in most cases the only form of individualisation is the ad-hoc adaptation of scenarios. This has, however, two important disadvantages: it takes up a lot of the instructors' time and attention, and it makes the content of training untraceable. Trainees get different exercises, depending on what the instructor changes. This means that, especially when training time is fixed, there is no guarantee that all training goals are covered. It is possible that weaker trainees spend more time on easier tasks and never encounter some of the more complex situations. Moreover, which tasks and situations different trainees have encountered is not retraceable after training.

Some claim that adapting scenarios during training is necessary to adapt to the trainees' performance (see, e.g., Oser, Cannon-Bowers., Salas, and Dwyer, 1999). However, if a large set of scenarios is easily available, it is possible to adapt training to the needs of individual trainees by inserting, replacing or removing standard training scenarios, rather than changing scenarios on the spot. This would lighten the workload of instructors during training considerably. Perhaps more importantly, it would make the actual content of training retraceable. Combined with good assessment and extra training opportunities for weaker trainees, full coverage of all training goals can be guaranteed. Of course the development of a sufficiently large set of scenarios requires time, effort and good editing facilities. The instructional support system for training simulators developed by Kuiper (1995) is an example of a system that can support this aspect of simulator training. It provides facilities to create modules consisting of training scenarios with accompanying assessment criteria. During training sessions the system keeps track of the trainees' progress, presents scenarios based on their performance and provides relevant information to the instructor.

Individualised feedback and (remedial) explanations are other possible ways to individualise training. At the moment all this is the task of instructors, as only a few simulators provide feedback automatically. This means that the amount and quality of feedback depends very much on the instructors' expertise and experience. And the instructors' time for individual trainees is limited, especially when they are responsible for more than one trainee, or have other tasks during training, such as managing and adapting the scenario (see also § 2.3.4.2). Given the fact that simulators are mostly used to practice tasks of a procedural nature (see § 2.3.1.1), the lack of facilities to support feedback is remarkable. Feedback plays an important role, because it allows trainees to learn from practice, to correct errors and optimise their performance. Frequently occurring errors in procedural tasks are often easily detectable or measurable. In these cases, the simulators could provide feedback automatically. But even when this is not possible, the instructor could be supported with, for example, data about the trainees' performance in the past, standard explanations for often occurring errors, or system warnings when trainees are not performing at the expected level or show uncommon behaviour. Facilities to automatically register performance data, and to record and replay (parts) of the scenario, can support the debriefing with trainees afterwards. Not registering and storing performance data also means that trainee performance and the simulator itself cannot be evaluated over a longer period of time (see also § 2.3.3.4).

2.3.3 Trainees

2.3.3.1 Selection of trainees

In only one case does the selection of trainees include tests and skill aptitude judgements. In all other cases the mastery of prerequisite knowledge and skills is not explicitly tested, but is assumed on the basis of previous training or work experience. Entry requirements concern educational level, completion of other courses, rank, field experience or physical aspects. For example, to apply for the position of sonar operator, trainees need to have a high school diploma and unimpaired hearing and eyesight. The trainees for a civil truck simulator come from a vocational training school; they have a car driver's license and they have passed their theory exam for truck driving before they start practising with the simulator. When simulators are used for team training, existing or future teams are sometimes selected to train together.

2.3.3.2 Evaluation of performance

Most courses include some form of assessment at the end of simulator practice. The final score is often a weighed average of this assessment and the results of a (paper-based) test for the theoretical part of the course. Dropout of trainees during the course is not systematically recorded. It seems to be low: percentages of 5% or less are mentioned.

In four cases there is no assessment of simulator practice. In two vehicle simulators, a tank simulator and a truck simulator, the trainees practice driving for a fixed amount of time and then continue in the real vehicle. Assessment takes place at the end of this field training. In a simulator used to train command and control procedures, the instructor evaluates with trainees during debriefings, but does not formally assess their performance. The same is true for a ship simulator where training is provided for personnel of commercial companies (Note that if companies ask for assessment, they are allowed to attend training sessions and assess the trainees' performance themselves or to hire independent assessors).

Table 2.11: Types of assessment

Type of assessment	Number of simulators (out of 44)
(Partly) automated	3
Structured assessment by instructor	12
Unstructured assessment by instructor	15
Self-assessment by trainees	3
No information about type of assessment	6
Unknown whether assessment takes place	1
No assessment	4

The following types of assessment of simulator practice were encountered (see Table 2.11):

A) (Partly) automated assessment
Two gunnery simulators keep track of the actions of the trainee and provide a score based on whether the trainees followed the correct procedure. A third one has a more complicated scoring system, based on performance speed and accuracy, hit rate, following the correct procedure and answers to queries such as target prioritisation.

B) Structured assessment by instructor
Checklists or standard evaluation sheets are used in 12 cases to assess the performance of trainees. Sometimes, the final score is a weighed average of the instructors' scores on different aspects. In other cases, a checklist is used as a reminder of important task aspects, but the instructor is free to determine the final score. In one of the gunnery simulators trainees have to attain a minimum hit rate as well.

C) Unstructured assessment by instructor
In 15 cases the instructor gives a subjective impression of the trainees' performance during practice or during special test scenarios.

D) Self-assessment
In three command and control trainers the teams assess their own performance during the debriefing: a group discussion takes place, often led by the team's own commander.

The performance criteria used for assessment are often subjective, unclear or even absent. There are only four examples of measurable criteria: the three gunnery simulators with automated assessment, and a minimum hit rate for a fourth gunnery simulator. In six other cases an evaluation form with clearly defined criteria is available; three of these are simulators used for civil education (Note that in the other two civil schools there is no assessment of simulator practice, see above). The evaluation sheets contain performance criteria for the training goals to be attained; and there is a standard procedure to determine the final score based on the scores on these criteria. In two cases external examiners are involved.

2.3.3.3 Retention and transfer

Generally, information about the retention and the transfer of knowledge and skills is not available or only of anecdotal nature. Many instructors claim that procedural skills are forgotten quickly if they are not used regularly. Retention spans of a couple of weeks or months are mentioned. This is in accordance with research findings repeated in the literature review of McGuinness, van den Bosch, and Verstegen (1996).

Many interviewees report that feedback from the field staff has given them the impression that the transfer of the learned knowledge and skills to the real task environment is good. Systematic research into these issues is very rare. Quantitative information is only in a few cases available. For one of the gunnery trainers the hit rate of the first shot on the real equipment is more than 90%. For one of the tank driving simulators a research study showed that trainees who have trained with the simulator need less training during field training. For one of the ship simulators used in civil education a research study showed that 5 days simulator practice is equivalent to 15 days training on board due to the higher density of learning moments in the controlled simulator environment (Schraagen, van Breda and Schaafstal, 1994).

2.3.3.4 Discussion

The effectiveness of training can be expressed in terms of learning gain at the end of training: how much better are trainees able to perform their tasks as a result of the training programme? It can also be expressed in terms of transfer of knowledge, skills and attitudes to the real task environment: to what extent are trainees able to execute their tasks after they have finished training? The results of the field study show a general lack of data regarding these issues: entry level, dropout rates, training performance, retention and transfer are not systematically registered. This makes it impossible to establish whether simulator training adequately prepares trainees for their future jobs. The performance criteria used for assessment at the end of simulator training are often subjective, unclear or even absent, and the assessment procedure is often subjective. Therefore, it is hard to draw conclusions about the trainees' performance level at the end of simulator training.

In combination with the weakness of assessment procedures, the reported low dropout rates are an alarming sign. A low dropout rate may be explained by the severe selection of trainees at an earlier stage. This might be the case for pilots, where only a small percentage of the applicants are admitted to the training programme, but it is unlikely that this is true everywhere. For tank gunners or truck drivers, for example, there is no severe selection process. An alternative explanation for low dropout rates would be that trainees continue practising until they perform all tasks up to criterion level. However, the results in Section 2.3.2.1 showed that the time available for simulator training is often fixed, and the opportunities for extra practice are limited. It is unlikely that all the courses are so good that nearly all trainees attain the training goals. Therefore, the suspicion has to be that, in a number of cases, trainees who do not master the necessary skills at the end of training go undetected.

Performance data are not only important for the assessment of trainees. They are also necessary for the improvement of the training programme and/or the simulator itself. Systematic registration and long-term storage of data would make it possible to evaluate the effectiveness of specific parts of training or even individual training scenarios. A training goal, for example, obviously needs extra attention when a major part of the trainees does not attain the criterion level of performance at the end of training. The evaluation of transfer data may give insight into the kind of tasks that can and cannot be adequately trained with the current simulator, and thus be useful for the specification of updates, improvements, or an entirely new training system. Systematic registration of performance data is also a prerequisite for retention and transfer studies.

Training with simulators offers opportunities for automatic registration of performance data, and even for automated assessment if measurable performance criteria are available. When this is not possible, support can be provided in the form of evaluation sheets or checklists and data can be stored on computers. Assessment procedures are less subjective when external and/or multiple examiners are used.

2.3.4 Instructors

2.3.4.1 Recruitment and training of instructors

Usually, instructors are experienced operators who are selected for their expertise in the domain to be taught (see Table 2.12). There are some exceptions: in two of the flight simulators the instructor is experienced in a related field, such as avionics or maintenance, and in six other cases the simulator staff is only responsible for scenario management and not for instructional aspects.

Table 2.12: Background of instructors

Background	Number of simulators (out of 44)
Instructors are experienced operators	24[a]
Instructors are not experienced operators	8[a]
No information about background of instructors	13

[a] In a civil flight simulator the instructor during the first half of the course is a technician. During the second half an experienced pilot is involved.

Instructors usually have little or no background in training and instruction. Often, they receive some sort of training to prepare them for their job (see Table 2.13). Instructor training varies in focus and length: from a couple of days to more than a year, but in general it tends to be quite short (i.e. a few days to a few weeks). Sometimes, it is not more than a short introduction to teaching in general, or it only covers the operation of the simulator. In five cases the interviewees ask for better instructor training (see Table 2.6); in four of these five cases the interviewees ask specifically for more training in operating the Instructor Operator Station (IOS).

Table 2.13: Education for instructors

Type of instructor training	Subtotal	Number of simulators (out of 44)
No training		8
Training only covers operation of simulator		6
Training includes teaching skills		
- only short introduction to teaching in general	5[a]	18
- more elaborate training in teaching skills	5	
- no information about the length of instructor training	9[a]	
Unknown whether instructors receive training		12

[a] In a civil flight simulator the technician who provides the first half of training gets an elaborate training, the pilot who is instructor during the second half only gets a short training.

Syllabi for trainees are often available and instructors use these to organise training programmes. Technical manuals that describe the operation and maintenance of the simulator may be available as well. However, knowledge or guidelines for the instructors regarding the adequate delivery of training are usually not included in these manuals. Only five examples of specific instructor manuals are reported. Three of them are for simulators used in civil schools; one of the two military examples only covers briefing procedures.

2.3.4.2 Tasks of instructors

Instructors usually fulfil a wide variety of instructional and more technical tasks. During training sessions the workload can be high: a number of interviewees explicitly mention that supervising and monitoring training sessions is hard work and requires a lot of concentration. In six cases the interviewees state that the workload during training sessions is too high (see Table 2.6). According to the reports the tasks of instructors during training can include:

- Explaining theory: providing theory lessons, and explanations during briefing and debriefing.
- Monitoring: monitoring performance of trainees during training sessions; collecting information for the purpose of feedback and assessment.
- Feedback: providing feedback and guidance during exercises when trainees make errors or perform not at the required level.
- Assessment: evaluating whether trainees attain the training goals.
- Debriefing: discussing the trainees' performance during the debriefing at the end of the training session, providing additional instruction, if necessary.
- Management: loading scenarios from the database, setting parameters, and handling scenarios during training.

- Individualisation: adapting the amount of training, the kind or sequence of scenarios and/or the training scenarios themselves before or during training.
- Other players: playing the role of other players when these are not simulated by the system.

Additional tasks of the instructors can be:
- Organise training: organising courses, arranging classrooms, arranging the availability of simulators and other instructional products, administration, etc.
- Revise training: adapt training programmes to changes in, for example, available training time, updates of the real system, or changes in the entrance level.
- Develop scenarios: designing or revising training scenarios and implementing them.
- Maintenance: maintenance of the simulator itself and/or the IOS.

During training tasks can be divided between staff members, e.g., one instructor is responsible for scenario management and one or more others monitor trainees and provide feedback. Sometimes extra personnel is asked to play other roles in the scenario, e.g., enemy forces or air traffic control. In some cases technical support or specialised personnel is available for technical tasks, such as maintenance and implementation of scenarios.

2.3.4.3 Support for instructors

In most cases one or more specific Instructor Operator Stations (IOS) are available, although the amount and quality of the available facilities varies widely (see Table 2.14). In two cases, there is no IOS: in a gunnery trainer where no scenarios are used, and in a simulator for electronic warfare operators where scenarios are downloaded directly on the simulator itself. The instructors monitor the trainees' performance by 'watching over their shoulder'.

Table 2.14: Instructor Operator Stations

Type of facilities	available	not available	no information
Instructor Operator Station (IOS)	34	2	8
Monitoring facilities	30[a]	3	11
Facilities to provide (audio) feedback	20	6	18
Facilities for debriefing	15	13	16
Facilities to implement new scenarios[b]	29	3	12

[a] In six cases the monitoring facilities are very limited.
[b] See also Table 2.8.

All the IOS provide facilities for lesson and scenario management, e.g., to load training scenarios from the database, to freeze, abort or restart scenarios, fast forward or rewind scenarios, to adapt them before or during training, or to play the role of other players. Most IOS also provide facilities for monitoring, that can include one or more of the following: an overview of the scenario in the form of an event time-line, an overview of the route or flight path taken by trainees, a bird's eye view of the gaming area, a view of the trainees' control settings and/or control panels, the same 'outside' view that trainees have, or a video view of the trainees themselves. A few simulators automatically detect errors or automatically score the trainees' performance (see § 2.3.3.2) and present the results on the IOS. Some others provide a checklist of performance criteria for the current exercise, and/or information about the trainees' performance in the past. Although almost half of the simulators is (also) used to train teams (see Table 2.1), only one of them contains the kind of facilities necessary to monitor team performance (see, e.g., Salas, Bowers, and Cannon-Bowers, 1995). In this command and control simulator the instructors can listen to and record the communication between team members. The reported facilities available for monitoring range from almost nothing to very elaborate systems. They are often limited when the instructor is close to the trainees and can 'look over their shoulder' to see what they are doing. When the instructor is located elsewhere, the IOS provides audio communication facilities. A few simulators provide other feedback facilities as well. In the truck simulator, for example, the instructor can pause the scenario and show a bird's eye view of the situation to the trainee.

About half of the IOS also provide facilities to support debriefing, for instance to register and print out performance data, or to store and replay (parts of) scenarios. These facilities are often quite limited, e.g., only a print out of the flight trajectory or a list of parameters. Sometimes more elaborate facilities are available. For instance, in one of the ship simulators the instructor can type in comments and record them with the exercise. At a number of training sites separate classrooms are available for briefing and debriefing. Finally, facilities to implement new scenarios and add them to the database are available in most cases, either on the IOS that is used during training or on a separate workstation (see Table 2.8).

Many interviewees are not satisfied with the facilities to support instructors (see also Table 2.6). They are critical about the usability or the IOS in general (8 cases), the monitoring facilities (9 cases), the facilities to adapt scenarios during training (2 cases), the lack of facilities for data registration (6 cases) or their usability (10 cases), the lack of printing facilities (1 case), the usability of scenario editing facilities (10 cases), the incomplete databases (5 cases) or the lack of editing facilities (1 case).

2.3.4.4 Interviewees' opinion about training value

Most interviewees judge their simulator to be an effective training device, even though some of them claim that improvements are necessary (see Table 2.15).

Table 2.15: Opinion of the interviewees

Opinion	Number of simulators (out of 44)
Simulator is effective training device	24
Simulator is effective training device, but needs to be improved	8
Simulator is not good, but still valuable	1
Simulator is not good	1
No information about interviewees' opinion	10

The most common arguments are that simulator training is cost-effective and offers more opportunities than the real system. With a simulator emergency situations can be practised without risks, and training time can be used more effectively because the environment can be controlled and the density of learning moments can be high. Simulator training is cost-effective because trainees need less of the more expensive training time with the real system after practising with the simulator. Only in two cases are the interviewees not satisfied with the present simulator.

2.3.4.5 Discussion

Simulator instructors are responsible for a wide variety of tasks: they provide explanations, demonstrations, and feedback; they are responsible for assessment of trainees; they design and revise training; and, if possible, adapt training to the needs of individual trainees. In consequence, training requires large numbers of costly personnel (Kennedy, 2000). Moreover, the quality of simulator training depends heavily on the expertise of instructors. Since most instructors have little or no background in training and instruction, and instructor training is often quite short, it seems that instructors learn these tasks mainly on the job. Instructor manuals describing instruction, guidance, feedback and assessment procedures are usually not available. Thus, teaching is the personal expertise of instructors and it gets lost when they leave. Especially in military organisations, where job rotation is common practice, this is a problem.

Given the importance of the instructors' tasks, it is remarkable that the support for instructors in these training simulators is often poor and limited. Opportunities for automated error detection, standard feedback to common errors and automated assessment are hardly used (see § 2.3.2.4 and § 2.3.2.5). Even quite simple facilities, such as a view of the trainees' console and its settings, and facilities to register and print performance data, are often lacking or insufficient. And in many cases the implementation of new scenarios or training programmes is difficult and time-consuming (see § 2.3.2.3). There seems to be ample room for improvement in this area. The starting point should be that facilities to support instructors are taken into account during the design or acquisition of training simulators (see, for example, Korteling, Helsdingen, and von Baeyer, 2000, for possibilities to support instructors). If the simulator can take over standard tasks regarding feedback, performance measurement and assessment, the instructors can devote all their attention to monitoring the trainees and deal with the more complex problems they encounter (see for an example Kuiper, 1995).

2.4 Discussion

In this section I address the research questions:
§ 2.4.1: Which factors determine the (kind and level of) fidelity of training simulators?
§ 2.4.2: Do simulators have the necessary facilities to support training and instruction?
§ 2.4.3: Which issues should be addressed during the design of functional specifications for training simulators from an educational point of view?

2.4.1 Factors that determine fidelity in practice

From a theoretical perspective the required level of fidelity depends heavily on the training goals. For those tasks and task aspects that should be covered with the training simulator, at least functional fidelity should be guaranteed: the stimulus and response elements should be the same as in the real environments, so that trainees learn the behaviour that they need in operational tasks under similar conditions (see § 1.2.2). The field study indicates that this perspective is not used much in practice. In only three cases is there evidence that the specifications for the simulator had been deduced from training goals. In almost all other cases the original specifications were not available, which made it impossible to conclude which factors have determined the level of fidelity. However, the field study reports, my own experience, and the experiences of other authors (e.g., Farmer et al., 1999; van Rooij, 1997) all indicate that in most cases specifications were not derived from training goals to be covered. According to Nash (2002) it is surprising how often training needs are not analysed or, perhaps worse still, the results ignored when it throws-up some unpalatable training requirements. It its not uncommon for the designers of training simulator specifications to opt for as much (physical) fidelity as possible within the given financial and technological constraints, without an analysis from a training point of view (see also § 2.4.4).

It seems to be the other way around: the level of fidelity of the training system determines which training goals are covered with the simulator and which are not. This is not necessarily a problem when the remaining goals can be covered during field training. However, the necessity for high (functional) fidelity increases when higher demands are posed on the trainees' performance at the end of simulator training (Boer, 1991). Some amount of field training will probably always be desirable. Stammers (1986) warns that the deviations between the learning environment and the real task environment make it impossible to predict exactly how trainees will react under real operating conditions. Moreover, the actual transfer from learning to operational environment is often hard to check since certain tasks are almost never executed in reality: many trainees will never meet emergency situations and many soldiers will never meet the war situations that they train for. The additional costs of a more elaborate training simulator should always be seen in the light of the costs, financial and otherwise, of not completely trained operators.

This study confirms that differences between the simulated and operational environment do not always affect learning and transfer. The ship simulators that do not have a moving base are a good example: the illusion of movement that is brought about by moving the projected image of the outside world is sufficient for training purposes because trainees do not need more motion cues to execute their tasks. A more extreme example in this study is a tank simulator that needs an expensive moving base because it has plate steel armour; for the execution of driving and gunnery tasks, however, it makes no difference whether the armour is made of plate steel or cardboard. More realism is not necessarily better. Sometimes differences between the real task environment and the training environment can even improve training effectiveness, e.g., because the timing of events can be speeded up or slowed down, or because trainees can be forced to focus on certain task aspects. An example in this study is the case of helicopter training where trainees are forced to practice flying on instruments because there is no outside view. Wallace and Northham (1998) advise designers to explore beyond the realm of realism to search for synthetic or 'unreal' aids and techniques that may improve learning and transfer.

2.4.2 Facilities to support training and instruction

In the last decades the needs for simulator-based training changed and increased due to, for example, the development of more advanced operational systems that demand more complex and diverse training for complex tasks, decreased budgets for training, insufficient availability of training grounds and environmental restrictions. At the same time users became aware that training simulators do not always meet the expectations and/or are very expensive. The main reason is not so much a lack of knowledge but a failure to use knowledge gained elsewhere. Knowledge about simulators and simulator training is scattered and not integrated with methods and procedures to design training simulator specifications (Cannon-Bowers, Tannenbaum, Salas, and Converse, 1991; Farmer et al., 1999). This is confirmed by the results of this study: training simulators can be valuable instructional products, but at present their potential for training and instruction is not fully exploited. It is not clear why this is the case, because documentation about the original specifications of the simulators is rarely available. It is likely, however, that these issues did not receive enough attention during the design of specifications.

According to Reigeluth and Schwartz (1989) three major aspects of its design determine the effectiveness of a training simulator: the scenario, the underlying model and the instructional overlay. The latter, they claim, is by far the weakest at this moment. The results of this study confirm this conclusion: facilities to support training and instruction are often poor and a heavy task remains on the shoulders of the instructors, whose support facilities on the IOS are often quite limited as well. This is remarkable because with extra facilities to support training and instruction the effectiveness of training can be considerably improved with relatively small investments (compared to improvements in fidelity). The use of standard scenarios combined with (partly) automated feedback and assessment makes training more controllable and objective, and ensures that all training goals are covered. Standardising and documenting the instructors' tasks is also important from the viewpoint of knowledge management, i.e. to prevent that the instructors' expertise gets lost when they leave the job. Finally, and most importantly, automating or supporting (some of) the instructors' tasks will improve efficiency of training in two ways: on the one hand, it may be possible to reduce the instructor/trainee ratio (cost efficiency), while on the other hand, automating some of the instructors' tasks makes it possible for the instructors to focus on more complex problems, thus enhancing the learning process.

Earlier studies have shown, however, that facilities to support training and instruction are only used when instructors recognise their value and are trained to use them, and when the facilities are easy to use (Polzella, Hubbard, Brown, and McLean, 1987). Literature in the field of instructional design can help to decide how training for different kind of tasks should be designed (see, e.g., Gagné, 1985; Patrick, 1992). Checklists for investigating the validity of existing simulators that include facilities to support training and instruction are available (e.g., in Korteling, Padmos, Helsdingen, and Sluimer, 2001; van Rooij, 1997; Swezey, Owens, Bergondy, and Salas, 1998).

2.4.3 Requirements for training simulator specifications

In the introduction I stated that training simulators should meet two demands to be good training devices: they should adequately represent the real, operational situation and they should support training and instruction. Addressing both demands during the development or purchase of simulators is important, even more so because many simulators are used for a long time. The results of the field study show that some simulators dating from the mid-seventies are still in use. There is a tendency to give much attention to technical aspects, especially physical fidelity, during design. Therefore, I have defined which issues should be addressed from an educational point of view. Some of these issues concern the (part of the) course that will be covered in the future training simulator:

A) Simulator coverage
To ensure the simulator's functional fidelity, designers should analyse which tasks will have to be trained with the future simulator, i.e. which training goals will be covered and which performance level trainees have to attain with the simulator, which training goals will not be trained with the future simulator and how they will be covered. Refresher training should be taken into account.

B) Integration in the course
Practice sessions with the simulator can be combined with other forms of learning, e.g., lectures, CBT and field training. Together they should cover all the training goals. An integrated training trajectory should ensure transfer from one instructional product to the other and eventually to the real task environment.

C) The target group(s) of trainees
The difference between the trainees' entry level and the required skill level at the end of training, reflects what they need to learn. Other characteristics, such as background and educational level, also influence the required fidelity, the training programme and the way instruction should be provided.

Other issues concern the training process itself:

D) Individualisation of training
Facilities to adapt training to individual trainees are required to make it possible to ensure that all trainees attain all training goals, especially when the entrance level of trainees differs. This includes facilities to provide (remedial) explanations, to add or remove scenarios, and to allow more or less practice time. When training can be adapted to individual trainees, these differentiations in training programmes should be registered

E) Instruction
Facilities to support activities to enhance training efficiency -including explanations, demonstrations, feedback, etc.- during the training sessions and during the briefing and debriefing. The use of simulators offers possibilities to (partly) automate these tasks, thus reducing the workload of the instructors and standardising training.

F)Assessment of trainees
Criteria to evaluate the performance of trainees during and at the end of training should be specified beforehand. It may be necessary to test the entry level of trainees to ensure that they have mastered the prerequisite knowledge and skills. Automated assessment, if possible, ensures objectivity and lightens the instructors' workload.

Finally, some issues concern knowledge management and instructor support:

G) Instructor support
Different kinds of facilities to support instructors have been mentioned in the previous points. Another aspect that needs to be considered is the division of tasks between the instructor(s) and the training simulator. To ensure an optimal training process designers need to take the instructors' point of view, determining what the tasks of instructors will be before, during and after training and making sure that adequate support is available. Automating standard tasks as much as possible diminishes the workload of instructors.

H) Quality assurance and evaluation of training
The quality of the training programme should be evaluated and improved continuously. The analysis of performance data can provide insight into the strong and weak aspects, and give indications for improvements. For this purpose, facilities to register, store and analyse the performance data of trainees are required.

I) Editing facilities for training programmes and scenarios
Over time, the training programme may have to be adapted, e.g., because the operational tasks have changed or because the entry level of trainees is different from what was expected. New scenarios and maybe even entire new training programmes will have to be implemented. Editing facilities for this purpose should be available and easy to use.

J) Documentation of specifications and specification process
The training simulator specifications and the way they have been designed should be documented and remain available, not only to provide information about the intended training coverage, but also to serve as basis for the specification of future improvements or for the specification of a new training simulator. Simulators are used for a long time, so by the time they are updated or replaced, the people who were involved in the specification and procurement will often not be available anymore. The only way to reuse information from previous projects and learn from mistakes in the past is to store the simulator specifications, together with the experiences of the designers and the users of the training system.

2.4.4 Conclusion

To ensure functional fidelity, a careful analysis of the tasks that will be covered by simulator training is necessary: the designer needs to have a clear view of the content of training, i.e. the range of training goals that will be covered with the simulator. To exploit the full advantages of training with simulators, however, the designer also needs to know how training will be provided. At an early stage, insight into the future training methods and programmes is necessary to plan sufficient facilities for training programme revision and scenario editing, individualisation of training, instruction and instructor support, assessment, and training programme evaluation.

Based on the results of this field study, however, the suspicion arises that training needs are typically not the major focus during the design of training simulators. It is interesting to note that many instructors are inclined to ask for full mission simulators with expensive image and motion systems, even when they know sometimes that this is not strictly necessary for training purposes (see § 2.3.1.4). It is not unlikely that designers of specifications strive for as much realism as possible within the available budget without an analysis from an educational point of view. According to Farmer et al. (1999) another reason might be that those responsible for the specification of training simulators are usually not those responsible for training, but operational staff or those who designed the operational system itself. Even when instructors are involved they are Subject Matter Experts (SMEs) who have little or no background in training and instruction and are unaware of the opportunities of training with simulators (see also § 3.2.1). The result is that specifications are often technology-pushed and product-oriented, i.e. written in terms of a simulation of the operational system or describing an existing training system seen elsewhere (Farmer et al., 1999; Verstegen and van der Hulst, 2000; Wallace and Northham, 1998). In other cases, the functional specifications remain vague for a long time, and elaboration in more detail is postponed. This may cause unacceptable delays in the availability of training facilities.

From their experience Farmer et al. (1999) describe a doom scenario of what can happen: a new operational system is acquired before training is specified and, thus, introduced into the organisation before a training simulator is available. This time gap requires emergency measures such as outsourcing training or using other training facilities. However, since the newly procured system is often the leading edge of technology the available training facilities are likely to be out of date or not suitable. Under time pressure, shortcuts are used during the design of training simulator specifications, i.e. existing specifications of a previous system or another training establishment are used and adapted or an existing simulator is bought without further analysis. In this situation training programmes and scenarios are designed only after the simulator has been delivered and then, when the limitations of the simulator are discovered, training is fitted to the possibilities. Obviously, this may cause transfer problems.

In this chapter I have described the post-hoc analysis of data collected during field visits to 44 European training simulators that are used to train operator tasks. The data show some differences between the simulators used in military settings and those used in civil education: the involvement of the current instructors in the design and purchase is less in military settings, possibly due to frequent job rotation; and the use of systematic design of training, assessment and data registration is more frequent in civil settings, possibly due to the fact that most of them fall under the rules for (subsidised) state education. This field study concerns a large sample of training simulators, representative for the situation in the participating countries (i.e. military organisations in the Netherlands, Spain, France and the United Kingdom; and civil schools in the Netherlands), and probably also in the rest of the world. It should be noted that some of the simulators are quite old, and are scheduled to be replaced within five years, so it is possible that newer simulators make better use of the opportunities to improve training and instruction with simulators. However, the issues defined in Section 2.4.3 will remain valid and useful.

In Chapter 3, I will take a closer look at the process of designing functional specifications for training simulators. In Chapter 4, I will describe a method specifically tailored to the design of functional specifications for training simulators and I will discuss if and how this method addresses the issues defined above.

Chapter 3
Developing specifications for training simulators

In Chapter 3, I take a closer look at the process of designing functional specifications for training simulators. I compare the characteristics of this process with descriptive research about the process of instructional design in general and describe various types of prescriptive models that are available in the literature to support the instructional design process. The research questions addressed in this chapter are:

4. How are functional specifications for training simulators designed?
5. Which aspects of the design process need to be supported (process requirements)?

3.1 Introduction

In Section 1.3, I argued that training with simulators can have advantages for training and instruction and can be cheaper then training with operational systems. It does, however, require a large up-front investment in both time and money. Since simulators are technically complex systems, their production usually requires specialised personnel and is often contracted out to industry. Future users have to put time and effort into the design of functional specifications, i.e. the user requirements, in order to evaluate whether a simulator available on the market can fulfil their training needs, or to order a custom-made simulator (see § 1.3.2). The term user requirements is ambiguous in this context, since there are several types of users: trainees, instructors, and training designers, for instance. They have different needs: trainees need facilities to practice and learn, instructors need facilities to support the learning process (e.g., to provide guidance and feedback and to assess the trainee's performance), and training designers need facilities to make training scenarios and training programmes. Functional specifications describe what the future simulator should be able to do. In other words: they specify a simulator that can fulfil the requirements of the different types of users. In the development process functional specifications are a step before the technical specifications: they specify on a behavioural level what the simulator should be capable of, not how this can or should be technically implemented.

In the previous chapter the main question was: what makes simulators good training devices? The results of the field study show how training simulators are currently used and which problems users encounter. The solution for these problems lies in improving the design of training with simulators and the design of future simulators. In the conclusions I formulated a list of issues that should be covered in functional specifications. This list poses demands on the product, i.e. the functional specifications, but does not say anything about the design process, i.e. how functional specifications for training simulators should be designed to ensure that all these issues are covered. In the second half of this thesis I will describe research that I have done using the MASTER method for the design of functional specifications for training simulators (Farmer, Jorna, Riemersma, van Rooij and Moraal, 1999). The MASTER method does not, however, support all aspects of the design process (see § 4.3). In this chapter I will present a broader literature study and make an inventory of the different forms of support that are or could be made available. The design of functional specifications for training simulators is a rather specific subject. Since there is almost no literature that addresses this specific topic, I have broadened the literature research to theories about, and support for, the design of specifications for instructional products in general.

In the literature the terms design and development are used in different ways. Both terms are used to refer to the whole process of analysing what learners need to learn, designing how they can learn most effectively and producing the required instructional products (see Figure 1.1). Both terms are also used to refer to particular phases in that process (i.e. the term development is often used instead of production). The term instructional design can also be used as a noun and then refers to the product of the design phase, or to the body of literature or theory in this field. In the light of my research questions the distinction between product and process aspects is important. Therefore, I use the term instructional design for the product of the design phase and the term instructional design process for the activities executed by the instructional designer during the design phase. I use the term production for the third phase, and reserve the term instructional development for the whole process (i.e. the Analysis process, the Design process and the Production process, cf. Figure 1.1).

In Section 3.2, I describe the characteristics of the design process based on my own research and on descriptive studies reported in the literature. In Section 3.3, I discuss prescriptive models that can possibly support the design process: system theory, instructional design models, instructional development models, rapid prototyping, acquisition procedures, and organisational aspects. Then, I briefly discuss which kinds of tools are available. In Section 3.4, I look at the results of the literature study from the perspective of support for the different aspects of the design process.

3.2 Descriptive studies: the design process in practice

3.2.1 Designing specifications for training simulators

From experience and discussions with personnel responsible for the design of simulator specifications in the Royal Netherlands Army (RNLA), eight characteristics of this process were derived (Verstegen, Barnard, van der Hulst, and Sabel, 2000; Verstegen and van der Hulst, 2000):

1) Complex domains
Simulators are often used for training in complex domains with technically advanced systems, such as aircraft, (nuclear) power plants, or ships. The task analysis and the analysis of the knowledge and skills required to operate these systems are difficult and labour intensive. It involves a large amount of data being collected, structured and stored. It is difficult to decide beforehand which information will be relevant, which makes it hard to decide in how much detail the analysis should be done.

2) Different kinds of expertise needed
Ideally, the design of simulator specifications is the responsibility of a team. Co-operation in such a team is complicated by the fact that team members are experienced in their own field, but lack expertise in other fields. Communication may be difficult because the team members have different backgrounds and use a different vocabulary. Different kinds of expertise are required for the design of functional specifications for training simulators, such as knowledge about the operational domain, knowledge about the technological possibilities and the market, and expertise regarding training and instruction. The latter is often brought in by domain experts who have become instructors and course designers. These probably have experience with simulator training (both as trainees and instructors), but not with the specification of training simulators. There is no specific training to prepare RNLA personnel for a role in training simulator specifications. More in general, issues regarding training and instruction are addressed in the courses for instructor (total length 15 days) and for course designer (total length 22 days).

3) Multiple constraints
In practice, buying the 'ideal' training solution is usually not possible. The final set of specifications for a training simulator is not only determined by instructional factors, but also by a large number of other constraints such as the available budget, time limits, technological possibilities, personnel available for the specification and/or the delivery of training, the cost of maintenance and logistic constraints.

4) Different kinds of projects
Every design process is different. There are at least three distinct situations: simulators can be necessary to train personnel for new operational systems, they can be acquired to replace existing training simulators or they can be acquired to solve existing training problems, for example when not enough instructors are available or the opportunities for training on the job become limited because of environmental constraints. Projects also differ in complexity, the number of people involved, the type of constraints and aspects that play an important role, etc.

5) Contradicting constraints and interests
Constraints often contradict each other: with a cheaper simulator, for example, more instructors may be needed or the amount or trainees that can be trained may be limited; and time

limits may force the designers to execute less detailed analyses and take more risks than they would like to do. Moreover, the different parties involved in the design of functional specifications may have conflicting interests: the instructors aim for an optimal training system, the management and the financial experts aim for the cheapest solution and a technical expert may be looking for a chance to apply new techniques or systems.

6) Long and time-consuming process
The specification and acquisition of training simulators is a long and time-consuming process due not only to its complexity but also to the procedures that have to be followed for the procurement of (expensive) training systems (see § 3.3.6). Years pass between the identification of new training needs to the actual introduction of a training simulator. Four to five year is considered normal (van Rooij, 2002), and it can be much longer when complications occur, such as technical problems, personnel shortage or a delay in decisions about budgets. Kincaid (1997) gives an example of a major equipment acquisition project that lasted for 15 years. Over the years the constraints change, e.g., due to technical developments or budget cuts. In military settings job rotation is another complication: it is difficult to have a stable design team, when personnel changes occur every two or three years.

7) Information is incomplete and insecure
One of the biggest problems is that the available information is usually incomplete and uncertain. When new operational systems and training simulators are procured in parallel it may not be clear what the future tasks of trainees will be exactly because the operational system is still under development. Adducchio (1997) gives an example of a project where the requirements changed during the design process: when a large part of the instructional analysis had already been done, it was decided that the future simulator should not only be suitable for training but also for certification purposes. Dependencies in the design process itself also cause uncertainty: designers will have to make decisions when it is not yet clear what these are feasible both technically and financially. Later on, revisions of earlier decisions may be necessary. For example: when a set of training goals has initially been allocated to simulator training, further analysis may show that a much cheaper simulator can be used for most of them. In this case, it may be more efficient to train for the few remaining training goals elsewhere, e.g., during field training with the operational system.

8) Timing problems
The RNLA personnel reports timing problems: to be able to choose the best solution from an educational point of view, functional specifications should be based on training goals and a comparison of alternative solutions. This would imply that writing specifications for a training simulator should be postponed until all the necessary information is available and thorough analyses have been completed. In practice, however, this is impossible for two reasons: firstly, functional specifications are necessary at an early stage to claim financial resources. Secondly, training should be implemented and available before the operational systems are delivered.

Similar characteristics are mentioned by Dehncke and Brooks (1998) who report on the design of the Synthetic Theater of War (STOW), a complex simulation environment for the UK and US Armies: conflicting demands, the need for a design team that includes the key management and technical personnel (but is not too big), change of personnel during the five year project (the STOW project actually had five project managers) and timing problems because the information is always incomplete. They conclude (p. 38): "The fact of the matter is that you will never have enough information in the format you need it. You must push on with less than perfect information..."

Thus, the design of functional specifications for training simulators is an inherently iterative process. Preliminary decisions will have to be taken before all the necessary information and input become available. The costs and benefits of alternative solutions have to be weighed against each other continuously, taking conflicting and changing constraints into account. Earlier decisions will have to be revised when new information becomes available or when further analysis leads to new insights. The people responsible for the design of functional specifications for training simulators often do not have enough resources, expertise and experience to execute a thorough analysis of training goals and alternative solutions (see also § 2.4.4).

3.2.2 Designing instructional products in general

In this section I report findings from descriptive research about the design of functional specifications for instructional products in general. First, in Section 3.2.2.1, I describe the results of studies that give information about the instructional design process in practice by professionals and non-professionals. In Section 3.2.2.2, I take a step back, look at instructional design form a problem solving perspective and I discuss which characteristics of problem solving also apply to instructional design tasks. In Section 3.2.2.3, I focus on instructional design again and describe research regarding the differences between novice and expert instructional designers.

3.2.2.1 Instructional design by professionals and non-professionals

Instructional design by non-professionals
In practice, instructional design is often done by teachers or instructors who are not professional instructional designers. This is also the case for training simulator specification (see § 2.2.1). The results of the field study indicate that these non-professionals are not inclined to use a systematic approach for the design of instructional products (see § 2.3.1.3 and § 2.4.1). This is confirmed by descriptive research in other settings. Within the IMAT project (ESPRIT no. 29175), for example, methods and tools to reuse material from technical manuals for instructional purposes were subjected to evaluation studies with teachers (de Hoog et al., 2002). During these evaluations the selection of topics and the design of lessons was guided by the structure of the operational systems, i.e. its components and the associated tasks (Verstegen, Veldhuis, Staalstra, and Hendriks, 2001). The teachers did not use the available tool to make lesson scenarios based on learning goals. Odenthal, Kuiper, Voogt, and Terwindt (2000) report similar results: during a project for curriculum innovation school teachers often worked 'intuitively', based on their own teaching experience. Attempts to structure the design process were sometimes valued, but were frequently viewed as very restrictive. The resulting instructional products, however, were often not as innovative as had been intended. Another example is reported by Gilbert (1999): during the evaluation of the Teaching and Learning Technology Programme in the UK he found that Subject Matter Experts (SMEs) do not apply an integrated, systematic approach when they are given the task of designing computer-aided instruction. Moreover, they do not realise that they do not have the necessary expertise about instructional design and computer-aided instruction.

Hoogveld, Paas, Jochems, and van Merriënboer (2002) found indications that translating desired changes in the framework of the curriculum into concrete, new learning practices is difficult. Based on interviews with ten teachers of a teacher-training college Hoogveld et al. concluded that these teachers, when they design study units, do not give much attention to the analysis of the design problem and the preparations necessary to be able to carry out evaluations (cf. the fact that facilities for data registration and storage in training simulators are often poor, see § 2.3.3.2). In an empirical study Hoogveld, Paas, Jochems, and van Merriënboer (2001) found that the designs of eight teachers who had been trained in applying a specific prescriptive model (the Four-Component Instructional Design model, see § 3.3.4.1) were rated significantly better than the design of five teachers who had received the same amount of training to improve their own experience-based design method. Moreover, in a follow-up study Hoogveld (2003) found that low achievers also performed better when they worked collaboratively then when they worked individually. For high achievers, however, there was no advantage of working in a team. Van Berlo (2002) describes similar results from a study regarding the analysis phase: he asked ten novice, military instructional designers to analyse two team tasks and formulate learning goals. In the pre-test subjects completed their task without help. Before the post-test subjects spent one hour studying a set of instructional design guidelines. Half of the subjects received a version that explicitly focuses on team aspects, the other half got a control version excluding all team-related aspects. Van Berlo scored the quality of the task analysis process using think-aloud protocols. The results show that all subjects performed better on the post-test than on the pre-test, and the group that got the guidelines focusing on team aspects also performed better on the team aspects. Van Berlo also found very large differences in the quality of the products and the quantity of notes, possibly (partly) caused by the fact the available time (three hours) was not enough for some subjects. Winnips (2001) found that student instructional designers also reacted positively to the availability of design guidelines during a design assignment. Accompanying exercises, however, were only done when they got some kind of reward for them.

Instructional design by professionals

There are indications that there are differences between SMEs and professional instructional designers. Saroyan (1993) asked experienced SMEs and instructional designers to think aloud during the formative evaluation and review of an instructional text on microbiology. His analysis of the six protocols shows that SMEs approach the task as a specialist, they seem to be directed by domain knowledge and they use a sequential method of review. The instructional designers, on the other hand, take on the role of generalist: they appear to be directed by the heuristics of the Instructional System Development approach (ISD, see § 3.3.3). The two approaches led to the identification of different problems and the generation of different revision recommendations. It is not surprising that professional instructional designers design specifications for instructional products in a more systematic way, since they are trained to do so. Their curriculum usually includes courses on different kinds of prescriptive design and development models (see § 3.3.2 and § 3.3.3).

Rowland and Adams (1999) asked experienced designers to identify key components of the design process and found that the results could be clustered in seven categories: problem situation, goals, learners, content, instructional setting, instructional method and assessment. These aspects play an important role in prescriptive models as well. Other studies also show that professional instructional designers use systematic design and development models, although not very accurately: they conduct some of the prescribed activities, but not all of them and not always the same ones. Based on extensive interviews with four experienced designers, Perez and Neiderman (1993) conclude that the design process of these professionals reflects a systematic approach at the top level. At a more detailed level, however, the implementation and the way the designers describe the process vary a lot. In a laboratory study Le Maistre and Weston (1996) asked eight professional instructional designers to revise existing instructional modules. They found that the subjects do use procedures for formative evaluation that appear in standard textbooks but not as much as anticipated; about 80 % of the revisions was based on the revisers' own knowledge and not on the available feedback data from learners and experts. This is in accordance with findings from questionnaires reported in Pieters and Bergman (1995), and Visscher-Voerman (2000, 1999, see also below) who found that the execution of design activities depends, amongst others, on the size of the projects and the constraints impeded by clients. Wedman and Tessmer (1993) asked instructional designers why activities were omitted. The most frequently chosen answers were: 'decision already made before the start of the project', 'lack of time', and 'considered unnecessary'.

There are indications that following a systematic approach improves the instructional design. In a study of 17 cases, Kessels (1993) found that in successful cases a significantly better systematic approach had been applied than in unsuccessful cases. He also found evidence for the success of applying a relational or communicative approach that challenges stakeholders to become involved and reveal their perceptions of the ideal curriculum (see further § 3.3.7.1). In contrast, in a survey Holcomb, Wedman, and Tessmer (1996) found no clear relationship between the perceived success of a project and the inclusion or exclusion of prescribed activities, or the thoroughness with which they were carried out.

Visscher-Voerman (1999) interviewed 24 professional instructional designers with the purpose of finding patterns of design activities. She concludes that this is not possible. Designers perform a wide range of different activities, and what is more, in the context of different projects and settings the designers' explanations of why certain activities were or were not executed seemed in most cases valid and justifiable. Closer inspection of the results showed that especially the analysis and evaluation phases were not executed as elaborately as often prescribed (Note that Hoogveld, 2003, found similar results). The design and development phases were not clearly distinguished by the interviewees. It seems that professional instructional designers use different strategies to design and develop, and often perform activities in an integrated way. All instructional designers paid attention to the upcoming implementation of the instructional products that they were designing, but most of them were not involved in the implementation themselves. Searching for an articulation of design approaches and strategies embedded in specific project contexts, Visscher-Voerman defined four instructional design paradigms (Visscher-Voerman, 1999; Visscher-Voerman, Gustafson, and Plomp, 1999):

- Instrumental: The focus of the instrumental paradigm is planning-by-objectives, the project starts with an extensive analysis of intended outcomes and the design process is carefully planned with milestones and evaluation moments with these goals as reference. Logical reasoning and working systematically are considered essential.
- Communicative: Design is viewed as a process in which different perspectives and opinions are being communicated and negotiated until decisions can be made and consensus is reached.
- Pragmatic: The design process is shaped as an interactive and repeated try-out and revision of design prototypes.
- Artistic: Instructional design is seen as a form of art that cannot be reduced to any specific method.

Visscher-Voerman's study revealed that the majority of the interviewed instructional designers applied an instrumental (14 designers) or communicative (7 designers) approach. The other three used a pragmatic approach (Note that in four cases designers had been obligated to use an instrumental approach, and would have preferred another paradigm). She concludes that subjects seem to choose based on specific activities rather then the whole approach, and that at that level the approaches are not mutually exclusive. Visscher-Voerman found that the instrumental and communicative paradigm were also valued most during a workshop with instructional designers. Most designers (75%) agreed with the statement that successful design is served by the use of step-by-step schemes and design models, provided that they are adapted to the specific project.

Many prescriptive development models follow the instrumental paradigm (most notably the ISD models, see § 3.3.3). A clear example of the pragmatic paradigm is the rapid prototyping approach (see § 3.3.5.1). Some examples following the communicative paradigm will be described in Section § 3.3.7.1.

3.2.2.2 The problem solving perspective

In this section I will take a step back, I will look at instructional design from a problem solving perspective, and I will discuss which characteristics of problem solving also apply to instructional design tasks.

Design tasks can be seen as problem solving tasks. Solving a design problem will result in the sketch of an artefact that, when realised, will fulfil or help to fulfil a need (Dijkstra, 2001b). Problem solving is characterised as the movement from one knowledge state to another, by the application of operators, until the desired end state is reached. Typical for most problems is that the problem-solver does not immediately know what to do (de Jong, 1986; Mettes and Pilot, 1980). Design tasks differ from puzzles in the sense that they cannot be solved with general knowledge only, but require a substantial amount of professional, domain-specific knowledge: knowledge about problem situations as they normally occur in a specific domain, declarative knowledge such as principles and laws that are valid in the domain and procedural knowledge about the actions that are permitted to get to a solution. Not only the presence of different kinds of knowledge is important, but also its organisation: the problem solver should be able to retrieve the right declarative and procedural knowledge for different kinds of problem situations. On top of that, problem solvers need strategic knowledge that indicates how the problem solving process should proceed. This knowledge is less specific, but can still be different for different domains. In the context of solving problems in the physics domain, Mettes and Pilot found that novice problem solvers could be helped to solve thermodynamics problems when a specific instructional strategy was used to teach them explicit problem-specific schemata with a combination of procedural knowledge (heuristics) and declarative knowledge (relationships between concepts). On the other hand, de Jong found that an instructional strategy focusing on strategic knowledge did not make a difference. His subjects did not use this knowledge in solving problems, but followed the same strategy as the control group: they quickly choose a solution after a (too) short analysis, so they frequently had to return again to the analysis in order to correct their original faulty solution (i.e. a 'kick and rush' strategy).

Greeno, Korpi, Jackson and Michalchik (1990) claim that design tasks differ in two ways from other problem solving tasks, such as calculation, proof and explanation problems. First, the problem solution space is open: it is impossible to predict which solutions a designer might come up with since design is an inherently creative activity. Second, the final state is a matter of judgement: the designer decides when the task is completed. Thus, design problems have the characteristics of ill-structured problems: the real-world conditions are ill-structured and involve many complex variables, and there is no perfect solution. Designers can only hope to find a satisfactory solution that meets (most of) the demands (Kerr, 1983; Rowland, 1991, 1993). There is no clear solution process, no algorithm or established set of procedures. Designers have to employ heuristics, problem solving, creativity and decision making (Nelson, Magliaro and Sherman, 1988). Finally, it is not always clear what the problem is. The designer's task includes finding as well as solving problems (Rowland, 1993); or in other words, the design task includes constructing the problem space, as well as finding a solution within the problem space (Greeno et al., 1990). An architect who participated in a study about architectural design expressed it as follows: "A designer only understands the problem that is called an assignment, when the design is finished" (Hamel, 1999, p. 91).

Goel and Pirolli (1989; 1992) defined generic features of design tasks from the viewpoint of (ill-structured) problem solving. Analysing think-aloud protocols they found that these features indeed distinguish design tasks (in the fields of instructional design, architecture, and mechanical engineering) from non-design problem solving tasks. Some of these features are:

- Problem structuring: Many degrees of freedom exist in the problem statement. This means that extensive problem structuring is necessary before problem solving can begin. Problem structuring occurs mainly at the beginning of the task, but reoccurs periodically as needed.
- Distinct phases: Design problem solving can be categorised into three distinct phases: preliminary design, refinement and detailed design. These phases are

generally executed sequentially, but it is not uncommon for subjects to return to an earlier phase as previously unnoticed aspects emerge.

- Modularity: Because of the size and complexity of design problems, designers decompose the problem into a large number of modules. However, these modules are 'leaky', i.e. they are heavily interconnected.
- Incremental design: Interim design ideas are nurtured and incrementally developed until they are appropriate for the task. They are rarely discarded and replaced by new ideas.
- Control structure: The design task is so complex that full control is impossible. Designers use a limited commitment mode control strategy with nested evaluation loops: they generate and evaluate a design component on its own, before evaluating it in a larger context and eventually in the context of the complete design.
- Reverse direction: Designers occasionally stop and attempt to shift problem space boundaries by transforming the problem to fit an existing plan or template. In other words, they reverse the direction of the transformation function and work bottom-up.

Furthermore, Goel and Pirolli describe the tension between keeping options open and making the commitments that are necessary to complete the design task within a finite amount of time, and the complications that arise because there are no objective stopping rules: there are no right or wrong answers and direct feedback during the design process is lacking. Since it is usually not possible to directly manipulate the real world situation to try out solutions, designers construct and manipulate models of possible solutions at various levels of detail using artificial symbol systems.

To investigate how a problem space is constructed during an instructional design task Greeno et al. (1990) analysed think-aloud protocols of eight student teachers (four recent graduates and four just starting the 12 month course). On two separate days the subjects were asked to make a plan to teach two different topics regarding a fictional device. The sessions lasted one to two hours each. They found that the subjects did spend time on shaping the problem space, but they spent far more time on the design itself. Most of the activity involved proposing material, i.e. the content of information to be included and the transactions by which that content would be communicated to the learners. A considerable amount of time was spent on meta-design activities such as recapping, reflecting, evaluating, monitoring and justifying. However, material was only rarely modified or removed once it had been placed into the design. Examining the protocols in great detail, they found some prototypical patterns, but also differences in the kind and sequence of activities that subjects used during their tasks.

Similar studies have been done for other kinds of design tasks. In the domain of architecture, for example, Hamel (1990, 1999) claims that the design task of an architect consists of five aspects: gathering information, decomposition of the design problem, solving the sub-problems, integrating sub-problems solutions and styling the solution to complete a design that meets aesthetic and professional criteria. Hamel combines the first three aspects to one stage in the design process and proposes a model consisting of three main stages: Analysis, Synthesis and Styling. When the architect is confronted with a design problem four types of knowledge are activated: declarative knowledge about the design problem, and procedural knowledge regarding each of the three main phases. In a design process activities are executed in a serial order, and because working memory is limited, the planning of this sequence of activities is important. Moreover, Hamel says, designers use an 'external memory' in the form of books, paper, etc., to deal with the quantity of information (cf. the models and artificial symbol systems mentioned by Goel and Pirolli, 1989, 1992, see above). This leads to a nested model where the same categories of activities -orientation, execution and evaluation- are executed for all four types of knowledge, although the concrete activities can be different, e.g., orientation for the synthesis phase consists of re-reading the available material and making estimates of the combination of solutions to sub-problems, and in the styling phase it consists of studying one's own sketches and making estimates of the future appearance of the building. The model postulates the order of the categories of activities, but not of the activities within one category. Hamel tested his model with think-aloud protocols of 15 experienced architects and found that it fitted their design process, much to the surprise of the subjects who often thought that their way of working was quite chaotic and that trying to model their design process was doomed to fail. Probably, Hamel says, these experienced architects have developed 'good habits' forced by the characteristics of architectural design tasks and their own limited memory capacity, and developed task schemata that are activated with each new design task. Experienced designers do not have to consciously manage the design process and, thus, can have more attention for solving the design problem.

Braha and Maimon (1997) discuss their field, engineering design, from a wider theoretical perspective. They claim that the design process can be viewed as a stepwise, iterative, evolutionary transformation process: as the design process develops, the designer modifies either the tentative design or the requirements based on new evidence (information), so as to remove the discrepancy between them. They give the following characteristics of engineering design:

- Generally, the requirements are, to a large extent, not comparable and therefore the preferred ordering among them is incomplete.
- Alternatives, options and outcomes are usually not given in advance and must be found and developed by some research process.
- Usually, a satisfactory solution is good enough and will be fully accepted.
- For those (parts of) design problems that are well-structured, the solution process is so complex that designers might have to accept a satisfactory solution, even when they know that a more optimal solution exists.

Braha and Maimon compare the design process to scientific research: when an anomaly is discovered in a scientific theory (or in a design), the outcome is often an adjustment of the theory rather than a total dismissal (incremental redesign). The limited information-processing capacity of the designers is at least partly to blame for this. It often forces them to make decisions similar to those previously made. Innovative design, on the other hand, corresponds to a transition to a new paradigm in scientific research.

Blessing (1994) conducted a long-term case study regarding the design process in the field of mechanical engineering. For a period of 34 weeks she collected activity sheets of members of the (large) design team, she observed their work and registered discussions. On top of that she studied time-writing data for the entire length of the project (i.e. 3.5 years). Blessing compared her results with the results of eight overview studies and 66 case studies in the descriptive literature regarding mechanical engineering. She concludes that designers generally seem to follow a product-oriented approach focusing mainly on the refinement of an initial product idea. The abstraction steps typical for a more problem-oriented approach are not addressed. Designers do also not follow the advice to generate and consider alternative solutions in parallel. This is worrying, according to Blessing, since the explicit and frequent analysis of solutions to identify problems and failures, which is important for the successful application of product-oriented approaches, is hardly observed either. In general, designers tend to stick to their solution and do not execute evaluation in a proper way. She derives a list of success factors, which include using a systematic approach, executing all stages and activities, and a thorough problem analysis.

Blessing (1994) proposes a model of the design process as a combination of stages, activities and strategies. Stages reflect a subdivision of the design process based on the state of the product under development. In general three main stages are distinguished: the problem definition stage, the conceptual design stage and the detail design stage. Activities are subdivisions of the design process related to the designer's problem solving process, for example collecting information, generating solutions, evaluating and selecting. They reflect a much finer division than stages, cover a shorter amount of time and typically reoccur several times. Strategies are defined as the sequence in which design activities are planned or executed. Stages, activities and strategies together determine the flow of the design process. Stages are, in principle, executed sequentially and only once (see Figure 3.1-a). The main flow through the activities is cyclic (Figure 3.1-b). Some design models explicitly combine activities and stages, resulting in a cyclic flow as illustrated in Figure 3.1-c: the sequence of activities is repeated in every stage. Taking into account that the part of the solution space that is considered will get gradually smaller, a concentric model can be defined as depicted in Figure 3.1-d.

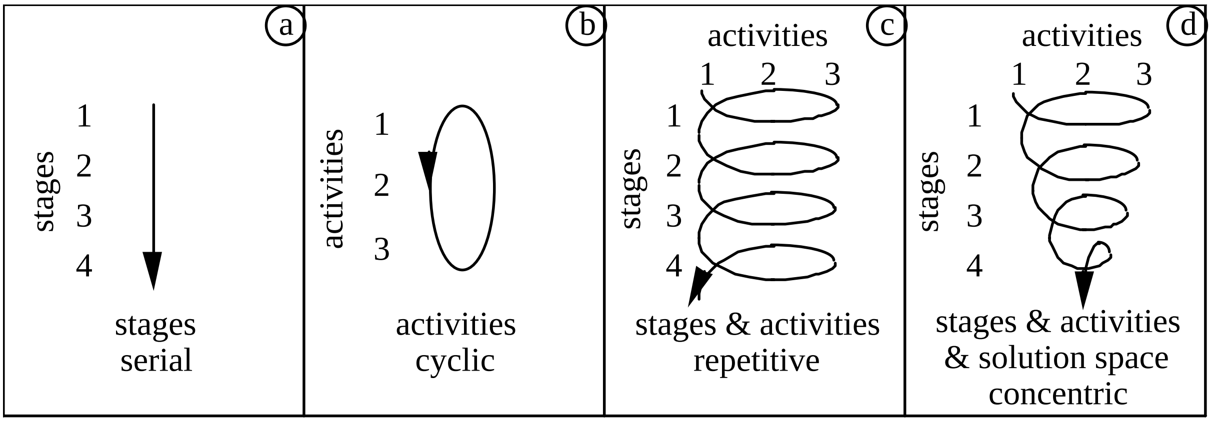


Figure 3.1: Main process flows (in mechanical engineering design, from Blessing, 1994, p. 41; used with permission of the author).

Designers can use different strategies during the design process. The stepwise strategy is most obvious in the linear sequence of design stages, and the planning of design activities is often seen as cyclic. Other strategies not represented in Figure 3.1 are iteration loops usually represented by feedback arrows in prescriptive models, and the decomposition strategy often concealed in the terminology used for different steps in the process. Concretisation strategies refer to the analysis and further elaboration of a product idea. Finally, some prescriptive models prescribe an abstraction and reconsideration of the design problem before generating solutions (i.e. the problem-oriented models). In descriptive studies the activities are often clearly recognisable, but there seems to be no clear distinction between the design stages. Blessing claims that this can be explained by assuming that the stages reoccur for every product element when the design problem is decomposed (and, thus, not for all elements at the same time).

In theory, how stages and activities should be planned and executed depends on the situation, especially on the design problem and the designers. Blessing (1994) concludes, however, that prescriptive literature hardly gives any indications for choosing and applying the different strategies. Moreover, in practice, reality forces designers to deviate from systematic plans (although Blessing does not exclude that the design process could still be called systematic if it was studied at a higher, less detailed level). One reason is that requirements for the problem solution are often extended and reformulated during the entire design process. In her own case study, Blessing also observed a continuous stream of external influences that would not occur in laboratory studies, partly caused by the size of the project and the amount of people (from different departments) working on it. Furthermore, some critical elements got precedence, while others were postponed. Later on, some of those turned out to be critical as well, leading to large modifications in the concept design. Due to time pressure stages were sometimes executed in parallel.

In the next section I will focus on the field of instructional design again, and discuss differences between expert and novice instructional designers.

3.2.2.3 Expert versus novice instructional designers

Goel and Pirolli (1989, 1992) describe solving a design problem as a complex, non-linear process, characterised by the incompleteness of information and the lack of a prescribed solution procedure (see § 3.2.2.2). Designers have found ways to deal with this kind of ill-structured task, e.g., by going through refinement cycles and by decomposing the problem into modules. Research described in the previous sections made clear that the design process varies between designers and between projects. These differences are partly caused by personal preferences and situational parameters, such as the constraints imposed by clients (see § 3.2.2.1). There is also evidence, however, that there are differences between experienced and novice instructional designers. Some relevant studies will be discussed in this section.

In one study, Le Maistre (1998) compared an experienced instructional designer (with 18 years of experience) with a less experienced one (5 years of experience). She found that the more experienced designer was faster and more efficient, decomposed the problem into manageable parts, waited longer before starting revisions, was more concerned with structural problems, constantly adjusted decisions, made more self-monitoring statements and more references to instructional design theory or principles. The less experienced designer worked at a more superficial level and in a linear way. These findings are in line with a number of earlier studies. Kerr (1983), for instance, found that novice designers have difficulty entertaining multiple possible solutions and eliminate alternatives very rapidly; they are not proficient in representing design problems to themselves or to others and they have trouble determining what a reasonable stopping point is.

Another study was executed by Rowland (1991, 1992). He asked four expert instructional designers with more than seven years of experience and four student instructional designers to design instruction in the physics domain and analysed the think-aloud protocols. Rowland distinguishes two phases: problem understanding and solution generation. The experts spend more time in the first phase than the novices do: they look for information to fill the slots of 'templates' they seem to have from experience, and if information is not available they try to infer it. As a result they produce a richly elaborated presentation of the instructional design problem. Novices, on the other hand just try to understand the given information and split the problem into parts based on surface differences. They quickly start the solution generation phase, and commit to one solution early on. Experts also start considering solutions early on, but they delay commitment to one solution until much later. When they are in the solution generation phase, experts try to match the current problem to problem-solution patterns in their memory, they employ domain-specific principles and design principles derived from instructional design theory, they generate alternative solutions, and maintain weak links between problem and solution elements. Experts consider a wide range of factors in combination with each other. Novices, on the other hand, only consider few factors and only one at a time. They rely on general knowledge and their own experiences as teachers, whereas experts draw from their experience as a designer. Rowland's study confirms that experts also differ amongst each other: the four experts in his study each represent the problem differently and design different solutions. There seems to be a high correlation with their personal experiences and perspectives. In a later paper, Rowland (1993) proposes another explanation: since individuals interpret and understand problems differently (a condition that is unavoidable for ill-structured problems), it may be more accurate to say that each individual solves a different problem rather than just generating a different solution to the same problem.

Perez, Fleming-Johnson and Emery (1995) executed a methodologically well controlled study with a formal coding procedure and checks for inter-rater reliability. They compared five expert instructional designers (more than 10 years experience) and three novices (less than two years experience) who were asked to think aloud while making a design for a lesson to teach trouble-shooting a diesel engine. The results are similar: experts use more design principles and rely on a larger variety of knowledge sources; experts spend more time on problem analysis and trying to understand the domain, reflect on past design problems and solutions and draw comparisons with the present problem. Novices immediately begin to consider solutions in detail. Perez et al. claim that the novices in their study must have had enough knowledge about instructional design principles and models because they had quite recently completed several courses on the topic. They conclude that the novices apparently lack the strategic knowledge that is necessary to translate theory into practice.

Goel and Pirolli's characteristics (1989, 1992, see § 3.2.2.2) can be recognised in the behaviour of expert instructional designers, which is in line with characteristics of expertise in other domains (Perez et al., 1995; Perez and Neiderman, 1993; Rowland 1991, 1992). Experienced instructional designers have the knowledge and skills required for design tasks. They collect knowledge about the domain and about the current design problem (i.e. different kinds of demands and available resources, see above) during the analysis phase and construct an elaborate representation of the design problem. The available descriptive studies indicate that professional instructional designers use prescriptive models and heuristics. However, prescriptive models are not followed in detail, but adapted to the situation and to the preferences of the designer. Expert instructional designers probably do not need to strictly follow prescriptive models, because they can see when a step has already been done or is less necessary, or because they have addressed issues implicitly. Probably, expert instructional designers have also learned to recognise and repair errors. In another domain, Mettes and Pilot (1980) found that that expertise was most visible in self-correction: while solving thermodynamics problems, university lecturers did not make fewer errors than their students, but they were able to recognise errors and repair them whereas many students were not.

Experienced instructional designers also reuse (parts of) solutions of other design problems that they have encountered in the past. Nelson et al. (1988) claim that designers develop cognitive structures or schemata which can guide the design process and can be used to categorise, consider and store information. The more experienced they get, the more elaborated and structured these schemata become. Thus, experts are able to recognise patterns, infer relationships, disregard irrelevant information and recall similar problems from the past. Schemata allow experts to quickly apply implicit knowledge gathered from training and experience.

Expert instructional designers solve design problems in different ways and come up with different solutions to the same task. This is not strange, since there is no best solution, only satisfactory ones, and there are no standard solution methods for ill-structured problems. For experts, therefore, any support that structures the design process should leave ample room for flexibility (Rowland, 1991, 1992). The designers of functional specifications for simulators, however, are usually not experts and they lack the necessary resources, expertise and experience (see § 3.2.1). Novices are inclined to spend little time in the analysis phase and to commit themselves to one solution early on. Moreover, several studies showed that designers find it hard to discard ideas or (parts of) solutions[2]. Novices seem to lack the necessary knowledge about possible solutions and solution procedures, and/or the strategic knowledge to apply them to the specific situation at hand. Therefore, they may need a more structured approach (Perez et al., 1995).

In conclusion, the descriptive studies confirm that instructional design can be seen as a kind of problem solving and is an inherently iterative process (even when the time-span in these studies is much shorter than design tasks would be in reality, i.e. a couple of hours, instead of weeks or even months).

---

2 It is possible that this is (partly) an artifact of the experimental settings: subjects may have felt that they did not have enough time to start all over; moreover, in reality designers may be forced to discard ideas when there are no resources to develop them and/or other stakeholders do not accept them.

3.2.3 Discussion: designing simulator specifications

Design tasks are ill-structured problems. This is also true for the specification of training simulators: the domains are complex, every project is different, it is a long and time-consuming process, there are different kinds of -often contradictory- demands and constraints that can change over time and so forth (see § 3.2.1). When the domain is rather small and all the relevant information is immediately available, the design of training simulator specifications might be an almost linear process. Usually, however, this is not the case. Designers will have to make preliminary designs based on incomplete and uncertain information, and they will have to review and alter their decisions frequently when new information becomes available. The outcome is not only determined by what would be best given the tasks to be learned and the target trainees. More practical complications, such as conflicting interests, limited resources, personnel changes and timing problems play a role as well.

Table 3.1: Different kinds of information that can influence the design process

Types of information
Information about project specific organisational and other demands
Information about technical possibilities and knowledge about the market
Information about the domain
Information about the training needs
Instructional design knowledge
Instructional development knowledge
Information about the structure of the organisation
Information about resources available for delivery of training
Information about resources available for acquisition/production of instructional products
Information about the resources available for designing specifications

Table 3.1 gives an overview of different kinds of information that can influence the design process:

- Information about project specific organisational and other demands: the stakeholders, environmental regulations that may apply, the acquisition procedures to be followed (see, e.g., § 3.3.6), the deadline for submitting proposals, etc.
- Information about technical possibilities and about what is already available on the market.
- Information about the domain: which tasks will trainees have to execute in practice, under what conditions, with whom and with which operational systems tasks will be executed, etc.
- Information about the training needs: the training goals, a description of target trainees, the number of trainees expected, etc. (i.e. the results of the analysis phase).
- Instructional design knowledge: knowledge about how instruction and training should be provided, which instructional products to use, etc. (see § 3.3.2).
- Instructional development knowledge: knowledge about how instructional products should be developed (see § 3.3.3).
- Information about the structure of the organisation: who is responsible, who has influence, where different kinds of expertise can be found, which organisational policies should be taken into account, etc.
- Information about resources available for the use of training simulators (and the delivery of training in general): the number and the capabilities of instructors and of maintenance personnel, the available classrooms, available time slots, etc.
- Information about resources available for the acquisition and/or production of training simulators (and other instructional products): the budget, the capacity to produce certain instructional products in-house, etc.
- Information about the resources available for designing training simulator specifications: the available time and capacities of members of the design team, the methods and tools available, etc.

In a large scale design process, such as the specification of a training simulator, different parties are involved and they have to get to an agreement about the problem definition: what the problem is, what the demands are on the solution, which resources are available, etc. Subsequently, there are at least two ways to handle complexity: decompose the problem into sub-problems or design solutions on an abstract level first. The sub-problems are not unrelated and can, therefore, never be solved in isolation. Moreover, it is impossible to know beforehand which ideas will lead to a good solution and which will not, because:

- solutions have to be partially developed before they can be checked against the problem definition, and they may have to be adapted or discarded when they do not optimally fulfil the demands,
- good solutions for sub-problems may not be feasible when they are combined with solutions to other sub-problems, and
- practical factors, like changing resources or interruptions by other stakeholders, may actually change the problem definition during the design process.

This means that the design process can be only partially planned beforehand. Sometimes, designers will have to try out actions, assess the consequences and implications of those actions and then decide on further actions (Rowland, 1993). This is another reason why the design of training simulator specifications is inherently iterative: specifications for instructional products are gradually improved and refined. For a more detailed discussion of the concept of iteration see Section 3.3.3.3 and Section 5.4.1.

Apart from gathering different kinds of information and tasks that directly regard solving the design problem, two other aspects play a role: planning and managing the design process and communication. Thus, the design task has four main aspects:

1) Gathering information
Collecting information is obviously important during the problem analysis phase. Not all information will be immediately available, however, and information needs to be kept up-to-date. Possible information sources include system documentation, people (colleagues, stakeholders, experts), research literature, lessons-learned documents, etc. Other information is constructed during the design process: a draft training programme, draft specifications of instructional products, etc. All information needs to be organised and stored, and ready for review during the design process and afterwards. Information management is also important because the people collecting data may not be the ones who need to use them later on.

2) Solving the design problem
Solving ill-structured design problems involves problem analysis and problem solving, which may be a sequential or an overlapping processes (Rowland, 1993). Sub-tasks include, decomposing the problem into sub-problems, designing possible solutions for a sub-problem, selecting the most suitable alternative, integrating sub-problem solutions, conducting formative evaluations and checking a solution against the demands and constraints, etc.

3) Managing the design process
For ill-structured problems, such as design problems, there is no single solution procedure. Every design process is different. It is not possible to prescribe a fixed sequence of actions or tasks. The designers will have to organise and monitor the design process themselves, e.g., plan activities, reflect and evaluate, decide to go more into detail or to go back and review an earlier decision.

4) Communication
The design process takes place in an organisational setting. The task of designing functional specifications for training simulators is usually not the responsibility of one person, but of a team of designers. Other stakeholders may be less directly involved in the design process, but can still influence it, especially when they have power over resources. Thus, the design task also involves communication within the design team and with other stakeholders.

In Section 3.2, I have discussed descriptive research regarding the instructional design process. In the next section I will discuss prescriptive theories and models from the field of instructional design.

3.3 Prescriptive theory: supporting the design process

The design of specifications for training simulators is an ill-structured problem solving task that non-experts cannot do well without additional support. SMEs and novice designers who lack (part of) the necessary knowledge and skills need support from prescriptive theory that provides them with explicit instructional design knowledge and an explicit model of how an instructional design problem should be solved. For ill-structured problems, such as design problems, there is no algorithm that defines how to apply instructional design knowledge and guarantees a solution. There are heuristics, which are not always complete or unambiguous. Heuristics do not guarantee that a solution will be found, but they improve the chance of finding one (Mettes and Pilot, 1980). Since SMEs and novice instructional designers seem to lack the necessary skills to apply theoretical knowledge and solve problems, heuristics that make explicit how to tackle ill-structured design problems are especially important. Such heuristics can be derived from theory or deduced from the behaviour of expert instructional designers.

In this section I describe some examples of different types of prescriptive models from instructional design theory: System Theory (§ 3.3.1), Instructional Design models (§ 3.3.2), Instructional System Development models (§ 3.3.3), hybrid design and development models (§ 3.3.4) and rapid prototyping (§ 3.3.5). Some of them cover the whole development process, and not only the design phase (see § 3.1 for definitions). Since the descriptive studies described in the Section 3.2 made clear that the problem context plays an important role, I also describe acquisition procedures (§ 3.3.6) and other models and principles focusing on organisational aspects (§ 3.3.7). In the discussions with each section I discuss which aspects of the design process they do and do not address. In Section 3.3.8, I give examples of different categories of tools that are available to support the design and development process. At the end of this chapter, in Section 3.4, I will look at the prescriptive models from a different perspective and compose a list of forms of support that are or could be made available to support the design process.

Many authors explicitly warn against too high expectations of prescriptive models: they provide heuristics that cannot be executed in a linear 'cookbook' sequence. This is partly because they have been developed in and for specific organisational settings, and aim at different aspects of designing instructional products. It is also in part inherent to the instructional design task. First of all, instructional design problems are ill-structured problems (see § 3.2.2.2). Therefore, the actual process of searching for a solution is much richer and more complex than that suggested by any model (Pirolli and Greeno, 1988). Secondly, instructional design is an inherently creative task. Some people even claim that it is only 'art' (the artistic paradigm according to Visscher-Voerman, 1999, see § 3.2.2.1) It cannot be denied that the instructional design task requires both skills and creativity (Plomp and Feteris, 1991; Rowland, 1991, 1993). Spector and Song (1995) propose the view that determining which instructional strategy is appropriate for a particular kind of learning goal is more science than art, whereas the concrete implementation of that strategy may be more art than science. When it comes to the concrete elaboration of functional specifications for instructional products all models leave a lot of room for interpretation, and therefore the end results will always be influenced by the users' own preferences and experiences.

Please note that prescriptive design models do not aim at explaining or describing how designers work. They are aimed at helping designers and at improving the product of the design process (Braha and Maimon, 1997). Finally, using a prescriptive model does not guarantee a good product. Boring instruction can result from the use of any model. On the other hand, experienced and talented instructional designers can come up with effective (specifications for) instructional products with or without the use of prescriptive models (Dick, 1995).

3.3.1 System Theory

3.3.1.1 The perspective of System Theory

According to the System Theory any functioning entity can be viewed as a system of inter-related components and defined in terms of its objectives (Patrick, 1992). Each system gets input, it performs some tasks and activities, and delivers some output.

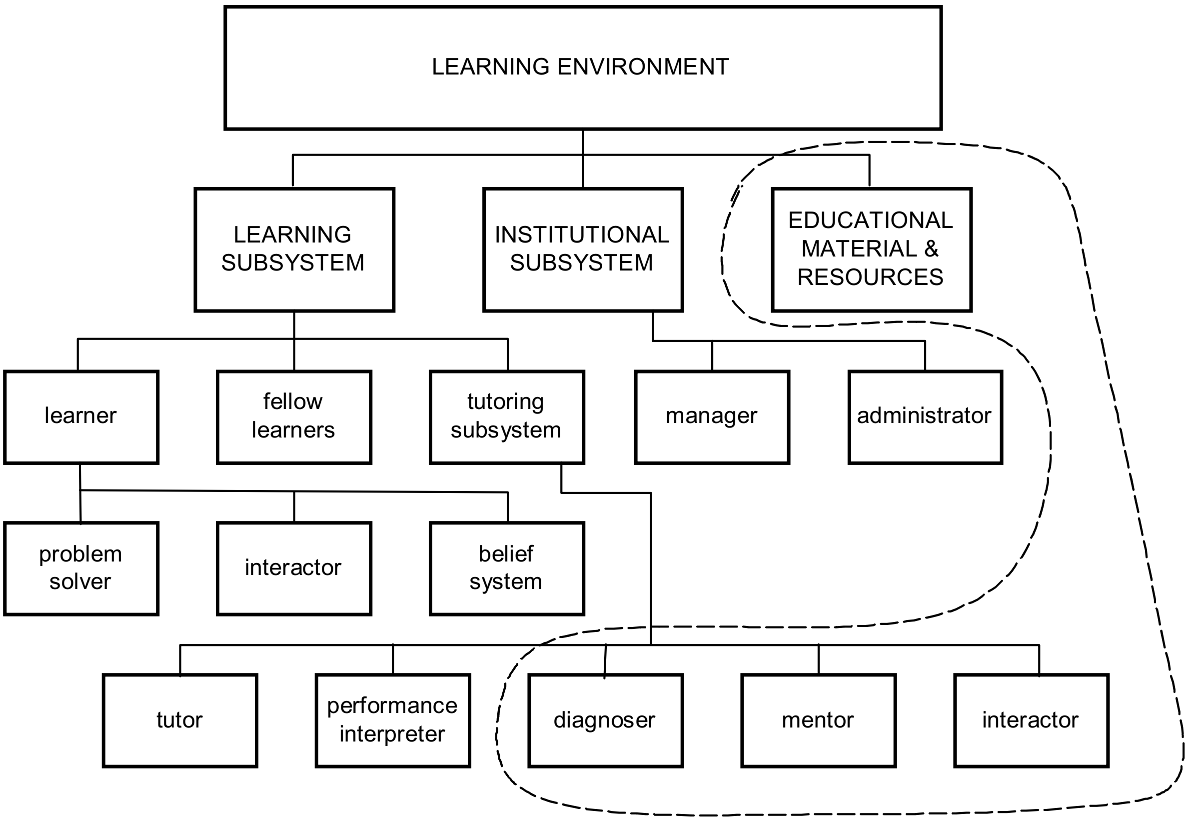


Figure 3.2: The learning environment decomposed (from Barnard and Sandberg, 1994, p. 33, used with permission of the author), with added dotted line that gives an example of what functions a training simulator could fulfil.

From this perspective, a learning environment can be seen as a system as well, with learners as input and trained personnel as output. A learning environment is a mixture of human and non-human elements: teachers, learners, simulators, books, classrooms, etc. Barnard and Sandberg (1994) propose a functional decomposition of learning environments that is represented in Figure 3.2. The different functions can be fulfilled by human beings or instructional products (except for the learner who is always human). In fact, one actor can take on a combination of functions. The dotted line gives an example of functions that a training simulator could fulfil.

When the function of components and their relationships are clearly specified, the components can be designed separately (Simon, 1969). In this context, this means that a training simulator can be designed separately from the rest of the learning environment when it is clear what will have to be trained with the simulator, i.e. what the input level of trainees will be and what the required output level is at the end of simulator training. And the training simulator itself can be seen as a system consisting of inter-related modules. When it is possible to identify the requirements for separate modules, they can also be designed separately. Figure 3.3 presents a decomposition of training simulators proposed in Farmer et al. (1999). Korteling, Helsdingen, and von Baeyer (2000) have further defined which aspects of the different components should be taken into account.

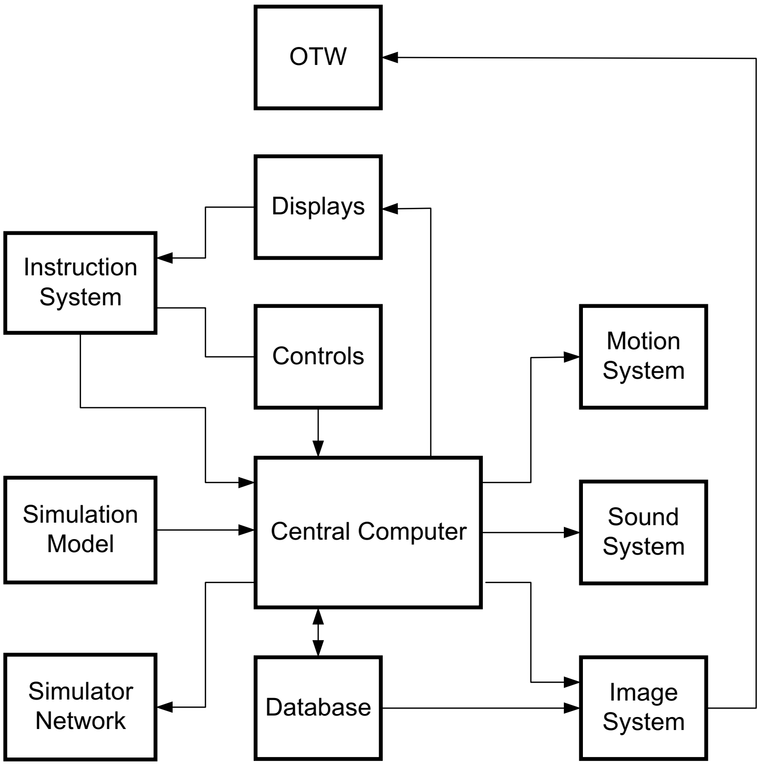


Figure 3.3: Schematic overview of a comprehensive training simulator (from Farmer et al., 1999, p.108, used with permission of the publisher), N.B.: OTW is Out of The Window view.

The more recent standardisation initiatives can support the application of System Theory by standardising terminology and implementation of instructional products. Examples are the LTSA model (Draft Standard Learning Technology Systems Architecture, 2001) and the ADL-SCORM model (Advanced Distributed Learning Initiative, 2001). ADL is developing a common technical framework for computer and Web-based learning that will foster the creation of reusable learning content as 'instructional objects'. The SCORM (Shareable Content Object Reference Model) standard is intended to provide the technical means for content objects to be easily shared across multiple learning delivery environments.

3.3.1.2 Discussion

System Theory proposes possible ways to decompose the task of designing a learning environment into designing different instructional products and parts of instructional products. This can help designers to manage the design process and to ensure that all relevant parts of the instructional products are addressed. The use of standard terminology describing instructional products and their components can facilitate the communication within the design team and between the designers and other stakeholders.

Descriptive research made clear that professional instructional designers decompose design tasks into modules (see § 3.2.2). However, these modules are not independent, they are 'leaky' (Goel and Pirolli, 1989, 1992). In the case of simulator specifications there is, for example, often a trade-off between the price of the training simulator and the extra expense of having to train some training goals on the job with the operational system. Only during the design process will it become clear which solution is most effective and efficient within the given financial, technical and other constraints. Thus, other parts of the learning environment have to be continuously taken into account and this means that the decomposition can change during the design process. The risk of decomposing design problems into sub-problems is that the parts rather than the whole are optimised (Rowland, 1993).

3.3.2 Instructional design models

Like all design sciences, the field of instructional design has developed rules for good design that roughly describe the boundaries of the 'design space'. There are different communities of instructional designers that have developed -and continue to develop- their own rules of good instructional design (Dijkstra, 2001b). Most of them are based on System Theory (see § 3.3.1), on theories about how people learn or teach (descriptive) and/or on theories about how people should learn or teach in order to be most effective and efficient (prescriptive). Different terms are used for these models in the literature and they are used inconsistently. Reigeluth (1983, p. 24) makes a distinction between:
- instructional design models that indicate what the instruction should be like, and
- instructional development models that indicate how to make it that way.

Following this approach I discuss instructional design models in this section and instructional development models in Section 3.3.3. Note that the term instructional design model, as used here, refers specifically to determining what the instruction and the instructional products should be like: the sequence of instructional topics and activities, the nature of interactive lesson elements, and motivational plans (Gustafson, 1991). A large number of design models is reported in the literature. I present some typical examples and discuss how they could support the design of functional specifications for training simulators[3].

3.3.2.1 Examples of design models

Instructional Design (ID) models are based on the assumption that it is possible to predict which instructional methods will lead to the desired outcomes under which conditions. Note that the term instructional method is used in a broad way here, and includes not only teacher-led methods but also learner-centred approaches. Thus, ID models prescribe what instruction should be like and give indications for suitable instructional products.

Some ID models are centred around learning goals. Gagné (1985), for example, categorises learning outcomes into five categories of 'learned capabilities': intellectual skills, cognitive strategies, verbal information, attitudes and motor skills. He claims that each of these categories requires different kinds of learning conditions. In addition, Gagné outlines instructional events that should provide the necessary conditions for learning to occur. In his Component Display Theory, Merrill (1983, 1987) uses a taxonomy of learning goals with two dimensions: content (facts, concepts, procedures, and principles) and performance (remember, use, generalise). Merrill focuses on the presentation of lesson content and proposes different combinations of primary and secondary presentation forms for the different kinds of learning goals.

Other ID models are based on theories of learning. According to Anderson's ACT* theory (Anderson, 1983), for example, learning takes place in three phases: acquisition of declarative knowledge, proceduralisation of declarative knowledge and compilation of procedures to speed up performance. ACT* has been the basis for several computer-based instructional

[3] Kearsley (2003) provides an overview of a large number of theories about instruction and learning in the Theory Into Practice (TIP) database. Another overview with many links to other web sites is provided by Ryder (Ryder, 2003). Alessi and Trollip (2001) have collected design heuristics for different kinds of instructional products. Recent design models can be found in, for example, Reigeluth (1999), Tennyson, Schott, Seel, and Dijkstra (1997) and Dijkstra, Seel, Schott, and Tennyson (1997).

products that all follow the same ID model: a structured set of problems is provided, adapted to the learning process of the learners. The proceduralisation of knowledge is supported by minimising the working memory load and solving the problems step-by-step (e.g., Anderson, 1987; Anderson, Boyle, Corbett, and Lewis, 1990;). Anderson and his colleagues argue that errors should be avoided, since they lead to faulty proceduralisations that are a waste of time and difficult to unlearn. Therefore, feedback on errors is given immediately (i.e. the model tracing approach). Based on a different learning theory Schank and his colleagues claim that generating expectations and encountering failures in the pursuit of a goal are essential for the learning process (Schank, Fano, Bell, and Jona, 1994; Schank, Berman, and Macpherson, 1999). They plead for learning environments where learners become active participants in a scenario where they can learn skills by practising them in a situation where they need those skills to achieve a motivating goal.

ID models differ in what they cover and the level of design they focus on. Merrill's Component Display Theory, for example, focuses on the details of the specific presentation forms for cognitive skills; it does not cover attitudes and psychomotor skills. Reigeluth's Elaboration Theory (Reigeluth, 1987, 1983; Reigeluth and Stein, 1983), on the other hand, focuses on higher level components of instruction: the selection of topics, sequencing of presentations, synthesising and summarising topics. In a way, these models can be seen as complementary (Snelbecker, 1983).

3.3.2.2 Discussion

ID models help designers to solve the design problem by prescribing suitable instructional methods, i.e. they specify what learning environments and instructional products should look like in certain conditions. It is generally accepted that factors such as the type of learning goals, the type of learners, the organisational setting and the instructional strategy should influence the design of learning environments. However, many ID models are available in the instructional design literature and they provide different and sometimes contradicting advice. Choosing suitable models is not easy. The problem is that ID models focus on one or a few factors and do not take interactions between different factors into account. They are often developed for a particular setting and it is not clear whether they are valid in other settings as well. Finally, ID models are often based on learning paradigms or assumptions that are not always made explicit.

The choice of ID models can have large consequences for simulator specifications. A model tracing approach, for example, makes it possible to foresee exactly which actions the trainees will be able (i.e. allowed) to execute, but it also requires an immediate and preferably automated feedback mechanism. Following a discovery approach, on the other hand, means that the simulator should allow a wider, partly non-predictable range of actions by trainees. Giving feedback will be more complicated and may have to be done by human instructors, who will need facilities to monitor trainees for this purpose. Simulators are often used for practising tasks with a large procedural component and for training of adult trainees in a job-oriented setting (see for examples § 2.3.1.1). This would suggest that ID models such as the model tracing approach would be more useful than discovery learning. This is a dangerous conclusion, however. Alternative approaches may be equally valid, and the tasks may well have other aspects for which a model tracing approach is not suitable at all. A careful analysis will give more detailed information about the trainees and the training goals for a specific course. In the end, however, the choice of ID models depends on the capabilities and preferences of the designer. Moreover, the application of ID models to any concrete situation requires a considerable amount of skill and interpretation. Maybe that is why they do not seem to be applied very much in simulator training: the results of the field study presented in Chapter 2 show little variety in instructional methods that are used and there is no evidence of the conscious application of ID models (see § 2.3.2).

3.3.3 Instructional development models

In Section 3.3.2, I described some examples of instructional design models that indicate what the instruction and instructional products should be like. In this section I give some examples of instructional development models that indicate how to make it that way. Development models have been reported in the literature under different names, such as Instructional System Design or Instructional System Development (both abbreviated as ISD), or System Approach to Training (SAT). I will use the abbreviation ISD in this thesis.

Instructional development models, regardless of the form a model may take, prescribe a process of systematically designing, sequencing, implementing, evaluating and constantly monitoring instruction with the intent of improving its quality and effectiveness, and thereby enhancing learning (Gustafson, 1991). Note that in this definition the term development is used in a broad way and covers the whole development process. Others use the term development for one phase in the process, that I refer to as the production of instructional products (see also § 3.1). The basic claim of development models is that instructional products will be better when they are developed in a systematic way. ISD models are based on System Theory (see § 3.3.1) and specify a number of steps beginning with an analysis of needs and goals and ending with an evaluated system of instruction that demonstrably succeeds in meeting accepted goals (Gagné, Briggs, and Wager, 1992). ISD models are based on the ideal that specifications for instructional products should be based on what learners need to learn (Analysis), and how they can learn it most effectively and efficiently (Design), as depicted in Figure 1.1. The development process is typically divided into five phases (e.g., Loube, 2001; Tennyson, 1993):

1. Analysis: analysis of the instructional problem/need.
2. Design: design of specifications to solve the problem.
3. Production: production of the instructional products (often called development).
4. Implementation: delivery of instruction and/or training.
5. Evaluation: evaluation and maintenance of instructional products.

Many different ISD models are described in the literature, some developed to train instructional designers (e.g., the Dick and Carey model, Dick, 1997; Dick and Carey, 1996), others developed for school settings or military organisations. Commercial organisations have also developed models adapted to their own situation, but these are less often published. At AIRBUS, for example, an in-house development model called ADOPT (AIRBUS Design and Operational Philosophy in Training) is used for the specification of simulators and other instructional products. ADOPT guides the development of instructional products step-by-step (Barnard, Boy, Tremaud, Payeur, and Fauré, 2002). For my own research I have used an ISD model specifically oriented towards the specification of training simulators, i.e. the MASTER method (Farmer et al., 1999, see Chapter 4). As examples, I describe two famous, early development models below[4]: the IPISD model that was developed in a military setting and the ISD model of Gagné and his colleagues that is used a lot in civil education. Subsequently, the issue of iteration in development models is briefly discussed.

3.3.3.1 IPISD

The Interservice Procedures for Instructional Systems Development (IPISD) is one of the earlier ISD models. It is depicted in Figure 3.4:

[4] Many other development models are available in the literature. See for overviews, for example, Gustafson (1991), Andrews and Goodson (1980), Hannum (2001), and the special issue of Instructional Science edited by Dijkstra (2001a). More recent descriptions of SAT as used in military settings can be found in e.g. Swift, Martindill, and Allender (1998) and Perez and Neiderman (1993).

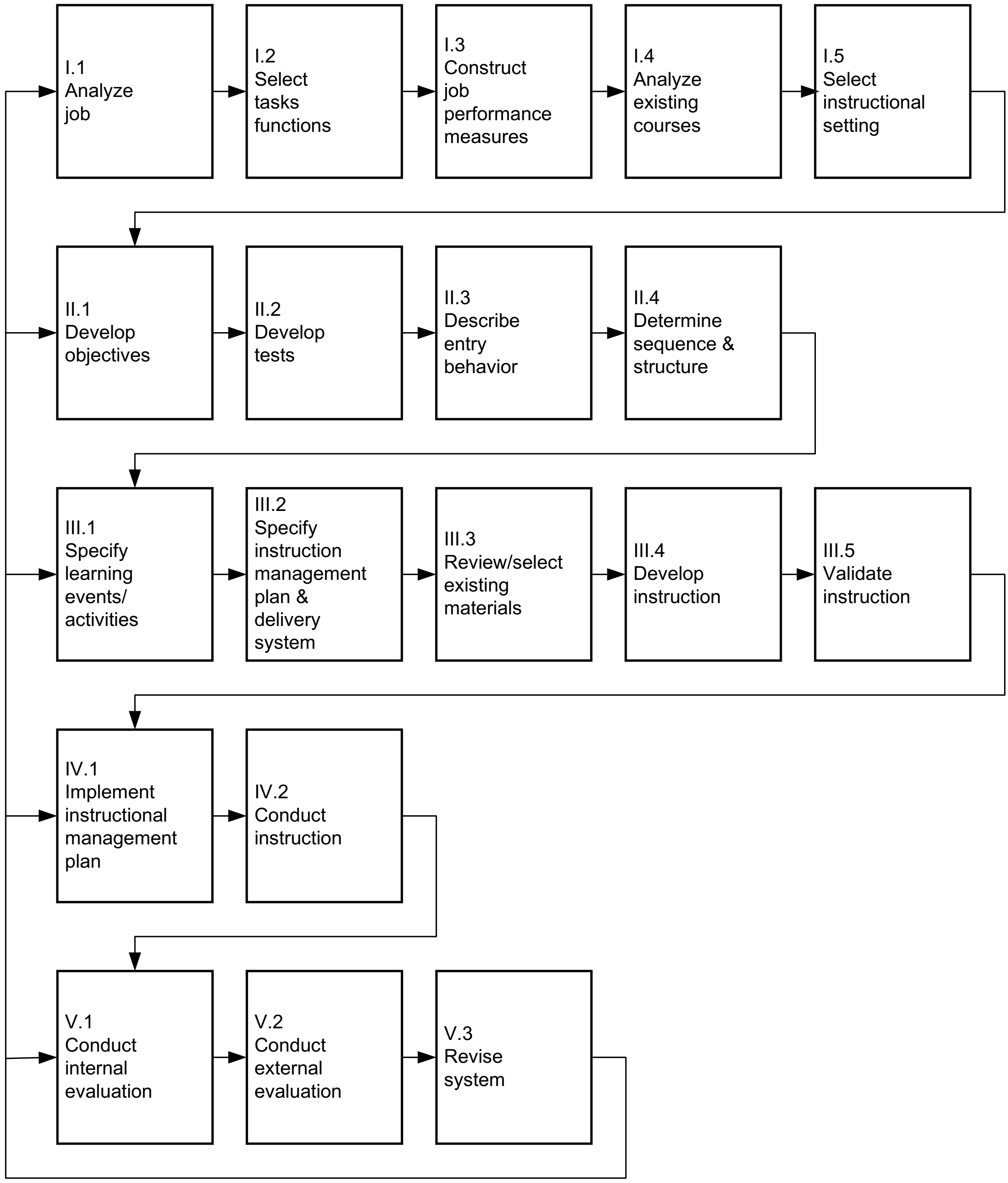


Figure 3.4: The IPISD model (after Branson et al., 1975, p. i).

It is, as the name suggests, a joint effort of the U.S. military services. The IPISD model is elaborated into great detail and reported in a five volume set (Branson et al., 1975). At its simplest level it has the five phases described above: analysis, design, production, implementation and evaluation. These five phases are divided into 20 steps, which can be further divided into hundreds of sub-steps. I only describe the second level here (see Figure 3.4).

In the first phase the future job of the learners is analysed, the tasks or functions that they need to be educated for are selected, and performance measures for these tasks are developed. Subsequently, existing courses are analysed to determine whether they can be (partly) reused and finally an instructional setting is selected from a list of standard settings used by the American Armed Forces at the time. In the second phase the learning tasks are further analysed and learning goals (objectives) are formulated, tests are developed, the entry level of learners is described, and the sequence and structure of instruction is determined. In the third phase learning events and activities are planned, types of instructional products are selected and an instructional management plan is constructed. Existing instructional materials are examined to see whether they are suitable and, finally, instructional products are produced and validated until the learners meet the learning goals. In the fourth phase instruction and training are implemented and conducted. And in the fifth phase data on the effectiveness of instruction and training (internal evaluation) and the job performance in the field (external evaluation) are collected. The results of evaluation can lead to revisions in any of the phases. In Branson et al. (1975) each step is defined in terms of the required inputs, the procedures that must or can be used, the outputs and the resulting management decisions.

3.3.3.2 The ISD model of Gagné and colleagues

The ISD of Gagné and his colleagues was originally developed in the seventies and has been further elaborated over the years. The description in this section is taken from Gagné et al. (1992). In accordance with the Dick and Carey model (Dick and Carey, 1996), Gagné and his colleagues decompose the development of instructional products into nine steps:

1. Instructional Goals
2. Instructional Analysis
3. Entry Behaviours and Learner Characteristics
4. Performance Objectives
5. Criterion-referenced Test Items
6. Instructional Strategy
7. Instructional Materials
8. Formative Evaluation
9. Summative Evaluation

In the first step the goals of instruction are established: the gap between the desired state of affairs and the present state is determined and the available resources and constraints are analysed. The second and third step, Instructional Analysis and Entry Behaviours and Learner Characteristics, can be executed in parallel. The Instructional Analysis determines which skills are involved in reaching a goal. Different types of analysis can be used for this purpose: a task or procedural analysis produces a list of steps and the skills used at each step of a procedure. An information-processing analysis reveals the mental operations used by a person who has learned a complex skill and a learning-task analysis reveals the enabling objectives. In the step Entry Behaviours and Learner Characteristics designers analyse which of the required skills are already mastered and which abilities and personality traits should influence the design. In the fourth step the learning goals are translated into performance objectives that are sufficiently specific and detailed to show progress toward the learning goals in terms of observable, measurable behaviour. From the classification into categories of learned capabilities (see § 3.3.2.1), the necessary learning conditions can be inferred and the sequence of instructional activities can be planned. Instructional planning proceeds to the design of units of instruction that are smaller in scope and thus more detailed in character. Subsequently, procedures for assessment are developed in the fifth step. In the sixth step lessons are planned using Gagné's nine instructional events (Gagné, 1985) and an appropriate (combination of) instructional products is chosen. Note that this can take the form of teacher-led instruction or learner-centred, learner-paced lessons. This step also results in the specifications for the necessary instructional products, which are then developed in the seventh step. Preliminary versions are tried out, evaluated and revised in order to make them as effective as possible (formative evaluation). Finally, in the last step, a formal summative evaluation is executed when the learning environment is no longer undergoing point-by-point revision. A more elaborate 14-step model is proposed for the design of entire curricula (Gagné et al., 1992, p. 31).

3.3.3.3 Iteration in development models

Most ISD are presented as a linear sequence of steps with explicit feedback loops only after formative or summative evaluation. However, it is known from descriptive studies that expert instructional designers adapt the development process to the situation that they are working in, that they switch back and forth between 'leaky' modules, gradually develop solutions in more detail and keep on comparing parts of or partly developed solutions to the demands (see § 3.2.2). Indeed, as stated before, development models should not be seen as algorithms: they are a set of heuristics which guide the process of instructional development (Dijkstra, 1997). It could be argued that they propose an ideal process, as if the designers work in a social vacuum in which time and money constraints are not present (Visscher-Voerman, 1999).

Rowland (1991) suggests that ISD models can be interpreted as a set of decision points or a set of categories of information to be sought, rather than a strict procedure. Along the same lines, Rowland and Adams (1999) propose that the linear looking models may represent the sequence of completion rather than the activities themselves. De Hoog, de Jong, and de Vries (1994) propose a spiral model where iteration is controlled by quality control. In each development cycle a component is developed further (Specificity), checked against what was specified (Compliance), checked for consistency with previous and foreseen states (Quality) and checked for coherence with other fully or partially developed components (Integration), as depicted in Figure 3.5:

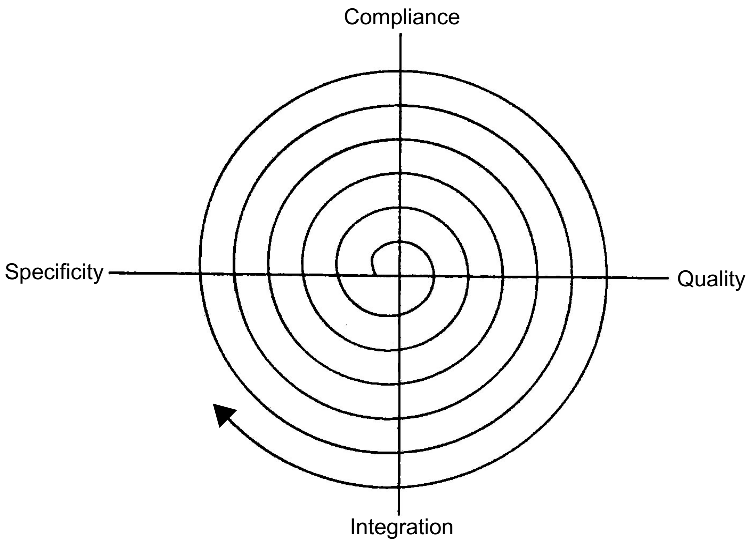


Figure 3.5: A spiral for partial product development (From de Hoog et al., 1994, p. 53, used with permission of the author).

Several authors suggest that ISD models should be adaptable. Designers should be able to decide to include or exclude steps depending on the problem situation, or to execute activities simultaneously rather than sequentially (e.g., Plomp, 1992; Tessmer and Wedman, 1995, 1990). Tennyson and his colleagues propose a fourth generation of instructional development models where the five phases cannot be strictly separated, they overlap each other as depicted in Figure 3.6 (Gagné, Tennyson, and Gettman, 1991; Tennyson, 1995). Depending on the situation the beginning point of development activities can be anywhere. Using ISD models in such a flexible way, however, requires a more explicit analysis of the project's context as part of the front-end analysis (a 'contextual analysis', see Tessmer and Wedman, 1995) and it requires expertise from the instructional designers (Tennyson, 2001). Tennyson and his colleagues advocate the development of intelligent tools that should be able to diagnose the user's situation and level of expertise and give appropriate advice or coaching in order to maintain a systematic development process (Tennyson and Breuer, 1994; Tennyson and

Elmore, 1995). For a more detailed discussion of the concept of iteration in the context of training simulator specification see Section 5.4.

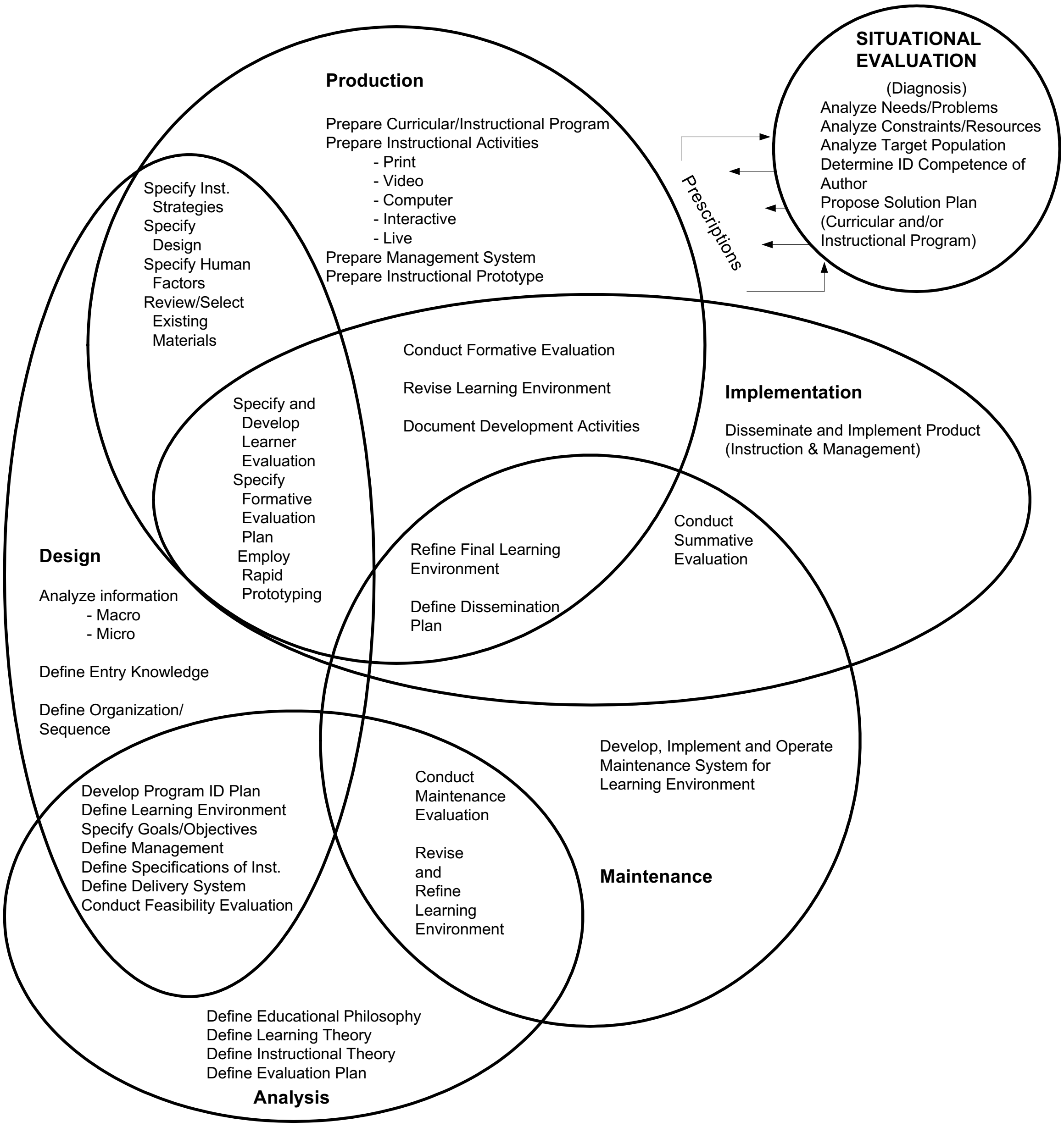


Figure 3.6: Fourth generation instructional system development, ISD4 (from Tennyson, 1995, p. 35, used with the permission of the publisher and the author).

3.3.3.4 Discussion

Gagné et al. (1992, p. xii) describe the purpose of their model as follows:

> "...describing a rationally consistent basis for instructional design. The procedures we suggest are more accurately viewed as 'what to do' than specifically 'how to do it'......The designer must ultimately deal with details of print, pictures, and sound, and these elements have their own technologies. We trust that the instructional designer who follows the principles described here can be assured that the details of instruction, however arrived at, will have a sound foundation in research and theory."

Saying this, Gagné et al. (1992) point to both strong and weak points of instructional development models: they support the management of the design process by prescribing which activities should be executed to develop a valid and coherent learning environment. However, they do not specify how these activities should be executed, nor when a step is completed or what designers should do when they encounter problems (Nelson et al., 1988; Perez and Neiderman, 1993). Part of this strategic 'How to do it' knowledge is captured in ID models, methods for knowledge engineering and guidelines about more specific issues, such as feedback, assessment, retention and transfer. This knowledge is, however, dispersed and not integrated with the activities prescribed by the development models. It is easy to see how guidelines about, for example, retention and transfer can be used to evaluate existing instructional products, but it is not always evident how and when this knowledge should be used during the specification of instructional products. Instructional development models do, however, make sure that important issues are addressed and provide a framework for gathering and organising information (Nelson et al., 1988). They also provide shared terminology that can be used to communicate with the different parties involved in the development process.

Many ISD models are available. They differ in scope: some do not cover all phases, while others are developed for specific organisational settings or based on specific instructional theories. Therefore, it is not clear under which conditions the different models are valid. Another problem is that although more recent views propose an iterative development process, they do not specify how users (or intelligent support tools) can decide what activity should occur next. The nature of iterations, such as what causes them to be initiated and how they occur, is not clear. Instructional designers still need problem solving skills to plan, monitor and adapt the development process, the kind of expertise that novice instructional designers and SMEs usually do not have (see § 3.2.2.3).

In some ways, the development of training simulators differs from the development of other instructional products. This somewhat limits the applicability of standard ISD models that, for instance, explicitly address neither decisions about what to simulate and what to omit (fidelity issues, see § 1.2.1 and § 2.1), nor the specification of typical training simulator facilities such as instructor support and scenario editing facilities (see § 1.3.3). The choice of instructional products is very important, especially when expensive facilities like visual systems or moving bases are involved. Alternative options have to be considered, and some amount of iteration and negotiation will be necessary to get to an acceptable set of specifications. That immediately points to another difference: simulators are technically complex systems and therefore technical aspects have a large influence on the specifications, much more than for other instructional products such as books, CBT or classroom lessons (see § 1.3.3). The production phase is often contracted out to industry, i.e. not controlled by the people who were responsible for the analysis and design phase. This makes (formative) evaluation and improvement of the design difficult: once the contract is assigned, the specifications cannot be changed. Thus, iteration in the development process and the freedom to complete steps in a different order are limited. These simulator specific issues are addressed by the MASTER method (Farmer et al., 1999), which I have used for the research described in the second half of this thesis (see Chapter 4 and further).

3.3.4 Hybrid design and development models

In practice, the division between design models prescribing what the instruction should be like, and development models prescribing how to make it that way is not always so strict. Many authors of development models have included particular design models in their work (e.g., Kessels and Smit, 1989; Romiszowski, 1984, 1986; and the work on ISD and on conditions of learning by Gagné and his colleagues reported in § 3.3.2 and § 3.3.3). On the other hand, many ID models contain, explicit or implicit, prescriptions for how instructional products should be developed.

More recently, many authors have given up the idea of developing generic instructional design or development models that are always applicable and have focused on far more specific models that combine design and development aspects, but for one specific combination of conditions, e.g., for one type of learning goals, for a specific kind of target learners, for a specific kind of instructional product and/or for a specific instructional setting. Hybrid design and development models provide partly developed 'standard' solutions to specific categories of design problems. Descriptive research showed that expert instructional designers also compare the current problem to problems and solutions they have encountered in the past (see § 3.2.2.3). One could say that hybrid design and development models try to make this kind of 'knowledge from experience' explicit and available to non-experts. From this point of view, they can be compared to the collections of successful 'design patterns' that are used in software engineering (Gamma, Helm, Johnson, and Vlissides, 1995). Hybrid models can be models that have proven to work well for a certain domain (see, e.g., Halff, 1993), or models derived from instructional design and development theory (see below). I present two examples of hybrid design and development models below.

3.3.4.1 The Four-Component Instructional Design model

The Four-Component Instructional Design model (4C/ID model, van Merriënboer, 1997) is meant to design learning environments for learning complex cognitive skills. It was initially developed for industrial and vocational settings, such as management skills in the chemical industry, software engineering skills or air traffic control skills. The 4C/ID model stresses the importance of integrated performance objectives and focuses on real-life tasks as the driving force for learning (van Merriënboer and de Croock, 2002). The incorporated ID model is based on learning theory and claims that complex cognitive skills require learning of both procedural and declarative knowledge. They consist of a number of highly-interrelated constituent skills, of which some are performed as well-learned procedures while others rely on a good understanding of the domain and the ability to (re)formulate and solve problems. Van Merriënboer refers to these as recurrent skills and non-recurrent skills.

According to the 4C/ID model recurrent skills are learned by rule automation and non-recurrent skills are learned by the acquisition of cognitive schemata. The proposed ID model consists of four components (van Merriënboer, 1997; van Merriënboer, Clark, and de Croock, 2002):

- Component I (Induction):	During whole task practice cognitive schemata are acquired by a controlled cognitive process called induction.
- Component C (Compilation):	Recurrent aspects are acquired by an elementary cognitive process called compilation. They have to be practised repeatedly to reach the required level of performance. If the learning tasks do not offer enough practice opportunities additional part-task practice is necessary.
- Component R (Restricted Encoding):	The information that is relevant to recurrent aspects is presented just in time to support the automation of rules by a cognitive process called restricted encoding.
- Component E (Elaboration):	The information that is relevant to non-recurrent aspects is presented before the start of exercises in such a way that it can be elaborately encoded or connected to already existing declarative knowledge (in schemata) by a controlled process called elaboration.

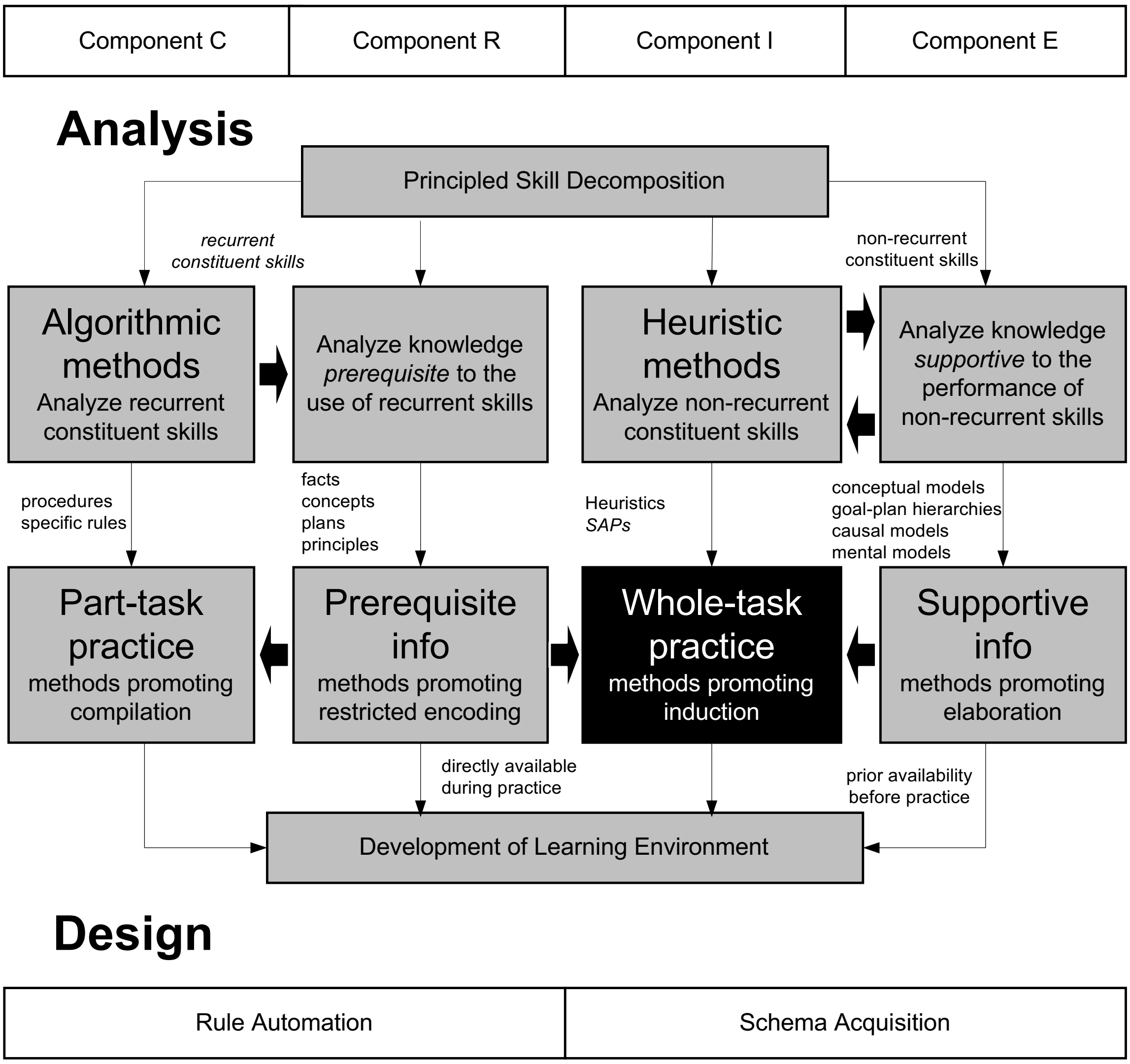


Figure 3.7: An overview of the Four-Component Instructional Design Model (from van Merriënboer, 1997, p. 8, used with permission of the publisher).

In the 4C/ID model the backbone of training are learning tasks that allow learners to practice the whole task or meaningful clusters of constituent skills that are treated as whole tasks. Learning tasks of equal difficulty are grouped in a task class. Each task class is more complex than the previous one. At the beginning of a new task class much guidance is offered. The guidance gradually disappears towards the end of the task class.

The development component of the 4C/ID Model consists of four layers (see Figure 3.7). The first layer covers the decomposition of complex cognitive skills into a hierarchy of recurrent and non-recurrent constituent skills (principled skill decomposition). In the second layer constituent skills, their relationships and the underlying knowledge structure are further analysed. In the third layer instructional methods for the design of whole-task and part-task practice, and for the different types of information presentation before and during practice, are selected and specified. And in the fourth layer a detailed blueprint for the learning environment is designed. In later publications this development model has been further elaborated into a ten step model (van Merriënboer and de Croock, 2002). Note that this model only covers the first two stages of ISD, Analysis and Design. Van Merriënboer (1997) recommends users to apply it in conjunction with an ISD model (see § 3.3.3).

3.3.4.2 ID2

Merrill and his colleagues aim for a set of tools that provide both authoring and delivery environments. They have called their approach Second Generation Instructional Design or ID2. The aim of ID2 is to develop instructional algorithms that can be written once and reused (see ID2, 2003; and e.g., Merrill and the ID2 Research Group, 1996; Merrill, Li, and Jones, 1992). According to Merrill and his colleagues one of the main problems of traditional ID and ISD models is the lack of integration between the different development phases: each phase is performed independently of other phases, using separate tools and separate knowledge representations. To overcome this limitation Merrill and his colleagues propose a single representation of the data that can be used throughout the development process. Furthermore, they try to provide design prescriptions for teaching an integrated set of knowledge and skills rather than individual learning goals (as many traditional ID models do).

ID2 focuses on developing CBT. In this context, instruction is defined as a sequence of interactions between the instructional system and the learner. The term instructional transaction is used to describe patterns of learner interactions. Merrill and his colleagues identify five main classes of tasks, that can be further subdivided: denote, manifest, execute, discover and design. The execution of each class of tasks requires specific types of knowledge and skills and can, therefore, be taught with predefined types of instructional transactions. The concrete implementation of these transactions can vary widely depending on the nature of the tasks to be taught, the target learners and the learning environment.

A set of transactions can be programmed into a so-called transaction shell, a piece of software that contains the empty structure of a suitable transaction for a particular type of knowledge and/or skill. With such a transaction shell a domain expert with little or no instructional experience should be able to make CBT. First, the authoring component helps the user to define the domain representation. Because the system knows the structure of the knowledge and skills for a specific class of tasks, it can simply prompt the user to identify objects, slots and links. It also prompts users to provide values for a wide range of instructional parameters regarding the learning task and the required outcome level, the learner population and the learning environment. Based on this information the system recommends instructional strategies, and in turn these strategies prescribe instructional transactions. The instructional delivery component then takes care of the actual interactions with the learner. Ideally, the transaction shell should also include an Intelligent Advisor that customises the delivery of instruction in real time, based on the performance of a particular learner. The first transaction shells that have been implemented concern relatively straightforward task categories, such as the 'identify' transaction to teach the name, location and description of entities. They have been evaluated successfully (Cline and Merrill, 1995; Hsieh, Halff and Redfield, 1999; Spector and Muraida, 1991, 1997).

3.3.4.3 Discussion

Hybrid models combine the 'what to do' knowledge from ISD models (see § 3.3.3) with the 'how to do it' knowledge from ID models (see § 3.3.2). The advantage is that they describe what designers should do in more detail, and even take some of the decisions -such as the choice of ID models- out of their hands. However, some hybrid models, such as the ID2 model, imply the choice for a particular instructional product and are, therefore, not applicable to the design of simulator specifications. Other models focus on a specific type of learning goals in specific conditions. When the conditions apply, they can be used to quickly and easily elaborate a design for specific domains or for a subset of learning goals. For the remaining parts other, more generic ID and ISD models will have to be used.

3.3.5 Prototyping approach

Traditional ISD models describe the development of instructional products as a sequence of steps. Even though many authors stress that this is not a linear process, the basic assumption remains that the different steps are executed one by one and in a particular sequence (possibly with iterations). Explicit validation, in the form of formative and/or summative evaluations with future learners, only comes in at the end of the process (Goodyear, 1997). This is risky, since the adaptation or redesign of the instructional product might then be very expensive or even impossible. Since design problems are often ill-structured, some authors claim that it is impossible to get a clear view on the goals and constraints before the design process starts. They propose to follow a more pragmatic approach of repeated try-out and revision of design prototypes. Pragmatic approaches have been observed in instructional design practice (Visscher-Voerman, 1999, see also 3.2.2.1). An example is the rapid prototyping approach.

3.3.5.1 Rapid prototyping for instructional design

Tripp and Bichelmeyer (1990) propose to apply the rapid prototyping approach that is frequently used in software engineering to the development of instructional products. Rapid prototyping for instructional products begins with the analysis of needs and content and a statement of tentative objectives. The statement of objectives is at this stage not more than the definition of a plan that serves to communicate to everyone involved the purpose of instruction and to delineate tasks that the learner will pursue. Rapid prototyping then continues with the parallel processes of design, construction and utilisation. It is assumed that full understanding of needs, content, and objectives is a result of the design process and not an input into it. Figure 3.8 represents the events that occur during rapid prototyping for the development of instructional products. The overlapping boxes are meant to represent the fact that the various processes do not occur in a linear fashion.

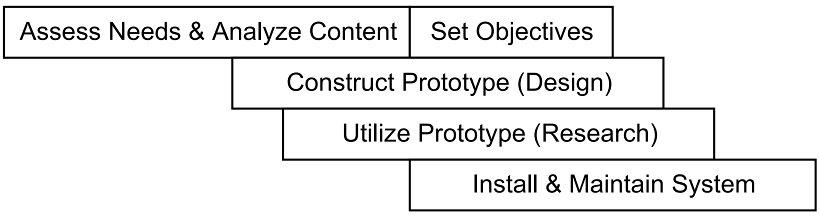


Figure 3.8: The Rapid prototyping ISD Model (after Tripp and Bichelmeyer, 1990, p. 36).

A crucial part of the prototyping process is the interaction with future users, both to clarify the problem at hand and the characteristics of the potential solution. For instructional products the users are the (future) learners. When learners use the prototype, the designer observes them, measures results and asks questions to discover strengths and weaknesses of the prototype. The discovery of problems results in the modification of the development objectives or the creation of new ones. Then the rapid prototyping process begins all over again. Thus, the development process is not strictly planned beforehand, but guided by the circumstances and the input from users (and possibly other stakeholders). Tessmer and Wedman (1995) advice to always produce at least two competing prototypes to help clarify design issues and alternatives, and to avoid the danger of committing to one prototype's features simply because it has already been created.

In software engineering the movement away from systematic methods is blooming. Other so-called agile methods have been developed (see Fowler, 2003). These methods stress frequent iteration and constant adaptation of requirements during design and development. Extreme Programming (XP), for example, proposes to involve users heavily in the design and development process (see Wells, 2003). In XP the development process relies on the elaboration of a simple base system in short iteration cycles. The future user is asked to write stories that clarify which features they expect from the future software package. The designers decide which tasks have to be completed to implement each of these features and negotiate with the user which one(s) they will develop within the next cycle. All design is centred around the current iteration; no design is done for anticipated future needs. Furthermore, XP emphasises the role of testing and the continuous involvement of users in the design and development process (Beck, 1999; or Extreme Programming, 2003). I am not aware of cases where these methods have been used for the design of specifications for training simulators or other instructional products, but several authors do plead for mixed design teams with representatives of users and industry working together on specifications (e.g., Harvey, 1998; McClure, 1998; Sommerlad, Danau, and Hendrikse, 1995; Toomer, Selvy, Howard, and Davies, 1998).

3.3.5.2 Discussion

The (rapid) prototyping approaches are best suited when the demands on the future product are uncertain or volatile, for example when a product is made for a new and unfamiliar context, or when clients are unable to formulate exact wishes (Tripp and Bichelmeyer, 1990). They require responsible and motivated designers and future users that are willing to get heavily involved in the development process (Fowler, 2003). Iteration is the central issue: as early as possible users try out proposed solutions, discover the problem areas, provide input, etc. (Nieveen, 1999). From the beginning, the development process focuses on an instructional product that is easy to use and suitable for the target learners. When prototypes of alternative solutions are developed simultaneously, their advantages and disadvantages can be evaluated and weighed. The rapid prototyping approach fits better than other approaches for those who stress the artistic side of design, and describe the design as driven by recognition of opportunities and carried out in iterative cycles (Rowland, 1993).

A potential problem is that instructional designers often have very limited access to actual target learners, and that their contact person often has a poor understanding of the actual needs of the users whom he or she represents (Goodyear, 1997). Furthermore, descriptive research has shown that instructional designers are not inclined to discard ideas or partially developed solutions (see § 3.2.2.2). The danger is that designers just incrementally develop their first idea and never reconsider their initial decisions that were based on a quick and incomplete analysis of the design problem. Tripp and Bichelmeyer (1990) warn that the rapid prototyping approach has a tendency to encourage informal design methods, which may introduce more problems then they eliminate.

Rapid prototyping is an alternative way to organise and manage the development process and to communicate with users and other stakeholders, but it is only possible when prototypes of instructional products can be produced easily and quickly and can be adapted at minimal costs. This is not true for training simulators, which usually contain many specific (hardware) components. Moreover, the construction of training simulators is often contracted out to industry partners, which means that the designers of the functional specifications have no control over the construction of prototypes. There are, however, several ideas that can be applied during the design of training simulator specifications. First of all, designers can involve users early in the design process, either as team members or as evaluators of draft specifications. This would cause iterations early on, during the design phase. Tessmer and Wedman (1995) propose to use 'instructional scenarios', brief descriptions of the future lessons or training programmes, as visualisations of alternative solutions. Secondly, with advanced modelling techniques it is possible to quickly prototype and try out parts of the future simulator, e.g., the interface. Note, however, that developing prototypes far enough to actually test their learning value with trainees will almost always be too expensive.

3.3.6 Acquisition procedures

Large organisations have established standard procedures that have to be followed during the acquisition of expensive systems. These procedures also apply to the acquisition of expensive instructional products, such as training simulators.

3.3.6.1 The RNLA acquisition procedure

As an example I briefly discuss the acquisition procedure of the Royal Netherlands Army (RNLA), as it was at the time when the research for this thesis was executed. This acquisition procedure consists of five phases (Commando Opleidingen KL, 1996; Directie Materieel, 1993; Ministry of Defence, 1996, 1997; Verstegen, Barnard, and van Rooij, 1999a;):

A) The concept phase
The development of a needs statement that describes which kind of systems are needed and why, and a rough estimate of the required budget.

B) The pre-study phase
The needs statement is elaborated into global functional specifications. After an inventory of what is available on the market and what is technically possible, a number of possibly suitable products is selected.

C) The study-phase
The functional specifications are elaborated further in detail; evaluation of whether the selected products fulfil the requirements. If necessary tests or experiments are executed.

D) The final report phase or acquisition preparation
The final functional specifications are formulated and a Request For Quotation (RFQ) is sent out. When the offers from the industry have come in, the final choice is made.

E) The realisation and evaluation phase
The functional specifications are elaborated into detailed technical specifications, usually by the manufacturer. The responsible RNLA personnel keep an eye on the production and on the delivery to the end user. An explicit evaluation is executed for expensive or special projects.

Every phase is concluded with a report that has to be approved by, depending on the amount of money involved, different levels of management (for very expensive acquisition projects including the Ministry of Defence and the Parliament).

Sometimes a separate project is started for the acquisition of a training simulator, but often the acquisition of instructional products is included in the project for the acquisition of the operational systems. In this case specifications for instructional products, such as simulators, are only a small part of the overall requirements (see, e.g., Paladeau, 1998; Schaafstal and Bots, 1998). Experience shows that instructional products are easily forgotten or considered less important. This leads to delays in the availability of instruction and training, and problems with financial resources for instructional products (van Rooij, 1997).

3.3.6.2 Discussion

Acquisition procedures structure the design process in time and define responsibilities. In each phase the specifications are further elaborated: from the global idea formulated in the needs statement to the detailed technical specifications at the end. Continuation of the process depends on permission by higher authorities that should be able to check whether the work has been done thoroughly. As such, they help in organising and monitoring the design process. Acquisition procedures are not specific for instructional products and they are oriented towards project management and budget control. They do not define how specifications for instructional products should be developed: what information should be collected, which analyses should be done, etc. For this purpose acquisition procedures should be combined with other models, e.g., instructional design and development models (see § 3.3.2 and § 3.3.3).

In practice, acquisition procedures are not always strictly followed. Sometimes phases are combined or only executed superficially. Design teams are not formally organised and are formed on an ad-hoc basis (van Rooij, 1997). Due to job rotation and other factors the composition of the design team may change over time and, depending on the situation, different parties may or may not be represented. RNLA instructional designers complained that the existing procedures only specify the official responsibilities of the team members, but not their roles in the project or the planning of the work to be done. Therefore, it often remains unclear who should deliver which input and when this should happen (Verstegen, Barnard et al., 2000). Based on the experiences during the STOW (Synthetic Theater of War) project, Dehncke and Brooks (1998) also recommend defining roles and responsibilities of team members more clearly.

Acquisition procedures build in safeguards, scrutiny and checks that should make sure that mistakes are corrected and that in the end the best solution is chosen. The disadvantage is that these checks make the acquisition process longer and more complicated: "It virtually guarantees that a training system will be both technically out of date and not as effective in meeting current needs by the time it is delivered" (McClure, 1998, p. 83). Kincaid (1997) claims that all the built in checks do not work because they are often done by people with little or no knowledge about the particular equipment. Moreover, in the maze of decision-making, nobody can be held personally responsible. Kincaid gives an example of how complicated the decision making can be in practice (Figure 3.9):

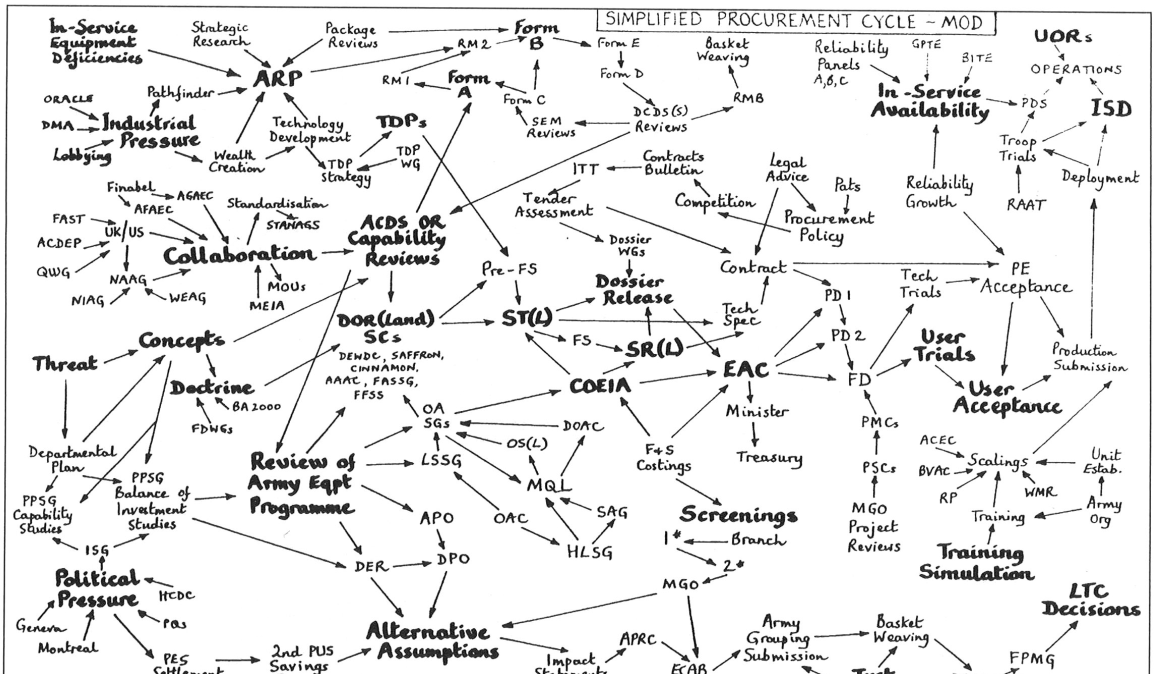


Figure 3.9: 'Simplified procurement cycle' (from Kincaid, 1997, p. 47, used with permission of the publisher and the author).

3.3.7 Organisational aspects

Most prescriptive models do not take the more practical aspects of the design process into account. Below I briefly discuss some prescriptive theories and models that aim at supporting these 'organisational' aspects of designing specifications for instructional products. I briefly discuss the communication and co-ordination within the design team and with other stakeholders in Section 3.3.7.1, decision making in Section 3.3.7.2 and the knowledge management aspect in Section 3.3.7.3.

3.3.7.1 Communication and co-ordination

The instructional design team
The design of specifications for instructional products requires different kinds of expertise. It is likely that a number of people will be involved, such as domain experts, teachers, instructional designers, software engineers, technical, legal and financial experts (see § 3.2.1). Some authors plead for the direct involvement of users, not just for the evaluation of prototypes, but as members of the instructional design team (see § 3.3.6). Working together requires communication and co-ordination, i.e. the exchange of information and the planning of tasks in time (Verstegen, 1997). Team members depend on each other: they need information from each other and some tasks need to be done before others. Therefore, clearly defining tasks and responsibilities is important (van Weert and Pilot, 2003; van Weert et al., 2002; see also § 3.3.6.2). The prescriptive models described above contribute to the communication process by providing a shared terminology, an understanding of the steps to be taken in the design process and a set of intermediate products (for example, a problem definition or learning goals). More concrete ideas to support the design team with a shared workspace will be described in Section 9.4.

Walker (1990) poses communication between instructional designers as the central factor in the design process. The design of instructional products is, he claims, a practical endeavour. Design problems are ill-structured and can therefore not be solved by the application of (prescriptive) theory alone. The best that designers can do is to choose a solution that is better than any other they know of at that moment. Continuous discussion between team members is the key factor in Walker's approach. It is the instrument to collect all available information about the situation at hand, to generate alternative solutions, to weigh the advantages and disadvantages of different solutions, to get a view of the ideas and values of all the interested parties and to reach a fair and balanced judgement. Theory about instructional design and development should not be the starting point, but it can support the design process by supplying and clarifying terminology, principles and lines of argument, and by drawing out hidden implications, pointing out needs for additional data and the like. Walker's approach is referred to as a communicative or relational approach (e.g., van den Berg, 1999; Visscher-Voerman, 1999, see also § 3.2.2.1).

Communication with other stakeholders
The need for communication goes further than the instructional design team. End-users, managers and other stakeholders are often not directly involved in the design process, but their approval is vital for the acquisition and successful application of new instructional products. The potential influence of stakeholders and the changes in constraints and the organisation itself during the project often create fuzzy situations and uncertainty (Moonen, 1999). Kincaid (1997) gives an example of how complicated decision making often is in practice (see Figure 3.9).

Kessels (1993, 1999) defines two kinds of coherence that are important in any instructional design project: internal and external coherence. Systematic approaches -such as the instructional design and development models described in Section 3.3.2 and Section 3.3.3- focus on internal coherence, i.e. coherence between the learning goals and the learning environments that are designed to reach these learning goals. The success of any educational effort also depends on external coherence: coherence between the perceptions of all stakeholders as to what the problem is and how it should be resolved. Kessels pleads for the combination of a systematic approach with a relational or communicative approach that focuses on actively organising frequent contacts with all stakeholders to give them the chance to give input, and to reach and maintain consensus.

Moonen (1999) formalises this idea into a multi-dimensional model with three activity spaces: a consensus space where global ideas are elaborated into structured design problems, a task space where specifications for (alternative) solutions are designed and half-products are implemented, and an implementation space where end-users are confronted with the half-products which are then further refined and improved. In an ideal situation, these spaces would be used one after the other, but in practice they will always exist in parallel at least part of the time. The three activity spaces are similar to the phases found during design processes in descriptive studies (see § 3.2.2.2). Note that Moonen's model is more in line with a pragmatic, prototyping approach than with an instrumental ISD approach.

Conclusion
Communicative approaches are promising, especially when they are used in combination with more systematic design methods. For the design of training simulator specifications several scenarios are possible. In some cases all experts work together in a team at one location. In other cases different (teams of) experts work on the specifications at different locations or at different moments in time, or the specifications are the responsibility of one person who calls in other experts when necessary. In all these scenarios the co-ordination of tasks and the communication between team members is important. Consensus with other stakeholders in the organisation is required before specifications will be accepted. Complications arise when the composition of the team changes frequently (e.g., due to job rotation), when team members have different backgrounds or interests (see § 3.2.1), and when their roles and the division of the work are not explicitly defined (see § 3.3.6.2). Furthermore, training simulators are often used in large (military and commercial) organisations, which are hierarchically organised. Final decisions are taken by higher management that the design team might never meet. A communicative approach may not suit this kind of organisational culture.

3.3.7.2 Decision making

Making decisions is one of the most difficult aspects of the design process for a number of reasons:
- Prescriptive instructional design models give contradicting advice regarding the kind of instructional products that are suitable for different types of learning goals, learners and settings (see § 3.3.2.2).
- In practice decisions are -and must be- heavily influenced by pragmatic factors, such as financial, technical, personnel, logistic and other constraints (see § 3.2.1). These constraints have to be taken into account from the beginning of the design process to avoid less optimal choices and unnecessary work. However, they may also distract the designers' attention from important instructional aspects.
- Decisions are influenced by different team members and stakeholders who may have different opinions and different interests; and teams may change over time due to personnel changes (see § 3.2.1).
- Systematic development models prescribe which decisions need to be taken, but not how this should happen (see § 3.3.3.4). Responsibilities and tasks of team members are not always clearly defined (see § 3.3.6.2).
- Frequently, many options are available with an abundance of data that may or may not be informative, credible and/or reliable. On the other hand, designers may lack information on some important aspects of the instructional problem situation (see § 3.2.1).

Support for decision making can take two forms. Expert systems give advice based on encaptured knowledge and heuristics regarding a specific kind of decision. Decision support systems, on the other hand, aim to support the process of decision making. Seel, Eichenwald, and Penterman (1995) describe an example of the latter, based on the idea that decision makers can optimise decisions by explicitly defining the alternatives, the factors that should influence their decision and their personal preferences. Verwijs (1998) chose a similar approach. Based on the idea that, in practice, instructors will want to make decisions about instructional products themselves and that their decisions will be based on a mixture of rational and social influence models (cf. § 3.3.7.1), she has developed a system to support the decision making process.

Van der Hulst, de Hoog, and Wielemaker (1999) developed a tool that provides both forms of support for decision making: BOOT (Dutch abbreviation for decision support for the selection of instructional products). BOOT focuses on those situations where the fidelity of instructional products plays a large role, and is therefore one of the few tools specifically geared towards the design of specifications for training simulators. Assuming that a thorough task analysis and an inventory of alternative solutions have already been done, BOOT supports the selection of types of instructional products. Users are asked to answer a number of questions about the situation at hand concerning the kind of tasks to be learned and the target group of trainees amongst others. An expert system determines the consequences and shows the advantages and disadvantages of different kinds of instructional products in this concrete situation. The system allows the users to assign weights to the different factors, such as cost, learning value, and the required amount of instructors, and then provides an overview of the alternatives. The tool visualises the results of this multi-criteria analysis and allows users to easily explore the effects of adjusting criteria and the weights assigned to them. It also supports the process of decision making by providing elaborate commenting facilities that encourage a structured discussion between team members and/or stakeholders. Thus, it supports communication within the design team and with stakeholders.

3.3.7.3 Knowledge management

Standards are being developed to enable the exchange of different kinds of instructional products, ranging from entire courses or modules to single pictures or other source material (see 3.3.1.1). Training simulators, however, are usually developed for very specific systems and situations and they contain many hardware components. Reusing entire simulators is usually not an option (see § 1.3.2). Reusing the information that is collected and the concepts that are developed can, however, save time and energy. Van der Hulst and Verstegen (2000) identified four possible ways to reuse data that are collected during the design of specifications of training simulators:

- Justification: the documentation can be used to justify decisions and to argue that the resulting simulator specifications will lead to effective and efficient training.
- Iteration during the design process: when designers want to review previous decisions, they can review the reasons and the alternatives that have already been taken into account. This is even more important when the decision was originally taken by somebody else.
- Development of technical specifications and training programmes: information can be reused when the functional specifications have to be elaborated into detailed technical specifications and, later on, when the training programme has to be further refined and implemented (see also Figure 9.2 in § 9.3.1).
- Other design projects: information can be reused when simulator specifications have to be designed for similar domains or when the simulator has to be updated or replaced in the future. The more general knowledge and experiences of the design team can also be documented and used to further optimise the design process.

For traceability purposes information about the design process is required, for instance, why decisions were taken and by whom, and which alternatives were considered. However, the field study showed that documentation regarding the original simulator specifications is almost never available (see § 2.3.1.3). This means that it is impossible to judge whether the current simulators fulfil the original requirements or not. It also means that for a new project the designers always have to start from scratch, even when it concerns the update or replacement of an existing training simulator. Good documentation ensures that all information is traceable and can be reused at a later stage. Indexing is necessary in order to make documented information easy to find, also for people that were not involved in the project. Boy (1999) suggests the use of Active Design Documents (ADD) for design projects: instead of designing an artefact and documenting it later, specifications and/or prototypes are developed and evaluated simultaneously and incrementally. Comments from members of the design team, users or other stakeholders are stored to document the design rationale and/or to further optimise the design. The commenting facilities of BOOT (van der Hulst et al., 1999), described in Section 3.3.7.2, serve a similar purpose. Another example is PROSUS developed by Blessing (1994) for the field of mechanical engineering. In a case study (see also § 3.2.2.2) Blessing observed that the design history plays an important role: new concepts were often based on other (previously refused) concepts or on existing products. Furthermore, designers also seemed to assess the quality and relevance of information. Blessing observed that decisions were often taken implicitly focusing on only a few, sometimes not even documented, requirements and that the design was continuously adapted without assessing costs and benefits. PROSUS represents the design process as a structured set of issues and activities. The system uses design matrices to provide facilities to document and structure the design process, and to provide context-sensitive advice.

In many organisations standard acquisition procedures are installed to make sure that the results of the development process are regularly checked by peers and/or superiors (see for example § 3.3.6.1). The reports that are produced within those procedures can be a form of knowledge management, although they tend to only describe the results of the design process, i.e. the functional specifications for the simulators. Good documentation should also describe the underlying reasoning, document the design process itself, and make it easier to identify the people that have been involved in a specific project, so that they can be approached for comments or advice.

3.3.8 Supporting software tools

Tools to support the development process have been built to try to increase the productivity of designers, to improve the quality of the resulting products and to make up for lack of expertise. In large organisations, such as military organisations, standardisation is often an important goal as well (Verstegen, Barnard et al., 2000; Wulfeck, Dickieson, Apple, and Vogt, 1993). Goodyear (1994) distinguishes a strong and a weak approach. Tools that follow a strong approach aim at automation of the design or even the entire development process, i.e. they replace human activity. Tools that follow the weak approach, on the other hand, provide support for the human designer, who remains in control. A large number of tools are described in the literature[5]. Below, I will only give some examples of different categories of tools.

3.3.8.1 Support for the human designer

There are different kinds of tools that can support human designers of specifications for instructional products. One kind of tools supports the step-by-step execution of a development model or a part of a development model (see § 3.3.3), sometimes offering database facilities to store the collected information and limited advice facilities as well. The commercial tool

[5] In the AIDA project, a project funded by the U.S. Air Force, different possibilities for tool development have been explored (Spector, Polson, and Muraida, 1993; Tennyson, 1994; Tennyson and Barron, 1995). More recent overviews of tools are given in, for example, Murray (1999), Nieveen (1997) and van den Akker, Branch, Gustafson, Nieveen, and Plomp (1999); in the latter the chapter by Nieveen and Gustafson provides a good overview on the tools that are discussed.

Designer's Edge (DE 2.0, 1997) is a well-known example in this category. Designer's Edge guides users through a 12-step ISD model: from 'Analyse needs' to 'Evaluate course' (see also Figure 5.1 in § 5.2.1). It stores the results of each step, and -where applicable- presents them as input for a later step. Explanations, guidelines and examples are available on request in a standard Windows help system. Designer's Edge focuses on the design of CBT and in later versions web-based applications. Some large organisations have developed their own in-house package to support their own ISD model, e.g., the GOOS (Dutch abbreviation of integrated instructional design system) system used by the RNLA. The ADAPT^{IT} (Advanced Design Approach for Personalised Training-Interactive Tools) tools are developed to support users of the 4C/ID model described in § 3.3.4.1 (de Croock, Paas, Schlanbusch, and van Merriënboer, 2002). Tennyson and his colleagues wrote specifications for an intelligent tool that should be able to adapt the development process to the user's situation and level of expertise and give appropriate advice (Tennyson and Breuer, 1994; Tennyson and Elmore, 1995; see also § 3.3.3.3).

Another kind of tools supports one specific task or aspect during the design process. These tools function as 'job-aids'. CASCADE (Computer Assisted Curriculum Analysis, Design and Evaluation), for example, provides computer-based support to assist curriculum designers, focusing on formative curriculum evaluation activities (Nieveen, 1997). CASCADE has been extended for other settings and goals (McKenney, Nieveen and van den Akker, 2002). The GAIDA (Guided Approach to Instructional Design Advising) tool guides the design of courseware lessons based on Gagné's nine events of instruction (Gettman, McNelly, and Muraida, 1999; Spector and Song, 1995). GAIDA provides step-by-step advice along with examples of lessons in four different domains. It is oriented towards novice designers. IDE-Interpreter (Instructional Design Environment Interpreter), on the other hand, is meant for experienced designers (Russell, 1988; Russell, Moran, and Jordan, 1988). IDE-Interpreter is a hypertext system that enables users to create a linked network of 'notecards' that represents the organisation of the course or curriculum and the design decisions and underlying reasons. This network can be used to get an overview in a browser or to trace back the reasons for particular decisions. Thus, IDE-Interpreter provides facilities to support structured decision making and knowledge management (see also § 3.3.7.2 and § 3.3.7.2).

Finally, there are general purpose commercial tools that can be used during the design process, e.g., project planning tools, e-mail, video-conferencing, web-based discussion groups and chat rooms. Still under development are several standards for packaging and indexing (parts of) instructional products, see § 3.3.1.1 .

3.3.8.2 Automating (a part of) the development process

Other tools follow a strong approach and aim at automation of (a part of) the development process. The SCO-generator (Boot and Bots, 2002; Boot and van Rooij, 2000), for example, helps instructional designers to produce Sharable Content Objects (SCOs) that meet the SCORM standard (see § 3.2.1.1). The SCO-generator incorporates instructional design knowledge and provides partly elaborated templates that designers can fill with domain-specific information (e.g., text, pictures or videos). The REDEEM tool, for example, assumes that electronic lesson material is already available (from previous CBT, text files or other sources) and provides tools to automatically organise the material and create exercises (Ainsworth, Underwood, and Grimshaw, 1999; Major, Ainsworth, and Wood, 1997). Users can define categories of learners and compose learning strategies by choosing settings on several menus. During delivery, the system will adapt to individual learners based on these settings. Another example is GTE (Generic Tutoring Environment), which provides an authoring language and facilities to construct instructional strategies separately from lesson content and instructional product, thus making it possible to automatically generate different kinds of lessons from the same source material (van Marcke, 1998).

Other tools are oriented towards specific settings, using for instance one of the hybrid models (see § 3.3.4). CBESS (Computer Based Educational Software System), for example, provides development tools for creating CBT based on templates for common types of learning goals such as repetitive fact learning, locating and replacing faulty parts in a simulation of a machine, and technical vocabulary learning (Wetzel, 1993). The CBESS tool automatically generates relevant learner activities, e.g., multiple choice, true/false, matching and definition building exercises, and provides facilities to create interfaces and lesson sequences. The LEAP (Learn, Explore and Practice) tool is another example: it provides facilities to create CBT to teach employees the knowledge and skills they need to respond effectively to customer requests and problems, based on predefined communication patterns (Sparks, Dooley, Meiskey, and Blumenthal, 1999). Based on the ID2 transaction shells discussed in Section 3.3.4.2 are IDXelerator (™ Mindware Creative, 2003; Merrill and Thompson, 1999) and XAIDA (Experimental Advanced Instructional Design Advisor, see, e.g., Hsieh, Halff, and Redfield, 1999; Spector and Muraida, 1997; Spector and Song, 1995). Tools such as SimQuest (de Jong et al., 1999; Kuyper, 1998; Kuyper, de Hoog, en de Jong, 2001) focus on the construction of CBT including simulations of technical systems. These tools allow users to easily construct simulations and to generate associated exercises or assignments using available templates.

3.3.8.3 Discussion

Tools from the 'support category' may be useful to support managing the design process or executing specific tasks during the design of training simulator specifications. They take over bookkeeping tasks, enforce a systematic design process and/or offer advice. There are two potential problems with these tools. Firstly, most of the tools assume that the users are instructional designers, either experienced (e.g., IDE-Interpreter) or novices (e.g., GAIDA). The designers of specifications for training simulators, however, do often not have a background in instructional design theory. Secondly, most tools are not specifically geared to specifying training simulators. Those that are, are developed for a specific organisation (e.g., ADOPT and GOOS) or support only one specific aspect of the development process (e.g., BOOT). None of the tools that follow the automation approach address simulator training. Moreover, most of them focus on the design and implementation of a specific kind of instructional product, usually CBT, not on the design of specifications.

3.3.9 Conclusions

In Section 3.2.3, I concluded that the design task has four main aspects:
1. Gathering information.
2. Solving the design problem.
3. Managing the design process.
4. Communication.

The prescriptive models and theories described in Section 3.3 support these aspects in different ways. Prescriptive models can provide a framework for gathering, organising and storing information, especially in combination with software tools (see § 3.3.8). Solving the design problem is supported by the System Theory that gives indications about decomposing the design problem into sub-problems (see § 3.3.1), and ID models that prescribe what instruction and instructional products should be like under certain conditions (see § 3.3.2). ID models can give a theoretical foundation for important decisions, and further support can be provided by different kinds of decision-support systems (see § 3.3.7.2). Hybrid design and development models (see § 3.3.4), if applicable, offer further elaborated models for specific kinds of design problems, a kind of half-products or templates comparable to the schemata that instructional designers develop from experience (see § 3.2.2.4). Prescriptive models provide a shared framework and terminology, and the communicative approaches specifically focus on the communication aspect of the design task (see § 3.3.7.1).

Instructional development models support the management of the design process. Ideally, specifications for instructional products should be based on what learners need to learn, and how they can learn it most effectively and efficiently (cf. Figure 1.1). Then the instructional designer can make a well-considered choice of what to cover with simulator training, and specify the appropriate level of fidelity (see § 2.1 and § 2.4.1). In order to specify a good instructional overlay, i.e. facilities to support training and instruction, insight into the future training method and programme is required. ISD models (see § 3.3.3) prescribe a sequence of steps or activities to reach this purpose. However, the design of functional specifications for training simulators is an inherently iterative process. Instructional designers have to deal with all kinds of events that 'disturb' the ideal design process, and with the uncertainty that is caused by the incompleteness and unreliability of information (see § 3.2.3). Experienced instructional designers have the problem solving skills necessary to organise, monitor and adapt the design process (see § 3.2.2.1 and § 3.2.2.3). However, the designers of functional specifications for simulators are usually not experts (see § 3.2.1), and descriptive research shows that novices are not very good at managing their own design process (see § 3.2.2.3). Thus, current ISD models permit iteration, but do not sufficiently support this aspect of the design task for novice designers: it is not clear how and when iterations need to occur and it is up to the designers to manage this aspect of the design process. The rapid prototyping models provide ideas to deal with iteration in a different way (see § 3.3.5). In practice, iteration will also to some extent be enforced by acquisition procedures, which structure the design process in time (see § 3.3.6).

3.4 Forms of support for the design process

In this section, I look at the results of the literature research from a different perspective, namely: instructional designers need facilities to support all four aspects of the design task identified in Section 3.2.3: First of all, they need support for information gathering:

A) Facilities for handling input information
Facilities to, for example, organise and store information, to provide an overview, and to retrieve information when necessary, in an appropriate format.

B) Facilities for modelling the problem and the solution(s)
Facilities to, for example, represent the results of different kinds of analyses, the decomposition into sub-problems and solutions to (sub-)problems; and facilities to store and compare alternative solutions to design (sub-)problems.

C) Facilities for documentation
Longer term storage of the design results and the design process (see § 3.3.7.3) concerns questions like: where did information come from? who took certain decisions? what were their reasons? which alternatives were considered?

Secondly, instructional designers need support for solving the design problem:

D) Strategic support
Advice on how to execute the different activities in the design process, for example advice about where information can be found and how it can be easily collected and organised, and who could give advice on specific issues.

E) Heuristics and guidelines regarding instructional design
In this context, knowledge about using simulators for training purposes and about simulator training programmes. This knowledge is currently dispersed over peoples' personal experience and research literature in different areas, e.g., hardware and software development,

instructional design theory and learning psychology. Guidelines should be linked to specific steps or activities in the design process.

F) Decision making support
At important decision points specific support can help designers to deal with multiple and contradicting requirements and constraints. Decision support can take the form of automated advice or facilities to organise information and arguments and compare alternatives (see § 3.3.7.2).

Thirdly, instructional designers need support for managing the design process:

G) Procedural support
A prescriptive model can help to identify important activities and decisions in the design process and make sure that no important issues are forgotten. The model will probably have to be adapted to fit the organisational context and the problem at hand.

H) Support to deal with iteration
Training simulator specification is clearly not a linear process (see § 3.2.3). For different reasons it will be necessary to go back and execute (a part of) some design activities again, for example to review a decision or to elaborate an analysis in more detail. This is called iteration (see also § 5.4.1). Designers iterate to design solutions in more and more detail and to make revisions. Related to this issue is dealing with uncertainty: designers will have to make assumptions and take preliminary decisions based on incomplete information, and they will have to keep on checking whether these decisions are still valid given the most recent information and insights. For possible ways to support dealing with iteration see Section 5.4.2 and the discussion in Section 10.3.

I) Support for quality control
Quality control is important for the organisations' managers, not only to ascertain that the designers have done their work well, but also to be able to claim resources and to prove that the acquisition of a new simulator is necessary to solve existing training needs.

And finally, instructional designers need support for communication:

J) Support communication within the team
Facilities to, for example, exchange information, and to divide and plan the work to be done.

K) Support communication with stakeholders
Facilities to, for example, retrieve important input information, to involve stakeholders in the design process, and to acquire approval necessary for the acquisition and successful application of new training simulators.

In the literature different kinds of prescriptive theories and models are described that all support one or more of aspects of the design task (see § 3.3.9). None of them, however, provides coherent support for all four aspects. In practice, designers are tempted to follow a strictly linear model, or to react intuitively and not use a systematic approach at all. The challenge is to offer a clear structure and a systematic model, while at the same time allowing flexibility and supporting iteration during the design process. Advice and guidance should be available when needed: just in time and just enough.

In Chapter 4, I will present an ISD based method tailored to the specification of training simulators (MASTER), and I will discuss to what extent this method fulfils the requirements for design process support defined above, and the requirements for the design product defined in Chapter 2 (see § 2.4.3). I have used the MASTER method for research regarding support for managing iteration (Aspect H above) in the second half of this thesis.

Chapter 4
MASTER method for training simulator specification

In Chapter 4, I describe a method based on the ISD approach and specifically tailored to the specification of training simulators: the MASTER method. The research question addressed in this chapter is:

6. To what extent does the MASTER method fulfil the product requirements defined in Question 3 and the process requirements defined in Question 5?

4.1 Introduction

Ideally, specifications for instructional products should be based on what trainees need to learn, and how they can learn it most effectively and efficiently (cf. Figure 1.1). Then the instructional designer can make a well-considered choice of what to cover with simulator training, and specify the appropriate level of fidelity. In order to specify a good instructional overlay, i.e. facilities to support training and instruction, insight into the future training method and programme is required (see § 2.1 and § 2.4). Instructional System Development (ISD) models prescribe a sequence of steps or activities to reach this purpose and help to organise and manage the instructional design process (see § 3.3.3). An ISD-based method for the design of functional specifications for training simulators has been developed in the context of the European defence research project MASTER (Military Applications of Simulator and Training concepts based on Empirical Research, EUCLID 11.1). In Section 4.2 the MASTER method is described, including a brief description of the envisioned software tools and the first reactions of users during a rather informal evaluation executed by the MASTER consortium. Subsequently, in Section 4.3, I discuss to what extent the MASTER method fulfils the product and process requirements defined in Chapter 2 and Chapter 3 respectively. Finally, in Section 4.4, three main areas that need further elaboration and research are identified.

As explained in Section 1.2.1 the terms trainee and instructor are commonly used in the context of simulator training (rather than learner and teacher). In line with this, the MASTER method uses the terms training goal (instead of learning goal) and training activity (instead of learning activity). To avoid confusion I use the same terms in this thesis when I refer to settings where simulators are used (even though the terms learning goal and learning activity might be more correct from a theoretical point of view, see also § 1.5). Note that training activities can be traditional instructor-led exercises in a training simulator, but also more active and trainee-led activities in the simulator or with other instructional products before or after simulator sessions.

4.2 The MASTER method

4.2.1 Background and scope

The reason to start the MASTER project was that European military services became aware that simulators often do not meet the expectations and/or are very expensive. There was a general feeling of dissatisfaction about the time lag between procurement of systems and delivery of (adapted) training packages or devices, the failure to consider implications for the role of training personnel and failure to reach the required level of performance after training. In other words, there were more and more doubts whether the simulators in use brought enough value for money in training (Farmer, Jorna, Riemersma, van Rooij and Moraal, 1999). The goal of the MATER project was investigate the areas for improvement and to work on a new method for design and development of training simulators (see for the partners in the project the footnote with § 2.2.1). The main idea underlying the MASTER method is that the specifications for training simulators should be derived from training needs instead of operational or technical considerations (Riemersma et al., 1996). For this purpose, a detailed description of the simulated system does not suffice: an analysis of what is going to be trained and how it is going to be trained is required in order to formulate valid training simulator specifications (cf. the ideal of Analysis-Design-Production, depicted in Figure 1.1). This systematic process should replace the frequently adopted strategy of striving for realism.

The MASTER method consists of three main phases (Farmer et al., 1999):
1. Training Needs Analysis (TNA),
2. Training Programme Design (TPD), and
3. Training Media Specification (TMS).
Evaluation is supposed to play a part in each of these phases as depicted in Figure 4.1.

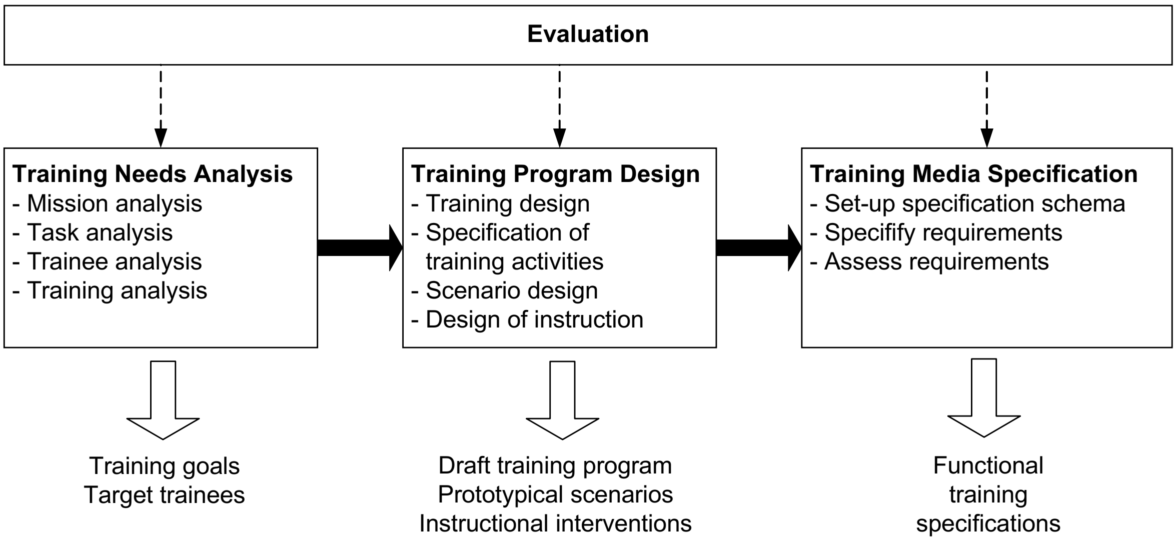


Figure 4.1: The main phases of the MASTER method, with their steps and outcomes.

The MASTER method structures the design of training simulator specifications, specifying a sequence of steps to be taken and the required input and output for each step. This also enables some form of quality control: decisions are made explicit; and, because the input for one step consists of the output of previous steps, it is not possible to skip steps or forget issues that should be addressed. The MASTER method starts with a general needs analysis, regarding the whole domain and the whole range of tasks and training goals. Later on, halfway the design phase, the focus shifts to the part that will be covered with simulator training, see § 4.2.2). From then on, the specification of other instructional products is outside the scope of the method. Note that the MASTER method is tailored to the design of specifications, and not towards the design of detailed and directly applicable training programmes. During the design phase the method works towards a draft of the training programme that is then used to derive simulator specifications. The final training programme can only be developed and implemented when the simulator has been built or bought. The steps that need to be taken then to further elaborate the training programme and training scenarios are very similar to those in the TPD phase of the MASTER method, but the method and the available tools are not equipped to support use for this purpose.

In terms of the classical five ISD phases (see § 3.3.3) the MASTER method covers the analysis phase with the TNA. The design phase is split up in the TPD, focusing on designing a plan for the learning process from an educational viewpoint, and the TMS, focusing on specifying the required training simulator from an instructional product viewpoint. The production and the subsequent evaluation and possible improvement of the training simulator are outside the scope of the MASTER method. This is not strange given that the production of training simulators requires specialised personnel and is often contracted out to industry (see § 1.3.3).

4.2.2 The steps of the MASTER method

In this section I will describe the main steps in the three phases of the MASTER method. For a more detailed description of all the steps and sub-steps, see Farmer et al. (1999).

The goal of the Training Needs Analysis (TNA) phase is to analyse what is going to be taught with the training simulator. It consists of four main steps: mission analysis, task analysis, trainee analysis and training analysis. The mission analysis is a general analysis of the domain, oriented towards a global description of the system (in the System Theory meaning: a unit composed of both humans and technical systems), the different operational objectives that the system should be able to accomplish and the relations with the physical and tactical environment. The goal of the task analysis is to identify and describe the behaviour expected from individual operators (if working in teams including the parallel tasks of co-ordination and communication) and the skills that operators need to acquire to perform their tasks adequately. Then the entrance level and other characteristics of trainees are analysed (Trainee analysis) and, finally, training goals are formulated (Training analysis). These training goals describe what the trainee is expected to be able to do after training, in terms of behaviour, level of competence, environmental conditions and systems to be used.

The goal of the Training Programme Design (TPD) phase is to specify how training is going to be provided. The input for this phase is the output of the TNA: a set of training goals and a description of the target group(s) of trainees. The TPD consists of four main steps: Training design, Specification of training activities, Scenario design and Design of instruction. Before the actual content of training is further elaborated, some general decisions are taken in the first step (Training design): a selection of general training strategies and/or learning principles that will be applied during the design (and later the delivery) of training, the sequence in which training goals will be addressed, an inventory of suitable instructional products and the construction of an assessment plan. The goal of the second step (Specification of training activities) is twofold. With the definition of training activities the designers specify how the training goals are going to be achieved. Then they decide which of these training activities are suitable for simulator training and which are not. In the last two steps only those that will be covered with the future training simulator are further elaborated. Scenario design is meant to get insight into the simulator facilities required for the execution of specific training activities. For this purpose exercises, usually called training scenarios in this context, do not have to be elaborated in full detail. A rather abstract description of prototypical scenarios is sufficient, provided that it contains enough information about the required behaviour of the future simulator, i.e. the kind of events that should be possible, the kind of environments that should be simulated, and the features and functions of the real system that should be available for training purposes. Finally, in the step Design of instruction, instructional interventions are planned explicitly to ensure that the facilities required for instruction will be included in the simulator specifications. For this purpose, the focus is not on the content of instruction, but on the way it is going to be provided: which kinds of interventions are foreseen and when, how and by whom will the required information be given to the trainees (see § 2.1 for the definition of the terms training and instruction in the context of simulator training). This TPD phase has been used for empirical studies. It is described in more detail in Chapter 5.

The draft training programme, the prototypical training scenarios and the instructional interventions, are the input for the third phase. In the Training Media Specification (TMS) phase the designers set up a specification scheme by selecting norm scenarios, i.e. the most relevant or critical ones, and a simulator configuration to use as starting point (for the standard configuration proposed by Farmer et al., 1999, see Figure 3.3 in § 3.3.1.1). The designers start with one training scenario and specify a simulator capable of executing this scenario. Then they proceed with the next scenario, etc. After a set of training scenarios has been processed, it may be necessary to pause and compare specifications across scenarios in terms of factors such as comprehensiveness and functional fidelity. This way of relating specifications directly to training scenarios makes it easy to review the consequences for training if certain facilities cannot be realised. The MASTER method stops at the level of functional specifications that describe the kind of events and environments that should be simulated, the system features and facilities that should be available to trainees and the facilities that should be available to the instructor.

Although the MASTER method prescribes a sequence of steps to be taken, it does not envision a linear design process. The assumption is that it is not possible to determine a priori in how much detail the steps need to be executed since this depends on the requirements of subsequent steps, which are not all known in advance. The purpose is to make these dependencies explicit and transparent. Iterations are possible, but, to keep information consistent, users are obliged to go back to the step where the information was entered to make changes. The method can be used at different levels of detail, so that users can execute steps on a more global level taking preliminary decisions and can easily go back to previous steps for further elaboration, review and/or the collection of extra information when necessary. In this way the iterative design process also helps to avoid unnecessary work, e.g., because users can focus on those parts or aspects that will probably be covered with the future simulator. See Section 5.4.1 for a more detailed discussion about the MASTER method and iteration.

With different kinds of demands and interests playing a role some compromises will have to be made during the design process. Farmer et al. (1999) distinguish two extreme strategies:

- Need-driven, i.e. first specify what is considered optimal, and only afterwards look at what is feasible; the obvious risk is that much time may be spent on a design that is simply not feasible.
- Constraint-driven: first determine constraints and then try to fit the design to these constraints; the risk of this strategy is that much time may be spent on designing a training programme that is feasible but not effective.

The ideal route through the method varies, depending on, amongst others, the size of the domain, the experience and preferences of the designers, and limitations on the available resources. For instance: when financial resources are limited, a more constraint-driven approach seems logical. Focusing on training goals that cannot be trained effectively with other instructional products or during field training, a set of minimal specifications is designed. At a later stage, designers can investigate whether the financial resources permit extra facilities needed to train a wider range of training goals. When, on the other hand, the availability of operational systems and experienced coaches/instructors is the main problem, a need-driven approach might be more appropriate.

4.2.3 Supporting software tools

Within the MASTER project a set of prototype software tools to support users of the MASTER method has been developed. The tools guide users through the steps of the method and take care of data storage and consistency. For each step, the necessary information is presented in input windows (see Figure 4.2, left side). Input can be information collected by the user in a previous step or information available in the system, e.g., guidelines, libraries, etc. Users can enter new information in the output window (see Figure 4.2, right side). Information is stored in an object-oriented database and the relations between objects are preserved in order to make their origin traceable. In this way, the user can easily review, for example, from which task and from which mission certain training goals were derived.

Since the results of each step are stored, it is easy to jump back to previous steps to review and adapt information collected earlier, if necessary. Navigation can be done either with buttons in the tool bar, or using an overview of all the steps of the method. The overview can also be used to get a quick view of the results obtained so far for each step. A special resource window is available on request to save information regarding available resources.

Apart from general help facilities, the tools reserve space for more specific help for each (sub)step. Guidelines can be given in a separate window that pops up on the user's request. They can provide advice concerning the current step, e.g., which information should be collected, how decisions should be taken, the advantages and disadvantages of choosing certain options, when to proceed to the next step, etc. Where possible, libraries can provide the user with a list of options to choose from, e.g., a library of training strategies or a library of possibly relevant environment characteristics. Additional information can be attached to the items in the library, e.g., definitions or information regarding the implementation costs. Guidelines and library items can also be added or elaborated by users. The actual content of guidelines and libraries, however, has not been specified within the MASTER project, though some examples are provided. The MASTER handbook (Farmer et al., 1999) contains background theory and knowledge from experience that could be used as a starting point for the elaboration of guidelines and libraries.

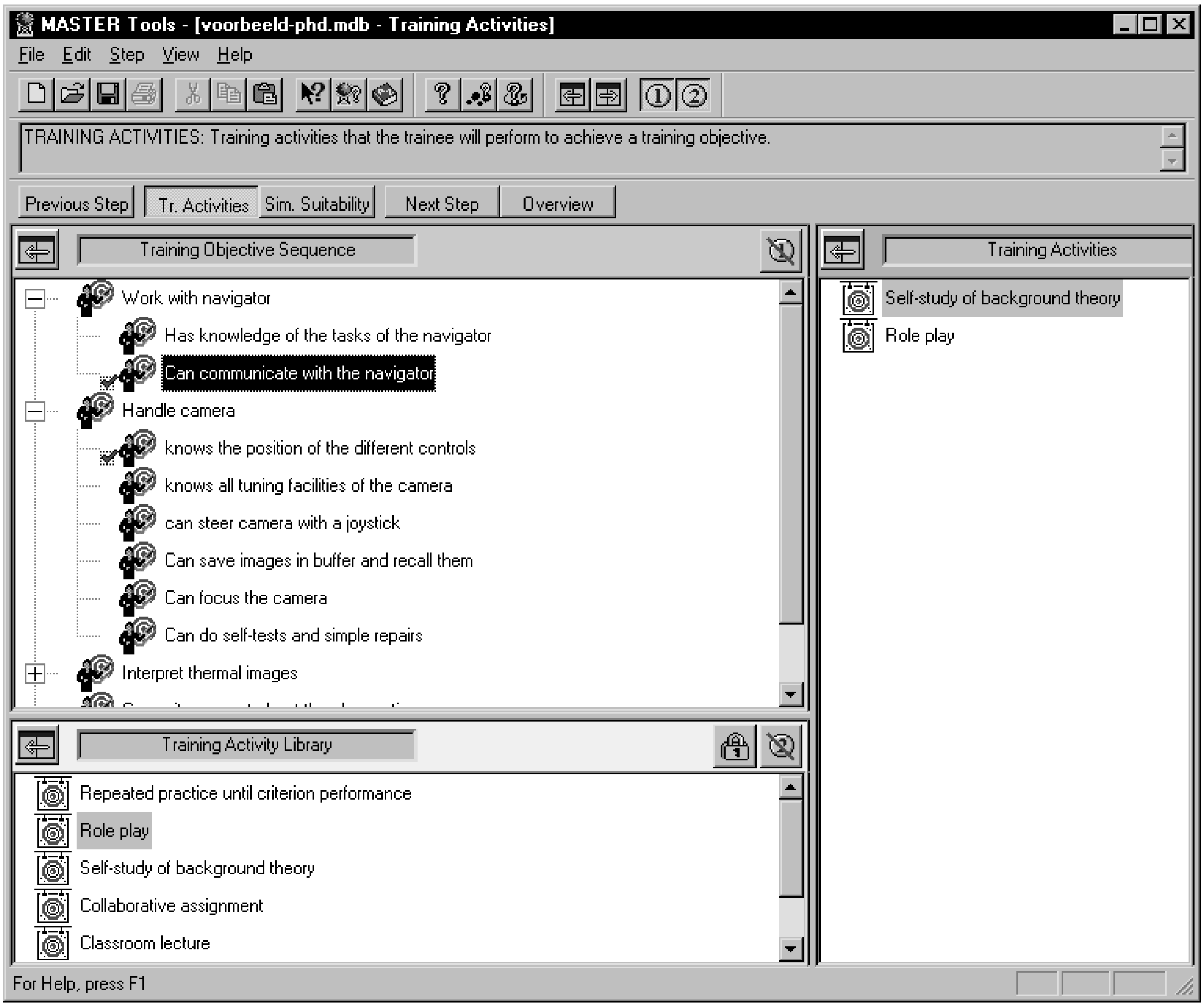


Figure 4.2: Screen dump of the prototype MASTER tools, in the TPD step 'Specification of training activities'. On the left side the input windows: training goals, and training activity library; and on the right side the output window for this step (N.B. in these tools the term training objective is used for training goal, see Appendix A).

4.2.4 First reactions of prospective users

Within the MASTER project a first, rather informal evaluation was conducted to guide the further development of the MASTER method and the implementation of prototype tools. The key questions were:
- Is the MASTER method valid, acceptable to users and easy to use?
- Will the tools, when fully implemented in future, offer all the necessary and desirable support?

I will focus on the first question since the prototype tools developed within the MASTER project have not been used for my research (see for reasons § 5.2.1).

Since there was no functional prototype at the time of this evaluation, it took the form of a guided walk-through. The subjects could not really work with the method and tool themselves and, therefore, the results cannot be used to draw firm conclusions about their actual validity or usability. The results reported in this section are a summary of the subjects' first comments on the MASTER method in general and the Training Programme Design (TPD) phase in particular, since I have used that phase for my own research (see Chapter 5 and further). See for a more elaborate description van Rooij et al. (1998b).

4.2.4.1 Method

Fifteen evaluation sessions were executed by different consortium members with one to three subjects each. The sessions lasted two to three hours and consisted of:
- an introduction covering the aim of the MASTER method and the goal of the evaluation,
- a short walk-through to illustrate the steps of the TPD phase,
- a detailed walk-through to collect comments on all steps,
- a demonstration of other facilities envisioned to be provided by the prototype tools in future,
- a short interview, and
- a questionnaire to elicit opinions and suggestions regarding the TPD phase and tool.

During the walk-through one evaluator went through all the TPD steps using a small example that was implemented in the tool. To keep evaluation sessions consistent and to avoid possible software problems all actions were specified in an elaborate evaluation protocol (see van Rooij et al., 1998b, Appendix B) and executed by an evaluator. A second evaluator took notes of the subjects' comments.

A total of 24 subjects from four countries (the Netherlands, the United Kingdom, France and Spain) participated:
- eight researchers involved in research regarding the specification, design and use of training simulators (with a background in education, human factors and/or experimental psychology),
- four subjects from industry involved in the production of training simulators, and
- twelve representatives of the target military users from the Army, Navy and Air Force.

4.2.4.2 Main results concerning the TPD phase

Clarity and rationale
With the information provided in the introduction and, where necessary, extra explanations from the evaluators, most subjects understood what would be expected of users of the MASTER method and felt confident that they would be able to execute the different steps of the TPD phase. They thought, however, that future users would need some background in instructional design and development theory. Two subjects did not see the necessity for such an elaborate method; in their view training simulator specifications can be designed on paper just as well.

Not all steps are equally clear to the subjects, and it is not always easy to understand how the different steps will influence the final results. Some subjects remarked that it was hard to get an overview of the method and/or the results. Another concern was that users might lose their overview of the whole training programme when they focus only on the part that will be covered with simulator training, or when they are working on individual training scenarios. They might be encouraged to specify too many (unlinked) requirements instead of focusing on a minimum configuration for the whole course. These problems may have been partly caused by problems with the interface of the prototype tools (as they were at the moment of this evaluation).

Structured approach
The MASTER method proposes a structured approach: each step builds upon the results of the previous one. The subjects appreciated this structured step-by-step approach. However,

some of them were not sure whether such a strict and compelling method is always desirable and cost-effective. It may be only necessary for novices, for instance. Furthermore, there was discussion about the required level of detail. Some subjects were afraid that the method aims at a level of detail that is not cost-effective; others, on the other hand, thought that a more detailed elaboration would improve the design process.

Usability
There seem to be several potential target groups of users for the MASTER method. However, these target groups have not been clearly specified and this caused confusion. Furthermore, it was hard for subjects to judge whether the MASTER method would be useful for the design of training simulator specifications, since they could not work with it themselves.

Several subjects remarked that they hoped it would support more objective decisions regarding the acquisition of instructional products. Subjects appreciated the comprehensiveness of the MASTER method, i.e. all the way from mission analysis to simulator specifications, and the facilities to capture and archive information. It was remarked, however, that the design phase should also address the integration of simulator training with other forms of training, e.g., with other simulators or with the operational system, and the specification of other instructional products.

Extra steps
Some subjects asked for extra steps or explicit points for: (self-)evaluation, the specification of the duration of training and the frequency of (refreshment) training, the collection or updating of information about available resources, and inspection of relevant data from completed projects or previous stages in order to avoid making the same mistakes. One subject stated that the method should be adaptable, so that organisations can fit in their own instructional development approach.

Support
In general, subjects appreciated the support that was provided. Guidelines and libraries should be expanded, however, and refer explicitly to the instructional design theory that they are based on. Letting users adapt guidelines and libraries can be positive, but there are also risks with uncontrolled expansion.

Subjects asked for 'more active' support and guidance, such as selective advice, warnings when a decision is inconsistent with available information or earlier decisions, or even a complete knowledge-based system. For example: supporting the consistent use of training strategies by applying filters later on, so that only sequencing principles or assessment methods will be presented that are consistent with the chosen strategy; or the development of specialised agents or small expert systems to support decision-making at specific points (cf. § 3.3.7.2).

Relations between data
A number of more detailed comments concern the restricted relations between data that are allowed by the MASTER method, as implemented in this prototype tool, e.g., being forced to select training strategies or learning principles for the whole course (and not at individual training goal level), not being able to indicate that sometimes the sequence of training goals is irrelevant, or having to select suitable instructional products for each training goal (and not being able to assign a whole cluster of training goals to one instructional product). One subject would like to link activities of the trainee to activities of the instructor.

4.3 The MASTER method reviewed

In Section 2.4.3, I defined issues that should be addressed during the specification of training simulators from an educational point of view. These can be seen as requirements for the product of the instructional design process, i.e. the training simulator specifications. In

Section 3.4, I defined which kinds of facilities are necessary to support the instructional design task. These can be seen as requirements based on characteristics of the instructional design process. In this section I discuss to what extent the MASTER method fulfils those product and process requirements. I have based this review on the description of the MASTER method in Farmer et al. (1999), and internal project deliverables (e.g., Riemersma et al., 1998; van Rooij, Barnard, Verstegen, Bermejo Muñoz, and Retamero Merino, 1998a; van Rooij et al., 1997). Further developments by the consortium are not expected as the MASTER project finished in 1998.

4.3.1 Product requirements for complete specifications

A) Simulator coverage
During TPD phase simulator coverage is explicitly defined: users choose which training activities will take place in the future simulator and which will not. Functional specifications are derived from prototypical training scenarios for those training activities. Thus, the method also helps users to narrow down the specifications to only those things that are necessary for training. It is easy to trace back for which scenarios -and consequently for which training activities and training goals- certain features are required.

B) Integration in the course
The result of the TNA is a comprehensive list of training goals representing all the knowledge and skills that trainees have to acquire to execute their future tasks. During the first half of the TPD users design an outline of the whole course and define training activities for all training goals. In principle, this guarantees that the simulator training is well integrated in the overall training trajectory. However, during the first evaluations prospective users commented that it may be difficult to keep an overview during the second half of the TPD phase when the focus is only on simulator training (see § 4.2.4.2).

C) The target group(s) of trainees
The entrance level and other characteristics of target trainees are analysed in the third step of the TNA (trainee analysis). Together with the results of the task analysis they are the basis for the formulation of training goals.

D) Individualisation of training
If fundamentally different groups of trainees are foreseen, the MASTER method assumes that the designers will formulate alternative sets of training goals and alternative training programmes. To allow for adaptation to smaller differences and individualisation of training within one group of trainees, different routes through the training programme can be specified during the TPD phase (e.g., when training goals are sequenced, when training activities are specified or when instructional interventions are planned). Assessment for this purpose can be planned as well. During the TMS phase, guidelines could remind users to include the necessary facilities for this purpose.

E) Instruction
At the beginning of the TPD phase, users are asked to select general training strategies and/or learning principles that will be applied during the design (and later the delivery) of training. In the last TPD step the kinds of instructional interventions that are foreseen before, during and after scenario execution are defined. During the TMS phase these descriptions are used to specify the facilities required for instruction. Guidelines and libraries could draw the attention to the opportunities to improve training and instruction in simulators.

F) Assessment of trainees
The evaluation of the trainees' performance has a prominent place in the first step of the TPD phase: users construct an assessment plan by specifying when assessment should take place, which kinds of performance data are required and how these should be collected. The assess-

ment plan is the basis for specifying the required facilities during the TMS phase. Guidelines and libraries could draw the attention to the possibilities for performance registration and more objective, (partly) automated assessment.

G) Instructor support
The draft of the training programme designed during the TPD phase gives insight into the tasks of future instructors and the facilities required to support instruction and assessment, individualisation of training, revision of training programmes, construction of new scenarios, etc. Guidelines and libraries could remind users of possibilities to (partly) automate these tasks or to provide support for instructors on the Instructor Operator Station. The required facilities are specified during the TMS phase. There is, however, no specific step that focuses on the instructors' tasks and needs from the instructors' point of view.

H) Quality assurance and evaluation of training
There are no specific steps or measures focusing on quality assurance and the evaluation of training. However, when all intermediate results are stored (see under Aspect J) they can be inspected at any time (see also § 4.3.2, under Aspect I). The assessment plan defined in the first step of the TPD phase should not only address the assessment of trainees but also the evaluation of the training programme itself. The assessment plan is the basis for the specification of facilities for the registration, (long term) storage and evaluation of performance data during the TMS phase. Guidelines and libraries could draw the attention to the necessity of these facilities.

I) Editing facilities for training programmes and scenarios
In the TMS phase the draft training programme and the prototypical scenarios are the input for the specification of databases with scenarios or scenario elements, and for the specification of editing facilities for the revision of training and the implementation of new scenarios. Guidelines and libraries could support the user. Elaborating and revising training programmes is facilitated by the storage of intermediate results from the TPD phase.

J) Documentation of specifications and specification process
The MASTER method and tools envision storage of not only the final specifications, but also of all the intermediate results of all the steps. Thus, information collected during the design can be reused at a later stage.

Conclusions
All the issues that need to be covered in training simulator specifications are addressed by the MASTER method. The necessary guidelines and libraries to support users and remind them of the opportunities to improve training and instruction in simulators have, however, not yet been formulated. Furthermore, the MASTER method does not explicitly address the instructors' tasks and needs from the instructors' point of view.

4.3.2 Process requirements to support the design task

In this section I discuss to what extent the MASTER method fulfils the process requirements to support the four main aspects of the instructional design task: gathering information (Aspects A-C), solving the design problem (Aspects D-F), managing the design process (Aspects G-I) and communication (Aspects J-K). These requirements were defined in Section 3.4.

A) Handling input information
The necessary input for each step is specified. In the prototype tools the information can be stored in relation to the step(s) for which it is relevant, thus helping users to keep an overview of the collected information and facilitating co-operation between different members of the design team. In the prototype tools a special resource window to store information about resources is always available.

B) Modelling the problem and solutions

In the prototype tools the output windows in each step allow some modelling of (parts of) the solution. The MASTER method helps the users to decompose the problem, narrowing it down to the part of training that will probably be provided with simulator training. Users can decide at any point to decompose the problem further, for example by focusing only on a subset of (critical or difficult) training goals during the TPD phase. It is also possible to design several alternatives and consider the consequences for simulator specifications. In the prototype tools alternative solutions can be described on the notepad, or saved under a different name.

C) Documentation

In the prototype tools intermediate results are stored together with the users' comments. The prototype tools can also be used for long term data storage. The clear relationships between input and output of steps make it possible to analyse the consequences of later decisions through 'backward chaining'. When certain features cannot be realised, users can trace which scenarios, and thus which training activities, cannot be executed and reconsider whether the related training goals can be achieved in another way. When designers make sufficient notes it will be possible to retrace the reasons for decisions, the alternatives that were considered, etc. Making notes, however, is not obligatory, and is not supported with specific guidelines or templates.

D) Strategic support

Some strategic support is given in the Manual and the so-called Training Cards that contain brief descriptions of all steps and sub-steps. Strategic advice can also be given in guidelines with each step, but these have not yet been elaborated.

E) Heuristics and guidelines regarding instructional design

The MASTER method provides a framework for context-specific help in the form of libraries and guidelines that are attached to specific steps and sub-steps. Guidelines and libraries can be used to make knowledge from literature and experience concrete and directly applicable during the design process. However, they have not yet been elaborated.

F) Decision-making support

Users can decide to design several alternatives and consider the consequences for simulator specifications. There are, however, no specific facilities to compare alternative solutions or to support decision-making.

G) Procedural support

The MASTER method provides a framework for the design of training simulator specifications, specifying which steps should be taken in the design process. Because the input for each step consists of the output of previous steps, the structure of the method is easy to understand and it is impossible to skip vital steps. Decisions are made explicit and the clear relationships between input and output help users to keep an overview of the impact of their decisions. The MASTER method can be adjusted to the users' organisational setting and preferences (see for an example Chapter 5), but the prototype tools are difficult to adapt.

H) Support to deal with iteration

The MASTER method enables an iterative design process where users can take preliminary decisions, make notes about assumptions and alternatives, and go back to previous steps when new information becomes available, or further elaboration gives more insight into the consequences of earlier decisions. To keep an overview users are strongly advised to go back to the relevant step, instead of making ad hoc changes later on. Users are also free to work on a more abstract level initially, or to work on sub-problems first (e.g., a specific set of training goals), and to go back to a step when further elaboration has become necessary.

I) Support for quality control
The MASTER method does not provide specific facilities for quality control, although the structured approach ensures that all important issues will be addressed (see also § 4.3.1). The storage of intermediate results and designers' notes and comments does make it possible to review the whole design process, also by people who were not directly involved in the design process.

J-K) Facilities to support communication
Apart from the facilities to store, organise and review information described above, the MASTER method and prototype tools do not specifically support communication within the team or with team members.

Conclusions
The MASTER method structures the design process and provides a framework for organising and storing all the data that designers collect. It also makes it possible to provide concrete, context-specific strategic advice to make up for possibly lacking instructional design and development knowledge at the moment that it is needed (although this has not yet been developed). It does not provide specific facilities to support decision-making, quality control or communication. See Section 9.4 for a discussion about facilities to support communication within a design team.

Although the MASTER method enables an iterative design process, it still looks mainly linear: the method prescribes a sequence of steps to be taken in a pre-specified order. It is possible to go back to previous steps to break up the design problem into sub-problems and/or to partly design and compare alternative solutions, but it is up to the user to decide whether this is necessary and how it should be done. Iteration is facilitated by the clear input/output relationships between steps and by keeping intermediate results and the user's notes available throughout the design process, but it is not actively encouraged or supported. The MASTER method does not, in other words, sufficiently support this aspect of managing the design process.

4.4 Areas for further elaboration and research

Given the discussion in Section 4.3 at least the following areas would need further research and development:

1) Empirical evaluation of the MASTER method
During the informal evaluations executed within the MASTER project, users did not have the opportunity to actually work with the method themselves. To prove its applicability and effectiveness for the design of specifications for training simulators, the MASTER method needs to be applied to realistic domains and empirically evaluated.

2) The content of libraries and guidelines
The concept of libraries and guidelines provides a good framework for context-specific user support, but they are still empty: only a few examples are given to illustrate their use. And even though libraries and guidelines are thought of as growing knowledge bases, and therefore do not have to be complete and conclusive from the start, a reasonably large basic set is required as a starting point. Information is available from the MASTER handbook (Farmer et al., 1999) and other research literature. However, distilling valid guidelines and libraries, formulated in a concrete and directly applicable way, will be a difficult and time-consuming job. Furthermore, guidelines can contain very different kinds of advice, such as: advice related to the specific decisions, strategic advice concerning how to collect information or execute a specific step, and process-oriented advice about monitoring the design process. To avoid confusion and to ensure that users can find the kind of advice they are looking for, guidelines should be structured and ordered.

3) Iterative design

Although the MASTER method enables an iterative design process, it is still up to the users themselves to decide when and how iterations are necessary, and to take the necessary actions. The optimal form of process management will be different for every project: if, when and how often users should go through iterative design cycles depends on many factors, including the complexity of the domain, the capacities of the designers, changes in the available resources, organisational factors, etc.

In the second part of this thesis I focus on the first and the third point: the empirical evaluation of the MASTER method and the iterative aspect of design. For my research I have used the Training Programme Design phase of the MASTER method, which was further elaborated and implemented in a prototype tool (see Chapter 5) to be used as a research instrument in evaluation studies (see Chapters 6 to 8).

Chapter 5
MASTER-TPD method and prototype tool

In Chapter 5, I describe the MASTER-TPD method and tool that will be used as research instruments in the evaluation studies that will be described in Chapters 6 to 8. Subsequently, I discuss the concept of iteration in more detail and describe different forms of support that have been implemented to help users manage iteration during the design process. The research question addressed in this chapter is:

7. How can designers be supported during the iterative design of simulator training programmes?

I will come back to this research question in Section 10.3.

5.1 Introduction

In Chapter 4, I described the MASTER method, an ISD-based method for the specification of training simulators. This method fulfils many of the requirements posed in Chapter 2 and Chapter 3, but not all of them (see § 4.3). In the second part of this thesis I will focus on one of the aspects that the MASTER method does not sufficiently support: managing iteration during the design process. Furthermore, I will evaluate the method's applicability and usability in empirical evaluation studies. For these studies I have used the Training Programme Design (TPD) phase of the MASTER method. Using only one phase, instead of all three, made it possible to let subjects work with a case of realistic size and complexity. At the same time, executing the TPD phase takes long enough to study iteration during the design process.

The TPD phase is important because in this phase users can get a clear view on all the advantages or disadvantages of using simulators instead of the operational system for training and instruction (see § 1.3.1) and plan facilities for instruction (see § 4.2.2 and § 5.3). In the MASTER handbook Farmer, Jorna, Riemersma, van Rooij and Moraal (1999) give the following arguments for executing the TPD phase:

- There is usually a relatively large gap between the global level at which operational requirements have been specified and the high level of detail of technical simulator specifications. Without the TPD phase it is difficult to provide sufficient underpinning of simulator specifications. This is a drawback especially when budgets are under pressure.
- For the training of many skills, only particular types of cues and/or limited level of cueing fidelity is (minimally) required. To determine this, the nature of the skills that are to be trained must be specified in sufficient detail.
- For training purposes it may be more effective or efficient to provide extra cues and feedback. These can only be identified on the bases of specification of appropriate training and instruction strategies.
- An accurate specification of what and how to train is required to choose between alternative learning solutions and instructional products, and to assess whether proposed instructional products meet all the demands.

Farmer et al. claim that the TPD phase is currently often not executed during the design of training simulators, probably based on the idea that the task analysis during the Training Needs Analysis (TNA) phase can directly lead to a description of training tasks. However, training tasks can differ from operational tasks, depending on the identified skill requirements or capabilities and the most efficient way of acquiring those skills (see also § 5.3.2). The results of the field study described in Chapter 2 showed that the opportunities to improve training and instruction in simulator training are currently not fully exploited and that the facilities to support instructors are often poor in existing training simulators (see § 2.4.2). For the purpose of my research the TPD method has been further elaborated and implemented in a prototype tool, to be used as a research instrument in the evaluation studies that will be described in Chapters 6 to 8. For this version the names MASTER-TPD method and MASTER-TPD tool are used in this thesis.

In Section 5.2, I briefly describe the implementation of the MASTER-TPD prototype tool. In Section 5.3 the steps of the MASTER-TPD method, as implemented in the prototype tool, are described in more detail. Subsequently, in Section 5.4, I discuss the concept of iteration in the light of the MASTER-TPD method and I describe four types of help that have been implemented to support the management of iteration during the design process: process-oriented guidelines, warnings, hints and the definition of global and detailed design phases. For the evaluation studies a limited set of guidelines and some strategic support is included for each step, but since this was not the focus of my research no claims are made for their validity or completeness.

5.2 The MASTER-TPD prototype tool

In this section I briefly describe the implementation of the MASTER-TPD tool. See Appendix A, Steutel (1999) and Verstegen, Steutel and Barnard (2000) for a more elaborate description.

5.2.1. Background

The prototype tools that were developed within the MASTER project could not be used for my research for three reasons. First of all, the prototype tools were not always stable, and as the project was finished no further improvement of the software was to be expected. Secondly, they were not consistent with the final version of the MASTER method (as described in Farmer et al., 1999). Finally, it was not possible to implement additional support for iteration. Therefore, other options were investigated. For the purpose of my research a tool should fulfil the following demands:
- guide users through the steps of the TPD phase,
- store, organise and present final results, as well as intermediate results,
- allow users to make notes with each step,
- register the design activities of users with time-stamps,
- enable the implementation of guidelines, libraries and an electronic manual, and
- enable the implementation of additional support for managing iteration.

Since the tool was intended to be a research tool to be used for research purposes only, some imperfections were acceptable. Using Commercial Off-The-Shelf (COTS) software was preferable because time and financial resources were limited.

The prototype MASTER-TPD tool is a customised version of Designer's Edge (DE), a well known commercial tool to support the instructional development process (DE 2.0, 1997, see also § 3.3.8.1). DE 2.0 follows a step-by-step method based on the ISD approach, mainly focusing on the design of CBT and the development of scripts for CBT lessons. Help is available in the form of on-line help and Training Cards describing what to do in every step. For a number of steps wizards are available to help users to insert new data in the form of objects with a number of attributes. For example: when a new training goal is created the wizard asks the user for a name, a description, the minimal performance level, etc. Users can tick the tick-box beside a sub-step to indicate that they have completed it. It is always possible, though, to go back to previous steps to review or elaborate on work that has already been done. Figure 5.1 shows a screen print of DE 2.0.

DE 2.0 stores, organises and presents all final and intermediate results, and it can be customised in several ways (see Appendix A). An electronic manual is available and its content can be adapted, and in some steps libraries are available as well. The ISD method implemented in DE 2.0 can be adapted by reordering existing DE steps and/or by implementing custom-made steps. Two problems were encountered during the implementation of the MASTER-TPD tool in DE 2.0:
- the names and attributes of objects that are used in DE steps cannot be changed, and
- not all objects and functionalities are available in all steps.

The first problem was solved by using objects with plausible names and, where necessary, explanations in the help for the steps concerned. To solve the second problem the TPD method was slightly adapted. An extra step was introduced to construct a Course Map with only those training activities that will be covered with the simulator. This extra step reflects the formerly implicit boundary between the first part of the TPD method, that concerns the whole course and the whole set of training goals, and the second part of the TPD method, that concerns only the part that will be covered with simulator training. In this extra step users can choose between 'minimal set of requirements' or 'maximal simulator coverage'. In this way, the formerly implicit decision between a need-driven and a constraint-driven approach (see

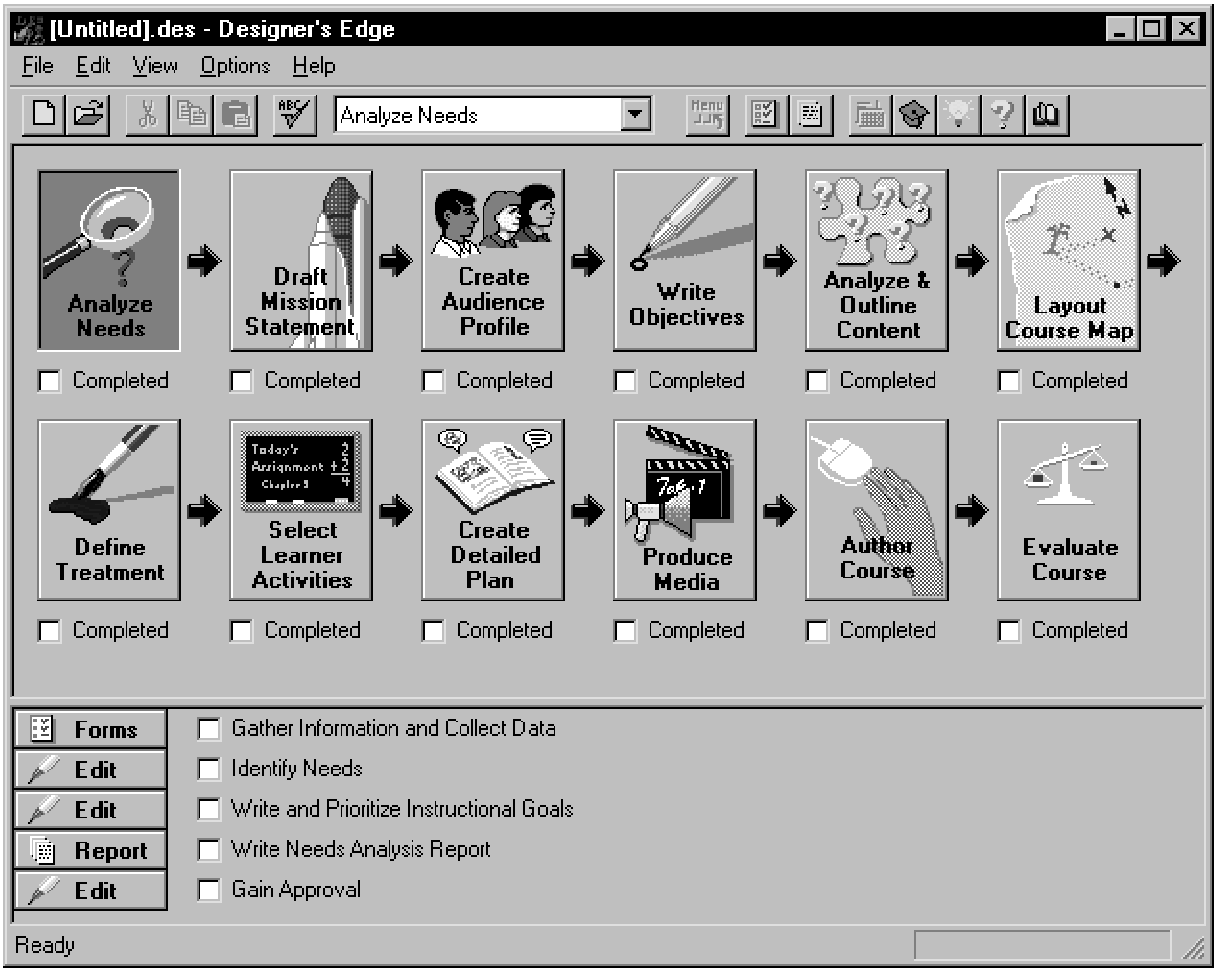


Figure 5.1: Screen print of Designer's Edge (DE 2.0, 1997): the 12 big buttons in the top half are the main steps, the lower half of the screen shows the five sub-steps of Step 1.

§ 4.2.2) is made explicit. The user is asked to decide whether to start with a core set of training activities or to focus on all training activities that are suitable for simulator training simultaneously.

A second change was to move the step 'Planning of assessment' forwards to the second part, since suitable objects are only available there. Moving this design activity does not violate the reasoning underlying the MASTER method. There are no indications that it caused problems during the evaluation studies (although some remarks of subjects indicate that this might still not be the optimal place for this design activity, see § 8.3.2.2 and the footnote with § 9.3.1). Finally, the library of training activities was implemented as a list of examples in the guidelines, since DE did not provide facilities to implement libraries in the first half of the method. This means that the users could not add their own training activities to the library. However, the concept of libraries was introduced in the MASTER project to guarantee ease of use and the reuse of data over longer periods of time. As this prototype is meant for use during research studies only, this knowledge management aspect does not play an important role (Note that expandable libraries have been implemented for other steps, see § 5.2.2).

DE 2.0 (1997) provides no explicit facilities to make notes, to register user actions, or to implement specific support for managing iteration. An external application that runs at the same time as DE 2.0 has been implemented to fulfil these demands. This external application provides a Notepad for each sub-step and it registers the user's actions in a log file. When existing DE sub-steps are used, the database is monitored and when objects are added or removed this is registered. The custom-made sub-steps write statements to the log file directly.

Thus, the following events are logged with a time stamp:
- Adding an object to the database.
- Removing an object from the database.
- Start of a sub-step for the custom-made Steps 1.1, 1.3, 2.2 and 5.2 (see Figure 5.4).
- End of a sub-step for the custom-made Steps 1.1, 1.3, 2.2 and 5.2 (see Figure 5.4).
- Opening a help screen.
- Closing a help screen.
- Opening a specific page of the manual.
- Selecting the Notepad for a specific step.
- Appearance of warnings (see § 5.2.2).
- Sending an e-mail (see § 5.2.2).

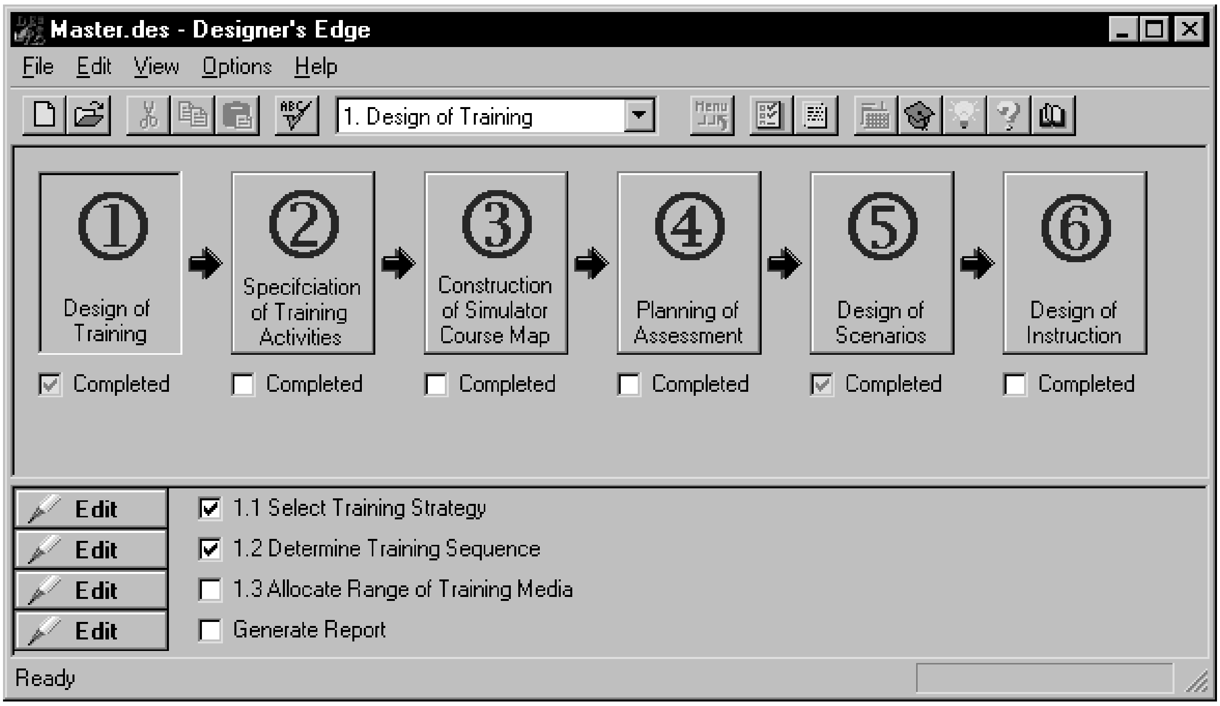


Figure 5.2: The prototype MASTER-TPD tool, a customised version of DE 2.0 (1997).

Figure 5.2 gives an overview of the main steps in the MASTER-TPD tool. In all the main steps of the MASTER-TPD method the existing DE report-facility has been added as a non-obligatory sub-step. This sub-step provides users with an overview of their work so far (in the form of a simple text document).

5.2.2 Help in the MASTER-TPD tool

The MASTER consortium stated that different types of help should, as much as possible, be directly linked to steps of the MASTER method in order to make the advice concrete and immediately applicable. This principle has also been applied in the MASTER-TPD tool described here. For the evaluation studies additional support for managing iteration was implemented. This results in the following forms of support in the MASTER-TPD prototype tool:

- Training Card: Short description of each sub-step.
- Tell-me-how: Description of how the tool works and which actions should be taken to complete the work in a specific step.
- Libraries: Expandable libraries or lists of typical answers or solutions, in Steps 1.1, 1.3, 2.2, 4.2, 6.1, 6.2 and 6.3 (see Figure 5.4).
- Guidance: Process-oriented guidelines focusing on the iterative aspect of design.
- Advice: Other kinds of guidelines regarding the work in a specific step.

- Notepad: Facility to take notes about the work in a specific step.
- Hint: Reminder about issues to take notes on in a specific step (based on guidelines provided in Guidance).
- Warning: Pop-up screen triggered by measurable events in Steps 1.1, 1.3, 2.2, and 5.2 (see Figure 5.4). Due to technical problems the specified warnings for the other steps could not be implemented.
- Manual: Containing three parts:
  - General: more elaborate description of all steps.
  - Guidelines: all the guidelines provided in Guidance and Advice.
  - Actions: tool description also provided in Tell-me-how.
- Experts: Facility to e-mail questions directly to different kinds of experts.
- Discussion: Facility to discuss with peers in web-based discussion groups.

See Section 5.4 for a more elaborate description of the content of Guidance, Hints and Warnings, and the Appendix of Verstegen, Steutel, and Barnard (2000) for a complete description of the content of all types of help for each step. Note that, since the prototype tool is meant to be used for research purposes only, there is no claim of completeness or user-friendliness of the different types of help or their content. Especially the content of the Libraries and the guidelines under Advice is by no means exhaustive. They just contain a set of examples sufficient for the empirical evaluation studies.

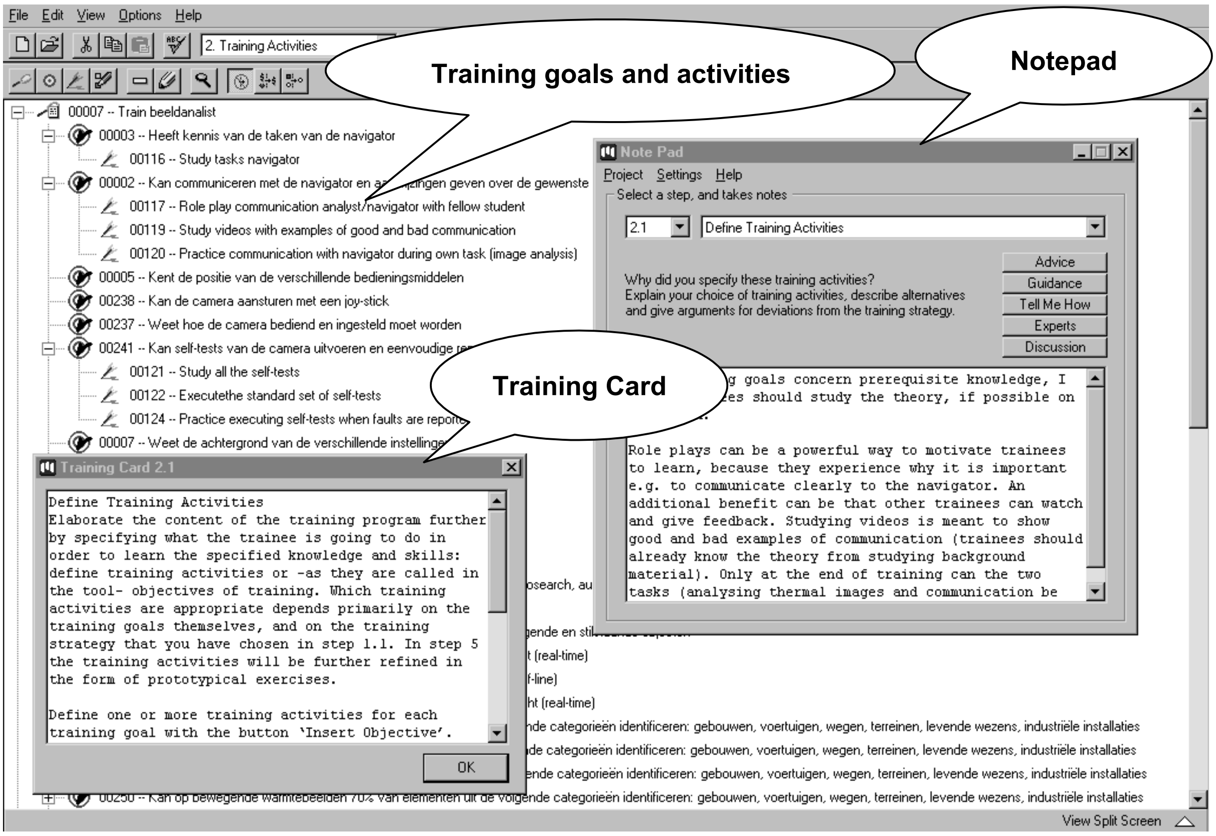


Figure 5.3: The prototype MASTER-TPD tool with the:
- the work field of Step 2.1 (At the back; training goals and activities in Dutch),
- the Training Card of Step 2.1 (Lower left window), and
- the Notepad of Step 2.1 with hint and help buttons (Lower right window).

In the MASTER-TPD prototype tool these help forms have been implemented in the external application that also provides the Notepads and takes care of data registration. Users can access the two kinds of guidelines through buttons on the Notepads. To maintain consistency, access to other context-specific help forms is also arranged through the Notepads. The Training Card appears automatically when the user accesses the Notepad of a specific step. Hints are placed directly on the Notepad, and the Manual can be found under the Help-menu.

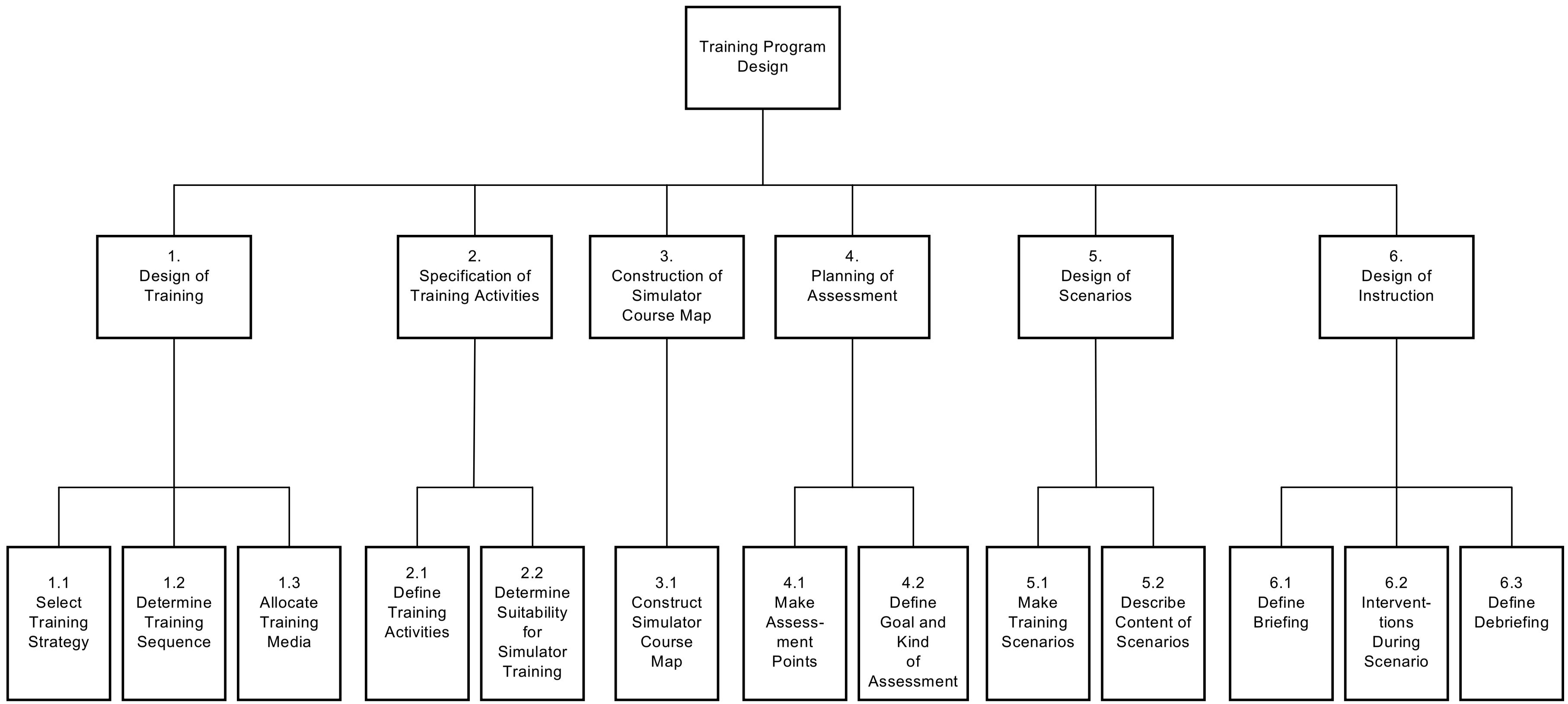


Figure 5.4: Steps and sub-steps of the MASTER-TPD method, as implemented in the prototype MASTER-TPD tool.

Advice, Guidance, Tell-me-how, Experts (i.e. forms to send e-mail to different kinds of experts) and Discussion (i.e. a link to the web site) are available under buttons on the Note-pads. Warnings pop up automatically in a separate screen and the libraries are implemented in those sub-steps where they apply (see above). Figure 5.3 shows a Notepad with the different kinds of help.

The MASTER-TPD prototype was implemented to be used as a research instrument during two evaluation studies (see Chapters 6 to 8). In the first study some subjects used a tool with a slightly different interface: the buttons on the Notepads where not available, but all information was provided in the Manual. The two versions only differ in the way that the different types of help are provided. There is no difference in the kind of support offered, nor in the content of the help. Since no important differences were found between subjects using the two versions they are not further discussed in this thesis (see also § 6.3.3, and for a detailed report Verstegen, Barnard, and Pilot, 2003).

5.3 The MASTER-TPD steps

In this section I describe the MASTER-TPD method as implemented in the prototype MASTER-TPD tool. Definitions of the terms used in this section can be found in the glossary that was also used during the evaluation studies (see Appendix B).

The goal of the TPD phase is to specify how training will be provided (see § 4.2.2). The input for this phase is the output of the Training Needs Analysis (TNA): a set of training goals and a description of the target group(s) of trainees. The output is a draft training programme with -for the part that will be covered with simulator training- a set of prototypical training scenarios and descriptions of the kinds of instructional interventions that are envisioned (see Figure 4.1). Figure 5.4 gives an overview of the six steps of the MASTER-TPD method as implemented in the in the prototype tool:

5.3.1 MASTER-TPD Step 1: Design of Training

In the first MASTER-TPD step an outline of the training programme is constructed. First, in the sub-step Select Training Strategy, one or a few important training strategies and/or learning principles are selected. This step forces the users to explicitly choose one or more ID models (see § 3.3.2). Applying these strategies or principles throughout the design (and later the delivery) of training should lead to a coherent approach suitable for the kind of trainees and the kind of training goals to be achieved. Users are free to define their own strategies or principles, but the MASTER-TPD tool also provides a list of suggestions (see Figure 5.5) and some information about their applicability.

Subsequently, in the sub-step Determine Training Sequence, the order in which training goals will be addressed is determined. The guidelines describe some sequencing principles that can be used, such as 'Start with knowledge and skills that are prerequisites for others', or 'Group training goals concerning the same kind of knowledge'. If necessary, intermediate training goals are inserted, for example when the step between two consecutive training goals is too big or when users decide to address only a part of the training goal early on and postpone the rest to a later moment. If the sequence of training goals should be adaptable to the training needs of individual trainees, the possible deviations are described as well.

In the final sub-step, Allocate Range of Training Media, the users decide which range of instructional products could be used for each training goal. Choosing a wide range of options at this stage will make it easy to select a reasonable alternative when the first choice turns out to be not feasible. Descriptions of different kinds of instructional products and their applicability, advantages and disadvantages are given in the guidelines. Further decisions are post

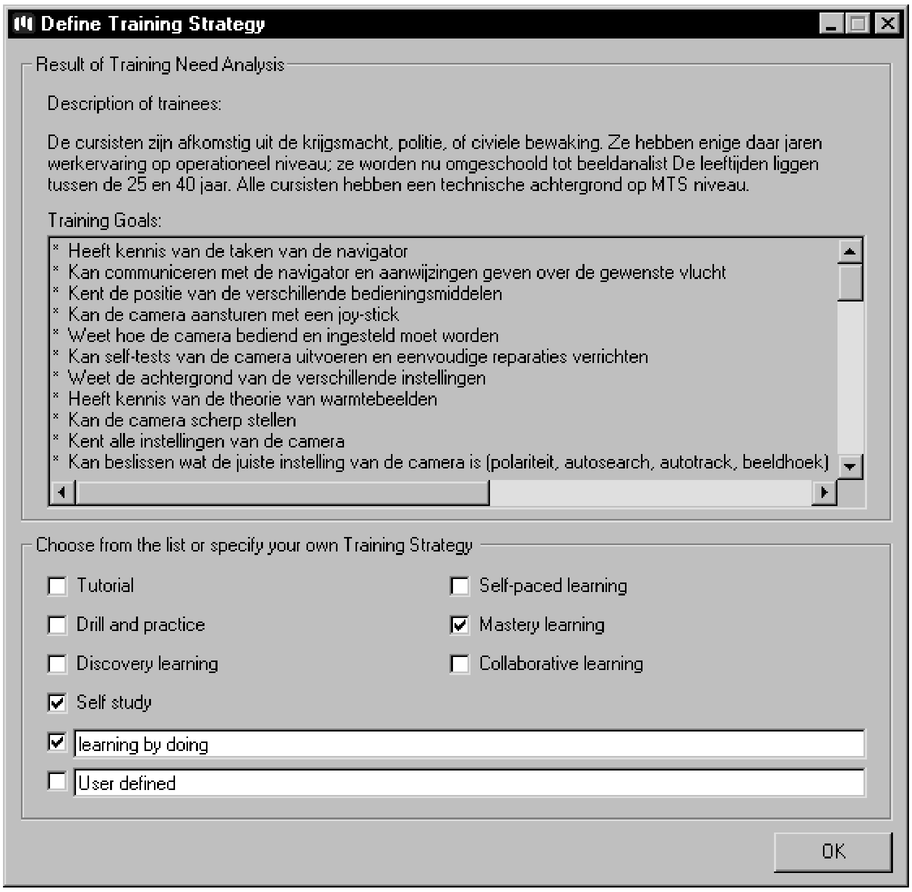


Figure 5.5: The custom-made sub-step Select Training Strategy; an expandable list of examples functions as library (N.B.: The training goals and description of trainees are in Dutch, see § 6.2 for a description of the case used during the evaluation studies).

5.3.2 MASTER-TPD Step 2: Specification of Training Activities

The goal of Step 2, Specification of Training activities, is twofold. First the content of training is further elaborated by specifying what trainees will do in order to achieve the training goals: one or more training activities are defined for each training goal (see Figure 5.3 in § 5.2.2). Training activities can be similar to the future tasks of the trainees. However, activities that never occur in reality can also have a high learning value, e.g., discovering how a system works, or changing roles with another team member to experience when and why others rely on your actions. The guidelines provide abstract descriptions of a number of training activities, their applicability for different kinds of trainees and training goals, and their applicability in the context of different training strategies (chosen in Step 1.1). These abstract descriptions still need to be refined by filling in domain specific information, e.g., 'practice-until-automation' into 'practice the starting procedure under different environmental conditions, including sun, rain, wind and snow, until executed without errors three times'.

The second sub-step is process-oriented. Up to here the MASTER-TPD method covers the whole task domain and all training goals. In the sub-step Determine Suitability for Simulator Training the focus shifts to the part that will be covered with the future training simulator. Training activities that are not suitable for simulator training are labelled 'not training simulator'. Training activities that cannot be safely or efficiently trained without a training simulator are labelled 'certainly training simulator'. The group that remains are training activities that can be trained with a simulator, but for which reasonable alternatives are available. These are labelled 'maybe simulator training'. Whether they will be covered with the simulator or not will depend on, amongst others, available resources and logistic factors.

5.3.3 MASTER-TPD Step 3: Construction of Simulator Course Map

The extra step Construction of Simulator Course Map has been introduced mainly for technical reasons (see § 5.2.1 and Appendix A). A second goal of this step is to allow users to vary their route through the remaining steps (cf. § 4.2.2): they can decide to work on specifications for an ideal simulator, or to go for a minimal configuration initially. In the first case they choose for 'maximum training coverage' and all training activities that can be trained with a simulator, labelled 'certainly simulator training' or 'maybe simulator training', are included in the Course Map. When users discover later on that the ideal simulator is not feasible they will have to go back to decide whether the training goals that cannot be executed with the simulator can be achieved in another way, i.e. whether the training activities can be executed elsewhere (Step 2.2) or whether the training goals can be achieved with other training activities (Step 2.1). Choosing 'minimal set of requirements' is the alternative route: only training activities that have been labelled 'certainly simulator training' are included in the Course Map. Later on, of course, users can come back to this step to decide whether a wider range of training activities can be executed with this 'minimal' simulator or whether the available resources permit the extra facilities required to cover a larger part of training.

5.3.4 MASTER-TPD Step 4: Planning of Assessment

The evaluation of the trainees' performance is an important aspect of training. Performance data are also necessary to evaluate and improve the training programme itself. Assessment is planned explicitly to make sure that the required facilities will be specified in the TMS phase. In the first sub-step, Make Assessment Points, the users decide when assessment is necessary.

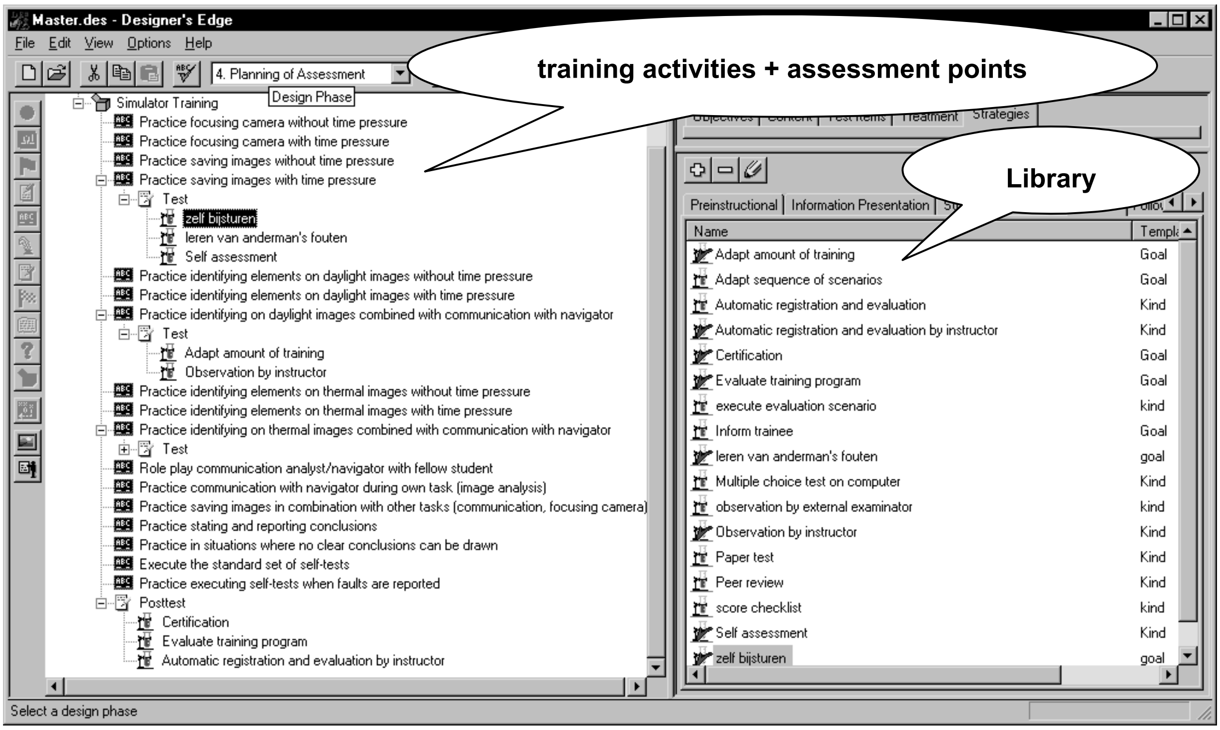


Figure 5.6: The planning of assessment in the Course Map (used in Step 4, 5, and 6) with:
- the work field of Step 4.2: the training activities allocated to simulator training with assessment points in between (Left window), and
- the Library with examples of goals and kinds of assessment (Right window).

The guidelines explain, for example, that likely places are at the end of training (end criterion), at the beginning of the part covered with the simulator (entrance test), at the end of a group of training goals, or at points where the training programme can be adapted to the needs of individual trainees. Figure 5.6 shows the planning of assessment in the Course Map used in Step 4, 5, and 6.

In the second sub-step, Define Goal and Kind of Assessment, the users decide how the performance of trainees should be evaluated based on the goal(s) of the assessment. For formal assessment at the end of training, for example, the registration of objective data may be required. On the other hand, informal observations of the instructor can be sufficient for the adaptation of the training programme. Guidelines also stress the possibility to decrease the workload of instructors by automating (part of) data registration and assessment. Note that the goal of this step is to get insight into the facilities required for assessment. Therefore, it is not necessary to specify assessment procedures in detail.

5.3.5 MASTER-TPD Step 5: Design of Scenarios

In Step 5 one or more exercises -called scenarios in the context of simulator training- are specified for each training activity that will be covered with the future simulator. In the second sub-step the content of scenarios is further described in three standard slots (see also Figure 5.7):

- a process description specifying which events will occur during the scenario (on a time-line) and which actions trainees are supposed to perform,
- an environment description specifying the relevant environment characteristics, and
- a system description, specifying which features and functions of the real system should be simulated.

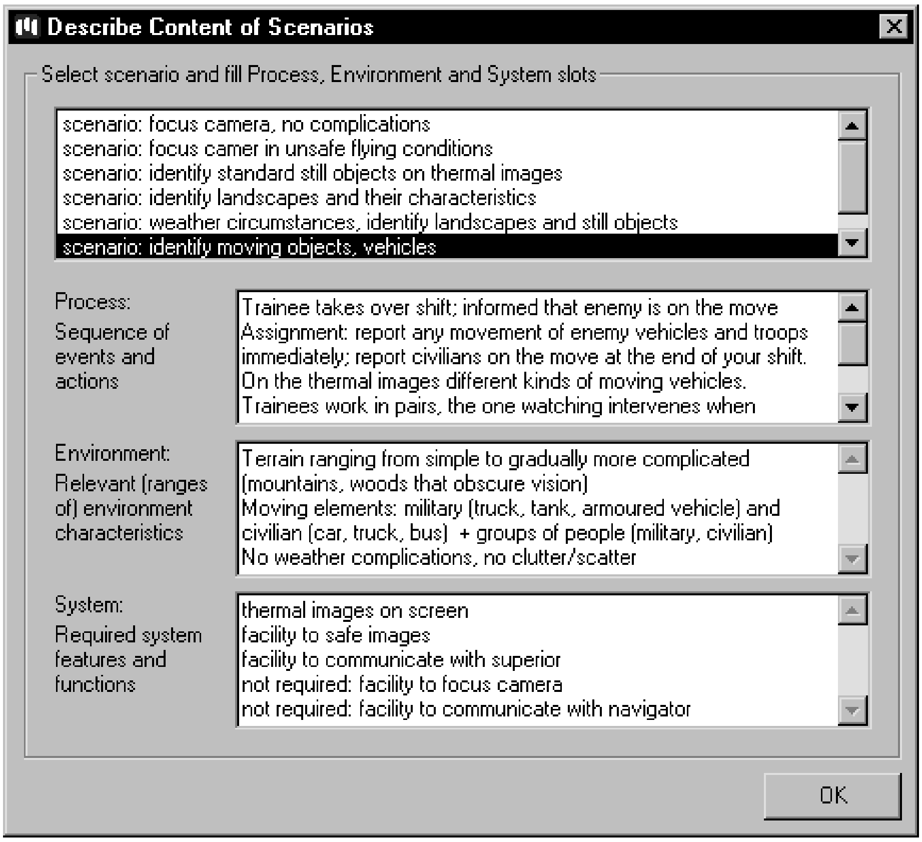


Figure 5.7: Custom-made sub-step for the description of the content of training scenarios in terms of process, environment and system characteristics.

Scenario design is meant to get insight into the training simulator facilities required for the execution of training activities. For this purpose, scenarios do not have to be specified at the detailed level that is required for implementation. A more abstract description will suffice, provided that it contains enough information about the required behaviour of the future training simulator to derive specifications during the next phase, the Training Media Specification (TMS).

5.3.6 MASTER-TPD Step 6: Design of Instruction

The step Design of Instruction is meant to get insight into the kind of instructional interventions that will be necessary to make simulator training effective and efficient. Different kinds of instructional interventions may be appropriate before, during or after a training scenario. Therefore, these are specified separately in three sub-steps that can be completed in any order. Examples of instructional interventions are provided in libraries (one for each sub-step). Guidelines address the applicability and timing of different kinds of interventions, and their applicability in the context of different training strategies (chosen in MASTER-TPD Step 1.1).

Instructional interventions are planned explicitly to ensure that the required facilities will be included in the simulator specifications in the TMS. For this purpose, the focus is not on the content of instruction, but on the way they are going to be provided: which kinds of interventions are foreseen, who is going to be responsible for them, and how they are going to be provided to the trainees. Again guidelines stress the possibility to decrease the workload of instructors by automating (parts of) data registration and assessment.

5.4 Support for iteration

In Chapter 3, I concluded that designing simulator specifications is not a linear process (see § 3.2.3). The design process is disturbed by many 'practical' factors, such as conflicting and changing constraints, interference from others, and new information. There is also a more fundamental reason for iteration: in an ill-structured design problem the sub-problems are not unrelated and can, therefore, never be solved in isolation. Moreover, it is impossible to know beforehand which ideas will lead to a good solution and which not. The MASTER method does not sufficiently support this aspect of the design process (see § 4.3.2). In this section I first discuss the concept of iteration in more detail and describe how the MASTER method deals with iteration. Subsequently, I describe additional support for managing iteration that has been implemented in the prototype MASTER-TPD tool.

5.4.1 Iteration and the MASTER method

In a general sense iteration means: executing activities again (van Wagenberg, 1992). The simplest form is the repetition of the same activity, e.g., users can adapt the thermostat of the heating according to their wishes once or more every day. Iteration can, however, also mean monitoring the results of one activity and -based on those results- choosing the next activity out of a range of activities. In the context of specifying training simulators with the MASTER method I define iteration as going back to a design activity that has already been executed. There are different ways to iterate and different reasons for iteration (see Figure 5.8). Designers can, for example, cycle through all the steps at a more abstract level first, and then again in more detail, or decompose the design problem and, for example, design training for a part of the training goals first and turn back later on for the remaining part. They can also jump back and forth between steps, e.g., because the design fails to satisfy one or more of the requirements, or because new requirements have emerged during the design process.

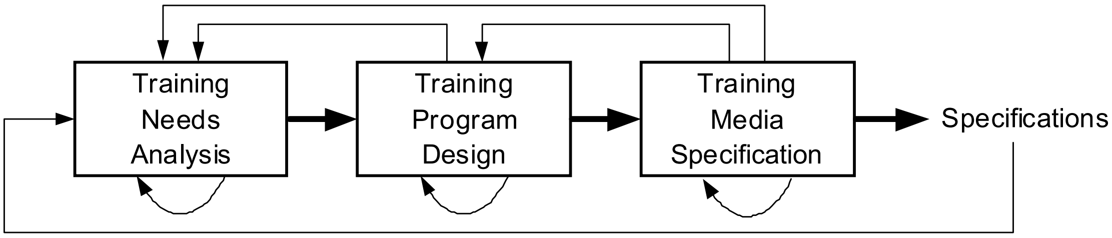


Figure 5.8: Different ways to iterate in the MASTER method: going through all the steps and then start at the beginning, going back to a previous phase or to a previous step in the same phase.

Van Wagenberg differentiates between four types of iteration that can also play a role during simulator specification with the MASTER method:

1) Iteration as repetitive activity
Iteration as repetitive activity concerns repeating the same design activity, for example for different sub-problems or at different levels of detail. The MASTER method allows designers to execute a step partly or not in full detail, and to return to it later. The most obvious example is that designers choose in MASTER-TPD Step 3 to focus only on those training activities labelled 'absolutely simulator training' and come back later to look at those labelled 'maybe simulator training'. In other steps it is also possible to decompose the design problem.

2) Iteration to correct errors
The necessity to correct errors is often seen as unavoidable, especially for novice designers. It is also likely to be necessary during training simulator specification, especially since those responsible are usually not experienced instructional designers (see § 3.2.1). Another reason for this kind of iteration can be that designers discover that they have based decisions on assumptions that are no longer valid.

3) Iteration to improve the design
Iteration to improve the design concerns fundamental revisions to get to a better design. This form of iteration is actually desirable: designers should be encouraged to keep on reviewing their design in order to find new directions or alternative solutions and further improve the final product. Elaborating alternative solutions in parallel is possible with the MASTER-method, if they are all saved under a different name. The MASTER method does not explicitly support comparing alternatives or decision-making (see § 4.3.2).

4) Mutual iteration
Mutual iteration happens when designers decide to execute two separable activities in parallel; this form can also be seen as a form of decomposition. An example of this form of iteration is the simultaneous design of specifications for training simulators (with the MASTER-TPD method and tool) and other instructional products, such as CBT (with other methods and tools). These two design processes are likely to influence each other, e.g., because the training activities that are not covered with simulator training need to be executed elsewhere (and vice versa).

Based on literature research in the domain of mechanical engineering and the design of chemical plants, van Wagenberg (1992) suggests that the desirable type and amount of iteration in a design process depend on the project's goals. A high amount of iteration, for example, supports the goal of improving the quality of the design and the goal of innovation, but it conflicts with minimising the design time and costs. Repetitive iteration supports the learning process of the designers and, thus, to certain extent, the goal of minimising the design costs. The relationship between project goals and iteration is not one-on-one, and can change during the design process, e.g., from mutual iteration and iteration to improve the design in the first phase (to enhance innovation) to mostly corrective iteration at the end of the design process.

Ideally, van Wagenberg claims, the definition of project goals leads to a conscious planning of iteration in the design task, and related to this issue, decomposition of the design task. Instrumental factors to reach and maintain the desired situation are, for example, the choice of design methods and tools and the composition of the design team. Van Wagenberg also proposes more concrete guidelines, that, however, still need to be further supported by research (see also § 7.5.2 and § 10.5).

The MASTER method structures the design process by specifying a sequence of steps that should be taken to design training simulator specifications. The method is constructed in such a way that there is a clear relationship between the steps: the output of one step is the input for the next, etc. The MASTER method envisions an iterative design process: decisions are made explicit and intermediate results are stored, so that users can go back to review decisions or elaborate their work. Designers are allowed to take notes and these are stored with the intermediate results (i.e. with each step) to ensure that the reasons for decisions can be retraced and can be reviewed in the light of new information. To maintain consistency and to make sure that designers can oversee the consequences of changes, the MASTER method does not permit direct changes of the final design. Users are obliged to go back to the steps that are affected by these changes. For example, when designers are working on the content of a training scenario in MASTER-TPD Step 5.2 and they want to change the place of this scenario in the training sequence, they have to go back to MASTER-TPD Step 1.2 to change the sequence of training goals and to go through the intermediate steps to adapt their design to the consequences of these changes.

Iteration is also possible by going through the whole method at different levels of detail: users can choose to go through the steps quickly, doing only global analyses and taking preliminary decisions in order to get a first set of global specifications. This may suffice, for instance, to write a needs statement at the beginning of the acquisition trajectory (see § 3.3.6). Later on users can choose to work on a more detailed level, to get the more detailed training simulator specifications. The data of a first, global run through the steps can be reused later on when elaborating the design in further detail.

Thus, the MASTER method allows iteration and supports it by keeping intermediate results and the users' notes available. It does, however, not actively encourage an iterative design process or support managing iteration. It is up to the users to decide at which level of detail they work, and at which moments they turn back to previous steps. In Section 5.4.2, I discuss the basis for developing support for dealing with iteration. Subsequently, in Section 5.4.3, I provide some examples of that support as implemented in the MASTER-TPD tool.

5.4.2 Support for dealing with iteration

Experienced instructional designers have the problem solving skills necessary to organise and monitor the design process (see § 3.2.2.3). They adapt the design process to the design problem at hand and to their own preferences (see § 3.2.2.1). However, the designers of functional specifications for simulators are usually not experts (see § 3.2.1), and descriptive research shows that novices are not very good at managing their own design process (see § 3.2.2.3). Current instructional development models do propose an iterative development process, but they do not indicate when and how iteration should take place (see § 3.3.3.3).

The optimal way of designing training simulator specifications depends on characteristics of the specific design task and the designers. It will be different for every project: if, when and how often iteration is desirable depends on many factors, including the complexity of the domain, the capacities and the preferences of the designers, the availability of information, possible changes in the available resources, organisational factors, etc. As all instructional design problems, the specification of training simulators is an ill-structured problem (see § 3.2.2.2). The management of the design process cannot be automated or formulated as an algorithm, and it is not possible to prescribe a fixed sequence of steps or fixed moments for iteration. Therefore, the only way to encourage and support an iterative design process is to help users to manage their own design process better. For this kind of support, I use the term process-oriented support. Process-oriented support should help users to keep track of what they are doing and to decide what to do next. For example, by making the relationships between different steps in the design process explicit, helping users to recognise triggers that might necessitate iteration and making it easy for them to go back to previous steps in the design process.

Documenting not only the (intermediate) design products but also the design rationale, is important in an iterative design process for several reasons:
- to maintain consistency,
- to (re)use all the information and arguments that are available (see also § 3.3.7.3),
- to explain or defend design decisions, and
- to make it possible for somebody else to take over (when designers are working in a team).

Especially in an iterative design process, it is important that designers make notes immediately when they make decisions, instead of at the end of the design process (Boy, 1999; Boy and Barnard, 2003). When designers go back to review earlier decisions, they need to know the reasons for the initial decision, the assumptions that it was based on, the alternatives that have already been considered, etc. Notes are even more important when more than one person is involved. In this case, designers may have to review or elaborate work that was done by another member of the team. There are indications that novices are not proficient in representing design problems to themselves and others, and not aware of why they made certain decisions (Kerr, 1983; see also § 3.2.3). Therefore, process-oriented support should include helping users to make adequate notes with each step in the design process.

In the prototype MASTER-TPD tool four types of process-oriented support are implemented:
1. Hints on the Notepads to help and encourage designers to make adequate notes.
2. Process-oriented guidelines to help users to manage their own design process.
3. Warnings when events that should trigger iteration can be automatically recognised.
4. Definition of design cycles: in each step, two design cycles are defined explicitly to help users to decide how far they should go into detail to get to global specifications, and later on to get to more detailed specifications.

Process-oriented support should help the users to take process-oriented decisions, supporting a methodological and systematic, and at the same time iterative, design process. The challenge is to offer concrete advice at the right moment and to clearly structure the design process, while at the same allowing and supporting users to organise their own work flow in different ways, depending on the characteristics of the project and their own preferences.

5.4.3 Implementation of support for iteration

Based on the principles described in Section 5.4.2, extra help has been implemented in the MASTER-TPD tool to support users in managing an iterative design process. In this section some examples are given (see Figure 5.4 for an overview of the steps of the MASTER-TPD method). A full description of the process-oriented support for all MASTER-TPD steps is given in the Appendix in Verstegen, Steutel, and Barnard (2000).

5.4.3.1 Process-oriented guidelines

Special guidelines were developed to help users to manage their own design process and to take process-oriented decisions. These process-oriented guidelines focus on the iterative aspect of design and are separated from other kinds of guidelines under the name Guidance. For each step they are divided into process-oriented guidelines for the global design phase and process-oriented guidelines for the detailed design phase (see § 5.4.3.4). Some process guidelines address the relationships between steps and explain when it may be necessary to go back and review or elaborate on previous work. For example:

(Step 2.1) "Sequence your training goals in a way that suits your training strategy. Go to Step 1.1 to look at it, if necessary."

(Step 2.2) "The training activities that are marked 'absolutely simulator training' will be the basis for the design of a minimum set of specifications. Training activities that you would prefer to execute in the future simulator as well (marked as 'maybe simulator training') will be included as far as your resources permit."

(Step 5.1) "Making training scenarios can give you reasons to reorder the training goals, e.g., when you discover that mastery of other training goals is a prerequisite for the scenario that you are defining. Go back to Step 1.2 to review the training sequence. Look at your notes before you make any changes."

(Step 5.1) "You can elaborate each training scenario immediately (in Step 5.2) or make scenarios for all training activities first and then fill the process, environment and system slots in Step 5.2."

(Step 6.2) "If you have to deviate from you overall training strategy a lot, you may have to review your choice. Go back to Step 1.1 to do this. Look at your notes before you make any changes."

Other process guidelines give strategic advice that might help to keep an overview and manage the design process. For example:

(Step 1.2) "If the set of training goals is very large, it is more efficient to determine a rough order first and then look at the more detailed level of single training goals."

(Step 1.2) "Postpone the definition of intermediate goals to the detailed design phase."

(Step 1.1) "Ask domain experts for clarification, if the descriptions of the training goals are not clear or not detailed enough."

(Step 1.3) "Look at the training goals one by one. Select all suitable instructional products. Include all reasonable alternatives, but discard unrealistic solutions."

Finally, some of the guidelines stimulate designers to take notes, for example:

(Step 1.1) "Give arguments for your choice in the Notepad; if you find out later that, for example, the entrance level of trainees will be different, you can come back and see if this is a reason to change your training strategy. You can also use these arguments to defend your choices to other designers or to your superiors."

(Step 1.2) "Use the Notepad to describe doubts and possible alternatives. Later on, you may want to use this information when you are reviewing your training sequence."

(Step 6.1) "Use the Notepad to give arguments for any deviations from the overall training strategy."

5.4.3.2 Hints

Hints placed directly on the Notepad with each sub-step. They are another way to encourage and help designers to take adequate notes. Hints are derived from process-oriented guidelines and are given in the form of direct questions, e.g.:

(Step 1.2) "Why have you chosen this training sequence?
Where have you inserted intermediate goals and why?
Describe which sequencing principles you have used; describe alternative sequences and take notes, if you have doubts."

(Step 6.3) "Why have you selected specific debriefing elements for these scenarios?
Explain your choice of debriefing elements, describe alternatives and give arguments for any deviations from the training strategy."

5.4.3.3 Warnings

In a number of steps, explicit warnings have been implemented. When they are triggered, they pop up in a separate screen. For instance, when only a small number of training activities is labelled 'certainly simulator training', users are urged to reconsider whether it is necessary to acquire a training simulator:

(Step 2.2) "It seems that the majority of your training can be provided without a simulator. Do you really need a simulator? Or can the remaining part be trained during field exercises or on the job?"

Another example is the reminder that pops up after working one hour on training scenarios:

(Step 5.2) "Elaborating scenarios is a lot of work. Ask domain experts, instructors or instructional design specialists for advice, if necessary."

5.4.3.4 Explicit definition of global and detailed design.

To encourage users to work in an iterative way two design cycles are explicitly defined: global design and detailed design. Separate sections are included in the description of each step in the Manual, the Training Card and in process guidelines (see § 5.4.3.1), describing what users should do in each step in the global design phase and what they should do in the same step when they are working on a more detailed design. An example is the Training Card of Step 1.2 (Determine Training Sequence). After an introduction of the step the two design cycles are described separately:

(Step 1.2) "....

Global design
Determine a rough order quickly: group training goals that will be trained together and put the groups in the correct order. Describe with a few keywords in the Notepad why you have chosen this sequence. Postpone the definition of intermediate goals to the detailed design phase.

Detailed design
Look at your training strategy (in Step 1.1). Use the sequencing principles in the guidelines and apply them to put the training goals in a well structured and consistent training sequence. Insert intermediate goals where necessary. Use the Notepad to describe possible deviations or alternative sequences. Give arguments for the sequence that you have chosen, using the sequencing principles.
If the set of training goals is very large, it is more efficient to determine a rough order first and then look at the more detailed level of single training goals."

The process-oriented guidelines are also divided in two sections. For global design a very limited set of guidelines is provided that encourages users to do what is described for the global design phase, and not more. For detailed design the list of guidelines can be much longer:

"*Guidelines for global design:*
- Determine a rough order quickly: group training goals that will be trained together and put the groups in the correct order.
- Describe with a few keywords in the Notepad why you have chosen this sequence.
- Postpone the definition of intermediate goals to the detailed design phase.

Guidelines for detailed design [N.B.: only a part of the list]
- Use the sequencing principles in the guidelines: select and apply them one by one. This will help you to put the training goals in a consistent and well-structured order.

- If the set of training goals is very large, it is more efficient to determine a rough order first and then look at the more detailed level of single training goals.
- If the order in which training goals are covered should be adaptable during training, e.g., to the training needs or preferences of trainees, then you should describe which deviations will be allowed in the Notepad.
- Sequence your training goals in a way that suits your training strategy. Go to Step 1.1 to look at it, if necessary.
- Sequencing training goals may give you reason to choose for another approach to training. Go back to Step 1.1 to review your training strategy. Look at the Notepad and read your notes regarding the reasons for your initial choice, before you make any changes.
- Ask instructional design specialists for advice on sequencing and sequencing principles.
- Ask domain experts for clarification, if the descriptions of the training goals are not clear or not detailed enough.

.... etc."

The MASTER-TPD method and tool, including the support for managing iteration have been used for two evaluation studies. These will be described in the next chapters. I will come back to the concept of iteration during the design process in Section 7.2 and Section 10.2.

Chapter 6
The first evaluation study

6

In Chapter 6, I describe the design and most important results of the first evaluation study, an empirical study where subjects worked on their designs at home at their own pace over a three month period. The research questions addressed in this chapter are:

8. How is iteration used in the design process?
9. Can novice designers design simulator training programmes with the MASTER-TPD method in an adequate way?

These research questions are also addressed in Chapter 7 and Chapter 9.

6.1 Introduction

In Chapter 4, I described the MASTER method for the specification of training simulators. In this chapter and in the next chapter I will describe the design and most important results of two evaluation studies in which subjects used the second phase of this method: Training Programme Design (TPD). The goals of these studies are to:
1. Evaluate the usability of the MASTER method for the specification of training simulators.
2. Evaluate the effects of different forms support in the prototype tool.
3. Study how instructional designers iterate during the design process.

The MASTER-TPD method and prototype tool that are used as research instruments in the evaluation studies have been described Chapter 5. Section 6.2 gives a short description of the domain that has been used as case for the studies. In Section 6.3, I describe the design of the first evaluation study and in Section 6.4, I present the most important results. Finally, the consequences for the second evaluation study are discussed in Section 6.5. Only the data that are relevant for the research questions in this thesis (see § 6.3.1) are presented here. For a complete description see Verstegen, Barnard, and Pilot (2003).

6.2 Case used for evaluation studies

For the evaluation studies, a sample domain was selected based on the following demands:
- A realistic case concerning realistic tasks, and realistic in size; preferably an existing case from a real design project.
- A case that can be solved within a realistic time-span for an evaluation study.
- A case that is complex enough to force subjects to look for additional information and to iterate during the design of a training programme.
- A domain that the subjects have little knowledge about (to prevent that some subjects would have an advantage due to more extensive insight into the domain), but that is not too difficult (i.e. not too much domain knowledge required).
- A domain for which enough knowledge is available to create an authentic setting.

The selected case is taken from another TNO project: the training of analysts of thermal images that are collected with Unmanned Aerial Vehicles (UAVs). UAVs are unmanned versions of aeroplanes that can be used to collect information in situations where using normal aeroplanes would be dangerous or inefficient (van den Bosch, Barnard, and Helsdingen, 1999). Military applications of UAVs could be the collection of information concerning enemy terrain or hostile troops. Civil applications include, for example, surveillance in rough terrain, gathering information about oil containers or oil spills, and collecting information in areas that have been contaminated with radio-active radiation.

A UAV makes images with cameras that are attached underneath and sends them to a base station. At the base station the UAV can be navigated, the cameras can be turned around and tuned, and collected images can be shown on screen and stored. This particular UAV has a thermal image camera and two operators at the base station: a navigator and a thermal image analyst. Analysts need to be specially trained, because thermal images are very different from normal daylight images. In thermal images the temperature (i.e. the heat radiation) of objects is shown in different colours or grey scales. Additional complications include the limited update rate, the variable quality of the images because of atmospheric disturbances, and the birds' eye perspective of the UAV images (instead of the horizontal perspective on the ground).

Besides the interpretation of the images, the thermal image analysts also have to operate the controls to turn around the camera to find interesting spots and hold them in sight, and they have to tune the camera, e.g., zooming in or out and adjusting the contrast and the luminance of the images. Controlling and tuning the camera has to be done quickly and accurately, because the UAV itself is moving. Finally, the thermal image analysts have to communicate with the navigator that controls the UAV and they have to report important information to their superiors, following the correct procedures.

As explained in Section 5.1 only the TPD phase of the MASTER method has been used for the evaluation studies. This phase requires a set of training goals and a description of the target group(s) of trainees as input (although these do not have to be final or complete, because training simulator specification is assumed to be an iterative process). In the MASTER method training goals and target group descriptions these the outcomes of the first phase, the Training Needs Analysis. For this case a task analysis had already been executed in another TNO project, and a list of training goals was available (van den Bosch et al., 1999). For my evaluation studies the training goals have been redefined in more general terms, i.e. on a more abstract level and not related to any specific UAV, to avoid mentioning military or UAV-specific details and to make the domain understandable for the subjects in the evaluation study who had no previous knowledge about the domain. The full list of 54 training goals is given in Appendix C. A few examples of are (translated from Dutch):

- Can focus the camera.
- Knows the important differences between moving and still objects.
- Can identify 80% of the elements on moving thermal images in the following categories: buildings, vehicles, roads, terrains, living creatures, and industrial installations.

The target group was described as follows.

> "Trainees are personnel from military forces, police or civil surveillance organisations. They have a few years of experience in operational jobs and are now trained for the job of thermal image analyst. Trainees are 25 to 40 years old, and have a technical background at medium vocational level."

To allow subjects to practice with the MASTER-TPD method and tool, another small set of ten training goals about car driving was implemented. The data regarding this practice domain have not been used for analysis.

6.3 Design of the first evaluation study

6.3.1 Research questions

Based on the goals described in Section 6.1 the following three research questions (corresponding to research questions 8-10 in § 1.5) with sub-questions have been formulated:

1. Can novice designers design simulator training programmes with the MASTER-TPD method in an adequate way? More specifically in this context:
   a. Can subjects design a suitable training programme with the MASTER-TPD method?
   b. Are subjects able to complete the steps of the MASTER-TPD method? Which steps are difficult?
   c. What is the subjects' opinion about the structured step-by-step approach of the MASTER-TPD method?
2. Do subjects use the different forms of support that are offered?
   a. How much do subjects use the different types of help offered by the tool? Is the quality of their training programme designs related to the use of help forms?
   b. Do subjects make notes? Is the quantity of notes related to the use of (different kinds of) help?
   c. Do subjects approach experts for support? If so, which kind of questions do they ask?
   d. Do subjects use the web-based discussion groups to get support from each other?
   e. What is the subjects' opinion about the different forms of support?

3. How is iteration used in the design process?
   a. How frequently do subjects iterate?
   b. Does more iteration lead to a better training programme design?
   c. Is the number of iterations related to the use of help?

In this chapter the results regarding research questions 1a, 2d, 3a and 3b will be discussed. The discussion of the other research questions is postponed until Chapter 8, where the results of this study and the second study regarding will be presented simultaneously and compared.

6.3.2 Setting

To answer the research questions formulated in Section 6.3.1 the evaluation study needed to have a realistic time-span, representative users and an authentic setting.

6.3.2.1 Time-span

The most important demand for the evaluation studies was a time-span that is realistic for the design of simulator training programmes. To evaluate the usefulness of the MASTER method, it needed to be tried with a case that is realistic in size and complexity. Going through the steps of the method with such a case takes time, especially when the design process is supposed to be iterative. In reality, the design of specifications for a training simulator often takes years (see § 3.2.1). That would be too much for the purpose of this research, even in an empirical setting. Therefore, the evaluation study focuses on an essential part of the MASTER method: the TPD phase (see § 5.1) with a realistic but not very large case (see § 6.2). The estimate was that one to two months (working full-time on the assignment) should be enough to complete the steps of the MASTER-TPD method, allowing ample time for iterations.

The aim was to focus on the use of the method, and not on the prototype tool (that was meant to be a research instrument only, see § 5.2). This was another reason to choose for a longer time-span: to prevent that subjects would be hampered in their design process by technical or interface problems, they needed time to get acquainted with the prototype tool and the available forms of support.

6.3.2.2 Representative users

The target users of the MASTER method are assumed to have little or no experience in the specification of training simulators: since the acquisition of training simulators is not a frequent event, most people do this only once or twice in their career. They usually have knowledge and experience in related fields, however: they are domain experts, experienced instructors, designers of class-room teaching materials, etc. (see § 3.2.1). At the moment structured design methods such as the MASTER method are not used (see § 2.3.1.3). Therefore, from a training point of view, all target users -even those with some experience- can be regarded as beginners with regard to the specification of training simulators with a systematic ISD-based method.

The target users are professionals that cannot participate in a long evaluation study. The subjects in this study are third year students in Educational Sciences. Like the target users, they have no experience in the specification of training simulators or the use of the MASTER method. They probably have more theoretical background knowledge about instructional design and development models (see § 3.3) than most of the target users, but not much experience in applying this knowledge. On the other hand, they have less domain knowledge and knowledge about training simulators. See Chapter 9 for further discussion of these differences, and the results of a short evaluation study with military target users.

6.3.2.3 Authentic setting

The evaluation study has been executed as a part of a module of the Educational Sciences curriculum of Utrecht University. The setting was as realistic as possible: the subjects worked with a realistic domain and a realistic time schedule (three months, working half time on this course). They worked individually, but they could discuss with each other during meetings and in web-based discussion groups. Furthermore, different kinds of experts could be approached for advice during meetings or through e-mail (see § 6.3.6.1). This setting is one of the situations that can also occur in practice where, for example, an instructor or SME could be responsible for this part of the design process with the input from a task analysis performed by himself, or by somebody else (see § 3.3.7.1). This designer is likely to be in contact with (domain) experts and peers that can answer questions or give comments.

In this study the subjects worked half-time on this design task and at their own pace. This is comparable to the real situation, where the designer is likely to have other tasks and responsibilities as well. The main difference with reality is that this evaluation took place in an educational setting: the subjects worked for study credit points, and were assessed by their lecturers at the end of the course. Furthermore, some guidance was provided (see § 6.3.6). During the evaluation study data about the design process were registered and subjects were asked to fill in questionnaires (see § 6.3.4).

6.3.3 Materials

The domain used for the evaluation study is the training of analysts of thermal images that are collected with Unmanned Aerial Vehicles (UAVs). See for a description Section 6.2. Subjects received an electronic version of the introduction about the MASTER-TPD method and tool provided during the first meeting (text with screen prints), a MASTER-TPD glossary (see Appendix B), and a reader with background material on thermal images, unmanned vehicles, ISD, the MASTER method in general and the TPD phase in particular.

For the evaluation study the MASTER-TPD research tool described in Chapter 5 was used. The prototype tool guides users through the MASTER-TPD phase in six main steps, each divided in sub-steps (see § 5.3), and offers additional support to deal with iteration (see § 5.4). The tool provides Notepads with each step and logs the users' actions (see § 5.2). The MASTER-TPD tool is a customised version of Designer's Edge (DE 2.0, 1997). It runs under Windows95 and Windows98. During the first week of the course the tool was installed on the subjects' own computer. Subjects who did not have a suitable computer borrowed one from Utrecht University.

During the first evaluation study two versions of the MASTER-TPD tool with slightly different interfaces (but providing the same information, see § 5.2.1) have been used. I will not further discuss this issue in this thesis, however, since no important differences regarding the research questions were found between subjects using the two versions. Furthermore, the total number of participants is rather small and due to uncontrollable dropout (see § 6.3.5) the subjects were not equally divided over the two groups (seven subjects vs. three subjects). Just to make sure, I did check whether there were any prominent differences between the scores of the two groups on all the relevant data, by inspecting the results and using a Mann-Whitney test for ordinal data (i.e. ranking scores) and a T-test for independent samples for interval data. I found differences in a few items of the questionnaires that specifically concern the way help forms are provided. These results are not relevant for the research questions formulated in Section 6.3.1 and are, therefore, not reported in this thesis. There also was a significant difference in the results for one question concerning the structured approach, see the footnote with Table 8.6 (See Verstegen et al., 2003 for a detailed description).

6.3.4 Data gathering methods

The following data were collected by the MASTER-TPD tool:
- log files with subjects' actions with the MASTER-TPD tool with time stamps (see § 5.2.1),
- the subjects' training programme designs, and
- the subjects' notes on the Notepads.

The tool provides a function that sends a compressed copy of the database to an ftp-site at Utrecht University. Subjects were asked to send in their intermediate results every week. Not all of them did, but for all subjects versions of at least three moments are available:
- at the end of the global design phase (at the first feedback moment, see § 6.3.6.2),
- at the end of the detailed design phase (at the second feedback moment, see § 6.3.6.2), and
- at the end of the course (final versions).

During the evaluation study four questionnaires were issued (see Table 6.2):
- questionnaire 1 at the first meeting, concerning the subjects' background,
- questionnaire 2 at the fourth meeting, concerning the subjects' first experiences with and opinions about the MASTER-TPD method and tool,
- questionnaire 3 at the sixth meeting, concerning the subjects' opinions about the MASTER-TPD method, its usability and usefulness, and
- questionnaire 4 at the last meeting, concerning the subjects' experiences with the prototype MASTER-TPD tool and the different forms of support.

All four questionnaires can be found in the Appendix of Verstegen et al. (2003).

Other data collected during the first evaluation study are:
- the questions put to experts (through e-mail),
- the feedback provided to subjects (see § 6.3.6.3),
- the discussions between subjects in the web-based discussion groups,
- video-tapes of discussions during face-to-face meetings, and
- the presentations that subjects gave about their training programme designs at the last meeting (see § 6.3.6.5).

In the log files the times are measured in seconds (s). For all the calculation the original measures in seconds were used. For readability purposes these measures of time have been translated into hours and minutes in the tables (rounding seconds to the nearest minute). The results of the study will be presented in tables and figures. I have computed correlations only as a way to find possibly interesting results, since the validity of correlation coefficients is not guaranteed with such a limited number of subjects. The results of these computations are reported in footnotes. The Spearman Rho Correlation Coefficient (SRCC) has been used for data on an ordinal scale (i.e. rankings) and the Pearson Product-Moment Correlation Coefficient (PPMCC) for data on a interval scale.

The video-tapes did not provide additional information that was not available from other sources. Therefore, they have not been formally analysed. The web-based discussion groups did not run well. After a few weeks the lecturers dropped the demand for two contributions per week. The discussion groups remained in use until the end of the course. I will only report the subjects' opinions about their usefulness in this thesis. The analysis of the log files of these web-based discussions is not discussed, since it did not provide information regarding the research questions that is not also provided by other sources (e.g., the questions to experts). See Verstegen et al. (2003) for a detailed description.

6.3.5 Subjects

The subjects were students that followed the course 'COO99' in the spring of 1999. The course was an optional module in the third year of the Educational Sciences curriculum of

Utrecht University. However, students majoring in other subjects and postgraduates could also participate. It was possible to follow the course in daytime as well as in the evening.

At the beginning of the course 17 students had subscribed. After one week two additional students were allowed to participate by the lecturers. However, within two weeks six of the 19 students dropped out again, and three others stopped later on (before the second feedback moment, see § 6.3.6.3). Four of these nine dropouts were students majoring in other subjects with no background in Educational Sciences, three were students of Educational Sciences and two were postgraduate evening students who followed this course as a separate module alongside their job. The nine dropouts gave the following reasons:
- Main reason: lack of time (8 subjects).
- Main reason: expected a course in implementing CBT (1 subject).
- Secondary reason: perceived lack of background in Educational Sciences (2 subjects, both postgraduate evening students).

According to the university lecturers a 50% drop out during courses is not unusual.

The data presented in this chapter concern the results of the remaining ten subjects who finished the module and presented their designs during the final meeting. The results of the first questionnaire issued before the start of the evaluation study show that the subjects had no experience with training simulators or tools for instructional design (see Table 6.1). There was no indication to expect problems with using computers, e-mail or the web-based discussion groups. And in fact, during the evaluation study, there were no signs of such problems.

Table 6.1: Background of subjects

Background	yes	no	unknown[a]
Students of Educational Sciences (third year or more)	8	2	-
Full time (i.e.) day students	9	1	-
Followed course(s) in instructional design	8	-	2
Experience with tools for instructional design	-	8	2
Followed course(s) in interface design	3	5	2
Followed course(s) in selection/design of instructional products	8	-	2
Experience with web-based discussion groups	8	-	2
Experience with training simulators	-	8	2
Experience with computer programming	10	-	-
Previously completed education for primary/secondary school teacher	5	5	-
Teaching experience	6	2	2

[a] The two subjects that started in week 2 did not fill in the first questionnaire.

Eight of the ten subjects were third year (or more) students of Educational Sciences. They had followed one or more courses in instructional design theory, and the selection and/or design of instructional products. Some had also followed courses in the field of interface design. Most of them had some experience in teaching, ranging from a few weeks to many years. The two subjects that joined later did not fill in the first questionnaire because they were not present during the first meeting. They were studying Computer Linguistics and Multimedia-psychology respectively. Both of them had no background in Educational Sciences or teaching.

6.3.6 Procedure

The evaluation study has been executed as a module of the Educational Sciences curriculum of Utrecht University. This module took three months (April, May and June 1999). The subjects were supposed to dedicate half of their time to this module, i.e. 20 hours a week. Table 6.2 summarises the procedure for the evaluation study:

Table 6.2: Procedure of the first study

Activity	Including	Week
First meeting (obligatory)	Explanation goals and planning Brief introduction MASTER method in general Introduction MASTER-TPD method and tool Introduction about simulator training First questionnaire	1
Installation tool	At subjects' own (or borrowed) computer	1
Practice	Practise with small 'car driving' domain (see § 6.2)	1
Second meeting (obligatory)	Presentation domain expert (thermal images/UAVs) Demonstration of two TNO simulators Opportunity to ask questions about TPD Explanation about tasks week 2-12	2
Global design phase	Go through all MASTER-TPD steps quickly to design preliminary version of training programme	2-3
First round of feedback	Feedback on global design	3-5[a]
Third meeting (not obligatory)	Opportunity to ask questions	3
Detailed design phase	Go through MASTER-TPD method again to elaborate design in detail	4-9
Fourth meeting (obligatory)	Demonstration thermal image camera Demonstration simulated UAV (navigation/camera) Opportunity to ask questions Second questionnaire	5
Fifth meeting (not obligatory)	Excursion to simulator training site of the Royal Netherlands Army (RNLA): simulator for short range air defence missiles	7
Sixth meeting (obligatory)	Demonstration of Designer's Edge 3.0 by Dutch retailers Third questionnaire	9
Second round of feedback	Feedback on detailed design	9-11[a]
Finalising stage	Finalising design based on second feedback round Prepare presentations	10-12
Seventh meeting (obligatory)	15 minute presentations by all subjects Fourth questionnaire	12

[a] Feedback was provided when subjects had sent in their global or detailed design (see §6.3.6.3).

6.3.6.1 The subjects' task

The subjects' task was to design a training programme with the MASTER-TPD method and tool for the purpose of designing specifications for a training simulator to be used for training the analysts of thermal images from a UAV (see § 6.2). Their input was a list of 54 training goals and a description of the target trainees (see Appendix C). Subjects were allowed to work at their own pace, but to give them some indication of the amount of work to be done, the work was divided into more or less equal portions described in the weekly assignments (see Appendix D in Verstegen et al., 2003). The subjects worked at home with the possibility to discuss with each other in web-based discussion groups and to ask advice from experts through e-mail. The following experts were available:

- Simulator training specialist: For questions about the MASTER-TPD method and other questions about the design of simulators.
- Tool specialist: For questions about the tool.
- Domain specialist: For questions about the domain, i.e. interpretation of UAV thermal images.
- Instructional design specialist: For questions regarding instructional design and development theory.
- Director: For questions regarding the university course (organisation, requirements, assessments, etc.).
- Assistant manager: For questions regarding the web-based discussion groups.
- Project manager: For all other questions.

Furthermore, in the first week subjects received an electronic version of the introduction about the MASTER-TPD method and tool provided during the first meeting, and a reader with background material (see § 6.3.3). The MASTER-TPD glossary (see Appendix B) was made available through the same web site as the discussion groups. Due to technical difficulties it was only available from the second week.

6.3.6.2 Global and detailed design phases

During the first week subjects had time to practice with the MASTER-TPD method and tool using a small practice domain regarding car driving. For the remaining 11 weeks subjects worked on the main domain (see § 6.2). The second and third week were dedicated to the global design phase: subjects were asked to go through all the steps quickly in order to design a first preliminary version of the training programme. In the detailed design phase subjects had six weeks to go through the steps of the MASTER-TPD method again and elaborate their design further. The remaining three weeks subjects finalised their design based on the second round of feedback and prepared for the presentation of their design during the last meeting at the end of the twelfth week.

6.3.6.3 Feedback

Beforehand, feedback on intermediate designs had not been planned. The lack of feedback, however, appeared to make subjects too insecure (see § 6.4.1.4). Therefore two feedback moments were introduced:

- First feedback moment: provided by e-mail as soon as possible after subjects had sent in their global training programme design, i.e. to seven subjects at the end of the third week (as planned) and to the other 3 subjects seven to ten days later (because they were late sending in their work).
- Second feedback moment: feedback was provided by e-mail as soon as possible after subjects had sent in their detailed training programme design, i.e. to seven subjects at the end of the ninth week (as planned), to one subject about four days later and to the remaining two subjects only in week 11.

One of the university lecturers and the author of this thesis formulated feedback collectively. Since the aim was to study the design process and the variety in designs, the feedback did not interfere with the subjects' design styles, choices or reasoning. The feedback focused on the way the MASTER-TPD method was executed, not on the quality of the design or specific design decisions. This was in accordance with the way subjects were assessed by their lecturers at the end of the course (see § 6.3.6.6).

6.3.6.4 Intervention in week 7 not executed

To enforce iteration within the detailed design phase, an intervention had been planned at the end of week seven. Subjects were going to be told -depending on the intermediate version of their design- that this design was too expensive, or that they had budget left for extra facilities. The aim of the planned intervention was to stimulate subjects to iterate and review their design. However, the two feedback rounds, that had not been previously planned, fulfilled the same role (see Table 8.10 for the content of the feedback). Therefore, this intervention has not been executed .

6.3.6.5 Meetings

There were seven face-to-face meetings, five of which were obligatory (see Table 6.2). During the first four meetings subjects received information about the MASTER-TPD method, the prototype tool and the domain. The subjects also had the opportunity to ask questions and/or discuss with the presenters. The fifth meeting was a field visit to a simulator, and during the sixth meeting the newest version of DE was demonstrated. During the last meeting subjects presented their designs. In the instructions, sent by e-mail beforehand, subjects were asked to explain shortly but clearly their training programme design in 15 minutes (including questions). The instructions stressed the need to motivate the most important design decisions, e.g., what is covered with simulator training and what is not.

6.3.6.6 Assessment for grading purposes

Subjects could earn seven study credit points for the course (N.B.: one credit point is equivalent to working one week full-time). Assessment was done by two university lecturers. The lecturers did not assess the quality of design decisions taken by subjects, but rather their ability to present their design (during the last meeting) and to explain and defend the design decisions they had taken (in a written report or an oral exam). Further requirements to pass the course were: a sufficiently elaborated training programme design at the end of the course, sending intermediate versions of the design weekly, and attending the obligatory meetings.

6.4 Results of the first evaluation study

In this section the results regarding the following research questions will be discussed:
§ 6.4.1. Can subjects design a suitable training programme with the MASTER-TPD method?
§ 6.4.2 Do subjects use the web-based discussion groups to get support from each other?
§ 6.4.3 How is iteration used in the design process? How frequently do subjects iterate? Does more iteration lead to a better training programme design?
The discussion of the other research questions is postponed until Chapter 8 (see § 6.3.1).

6.4.1 Usability of the MASTER-TPD method

6.4.1.1 Quality of the training programme designs

Since the subjects' products concern only a part of the design of simulator specifications, there is no objective measure of quality. It is not possible to try out the specified simulator with trainees to evaluate whether it fulfils the training needs. Moreover, the draft training programme design that is the result of the MASTER-TPD method is still far away from detailed (technical) specifications or actual training. Only subjective measures of the quality of the subjects' results are possible. For the purpose of this research, two experts judged the quality of the subjects' training programme designs. The two raters are TNO staff members with knowledge about and experience in specifying simulator training (among whom the author of this thesis). They spent two to three days on this task and used prints of the subjects' training programme designs, including an overview of the chosen training strategy, a list of training goals in sequence, the choice of instructional products and training activities for each training goal, an overview of the training activities that were assigned to simulator training, the subjects' descriptions of scenarios, assessments and instructional interventions, and all the notes that subjects made on the Notepads. The prints were anonymous, i.e. the subjects' names were erased.

In the light of the research questions formulated in Section 6.3.1 the focus was on the use of the MASTER-TPD method, and the overall quality of the draft training programme designs that subjects made. Basically, the question was whether these designs would be suitable as input for the definition of training simulator specifications. The two raters judged the overall quality of the subjects' training programme designs with the instruction to focus on the following aspects:
- Did subjects execute the steps of the MASTER-TPD method adequately?
- Are all steps sufficiently elaborated?

- Is the design coherent, e.g., do subjects apply the training strategy that they have chosen in the design of training activities and training scenarios?
- Does the design provide the information necessary for the specification of the future simulator?
- Do subjects adequately explain and defend their decisions in their notes?

The raters did not score these issues separately, because they are not objective, measurable criteria. It is not clear whether the aspects are independent of each other or whether they are equally important. For example: how can one decide whether a step has been adequately elaborated when subjects do not make good notes? Scoring aspects separately and then computing an average, suggests a more objective way of measuring quality, but is in the end equally subjective. Some important aspects may be forgotten, while others get too much influence. The risk is that raters focus on (measurable) details, and not on the overall quality of the designs. For example, a subject that is good at expressing decisions in words may get a high score even when he or she does not take the domain into account or does not pay attention to what information is necessary for simulator specification at all. Therefore, the two raters just ranked the subjects' training programme designs from the best (10) to the least good (1). The average scores were ranked again to come to an 'Average quality ranking' to use for comparison with other results below. The two subjects with an equal average (Subjects 8 and 9) have been given the same intermediate ranking.

Striking in Table 6.3 is that the rankings of the two TNO experts are very similar[6]:

Table 6.3: Quality of the designs: ranking by TNO raters[a]

	Ranking TNO Rater 1	Ranking TNO Rater 2	Average	Average quality ranking
Subject 1	5	4	4.5	5
Subject 2	10	10	10	10
Subject 3	2	2	2	2
Subject 4	3	5	4	4
Subject 5	4	3	3.5	3
Subject 6	8	8	8	8
Subject 7	1	1	1	1
Subject 8	6	7	6.5	6.5
Subject 9	7	6	6.5	6.5
Subject 10	9	9	9	9

[a] Rankings from best (10) to least good (1).

There are two other measures of the quality of the designs (see Table 6.4):

1. The grades given by the two university lecturers at the end of the course[7]. The lecturers based their grades not only on the training programme designs, but also on the subjects' presentations at the final meeting and their reports or oral exams.
2. Ranking scores from two instructors, also Subject Matter Experts (SMEs) in the domain, of the Army school who will be responsible for training thermal image analysts for UAVs[8]. During a meeting for another project later in the year (in fall 1999, see also § 9.2) these two instructors/SMEs spent about an hour quickly ranking the designs from the best (10) to the least good design (1). They spent less time on rating the designs than the other raters and were expected to look at the designs from a different, more domain-oriented and pragmatic viewpoint.

6 SRCC Ranking TNO Rater 1/Ranking TNO Rater 2 = 0.95.

7 SRCC Ranking based on grades/Average quality ranking = 0.93.

8 SRCC Ranking instructor-SME 1/Ranking instructor-SME 2 = 0.79; SRCC instructor SME1/Average quality ranking = 0.70; SRCC instructor-SME 2/Average quality ranking = 0.70.

Table 6.4: Other quality measures[a]

	Grades by university lecturers	Ranking based on grades	Ranking instructor-SME 1	Ranking instructor-SME 2	Average quality ranking (from Table 6.3)
Subject 1	6.7	3.5	9	7	5
Subject 2	8.7	9	8	10	10
Subject 3	4.7	2	4	3	2
Subject 4	6.7	3.5	2	1	4
Subject 5	7	5.5	5	5	3
Subject 6	8.7	9	6	9	8
Subject 7	2.3	1	1	4	1
Subject 8	8.3	7	3	2	6.5
Subject 9	7	5.5	7	6	6.5
Subject 10	8.7	9	10	8	9

[a] Rankings from best (10) to least good (1).

The quality rankings in Table 6.4, although based on slightly different criteria, are remarkably similar to those in Table 6.3. The two subjects lowest on the final ranking by TNO staff members also received grades less than 6 from the university lecturers and thus failed the course. They were given the chance to work further on their design and report, and resubmit them after the summer holidays, but they never did so. One of these subjects was not majoring in Educational Sciences (Subject 7). Whether this was an important factor is not clear. The other subject that failed did have a background in instructional design (third year Educational Sciences, primary school teacher and teaching experience). Moreover, the second subject that was not majoring in Educational Sciences (Subject 10) scored among the best. It is far more likely that the time spent on the design task was the most important factor (see below). In fact, it is clear from Table 6.5 that Subject 7 did not take the design task seriously: he spent only 12.5 hours on it.

6.4.1.2 Time spent on the design task

From the log files the amount of time that subjects were logged into the tool has been derived. Table 6.5 gives an overview:

Table 6.5: Time spent on design

	Time spent on design (total) [a]	Ranking based on total time	Number of sessions	Time spent on global design[a]	Percentage spent on global design	Average quality ranking (from Table 6.3)
Subject 1	39:38	2	17	13:56	35 %	5
Subject 2	121:26	10	47	40:42	34 %	10
Subject 3	54:06	5	20	13:53	26 %	2
Subject 4	50:12	4	26	20:35	41 %	4
Subject 5	55:12	6	39	9:21	17 %	3
Subject 6	99:53	9	30	11:04	11 %	8
Subject 7	12:31	1	22	5:25	43 %	1
Subject 8	81:48	8	31	37:04	45 %	6.5
Subject 9	44:45	3	37	5:25[b]	12 %	6.5
Subject 10	62:50	7	31	14:30	23 %	9

[a] Time in hours and minutes.
[b] Subject 9 had technical difficulties in the first weeks; this may have been a reason to spend less time in the global design phase.

The log times are used as a measure for the time spent on design task, although it is not certain that subjects were always working on their training programme design when they were logged in. All the log entries made before the subjects started to work on their main design task have been removed, i.e. entries from installation and testing and from the practice period in the first week. In a number of places the log files had to be repaired: when the tool had crashed or was not shut down properly, an extra entry was inserted to mark the end of the session. For this entry the time stamp of the last action of that session was used. Sessions where the users logged into the tool but performed no other action, i.e. did not enter any specific step and did not make calls to any of the help forms, were marked as 'void' and removed from the log files.

There are large differences in the time that subjects spent on the design task: roughly between 12.5 and 121.5 hours (average 62 hours and 14 minutes; standard deviation: 31 hours and 22 minutes). The number of sessions with the tool ranged from 17 to 47 (average: 30.0 sessions; standard deviation: 9.3 sessions). Subjects spent between 11% and 45% of the total time they spent on the design task on their global design (week 2-3)[9]. The amount of time spent on the design task seems to have had influence on the quality of the resulting training programme designs, as shown in Figure 6.1[10]:

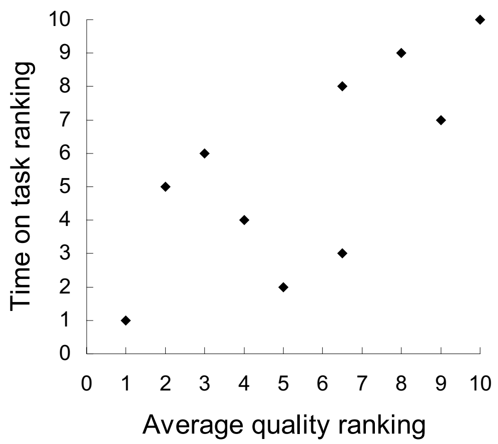


Figure 6.1: The time spent on the design task (ranking from Table 6.5) versus the quality of the resulting training programme designs (average ranking from Table 6.3).

6.4.1.3 Subjects' opinion about the usability of the MASTER-TPD method

The data presented in Table 6.6 are results from the third questionnaire issued at the sixth meeting (in week 9):

Table 6.6: Opinions about the usability of the MASTER-TPD method[a]

Question	Mean score	Standard deviation
You have worked with the MASTER-TPD method for 9 weeks. Did you find that:		
# Difficult ----------------- Easy	3.4	1.0
# Not useful --------------- Useful	3.4	1.0
# Boring ------------------- Exciting	3.0	0.8
The MASTER method is meant to support the design and acquisition of training simulators. Did you find this support:		
# Insufficient -------------- Sufficient	3.7	0.8
# Not useful --------------- Useful	4.0	0.7
If you had to specify a training simulator, would you use the MASTER method?		
# Definitely not ---------- Yes, certainly	4.1	0.7
Would you advice others (e.g., business relations) who have to specify a training simulator to use the MASTER method?		
# Definitely not ---------- Yes, certainly	3.7	0.8

[a] Scores on a 5-point scale from 1 (left side) to 5 (right side).

Subjects were positive about working with the MASTER-TPD method: it was easy, a useful experience and not boring. In their opinion the method sufficiently supports the design and acquisition of training simulators. If they would have to specify a training simulator, they

[9] PPMCC Time spent on design (total)/ Time spent on global design = 0.71.
[10] SRCC Average quality ranking/Ranking time spent on design task = 0.72.

would use it, and they would advise others to use it as well. Some subjects remarked, however, that they also did not know any other methods, so they would not know if there was anything better.

6.4.1.4 Conclusion

The results of the evaluation study show that the subjects can design a suitable training programme with the MASTER-TPD method: eight of the ten subjects that completed their training programme design passed the course. The two instructors/SMEs that looked at the designs (see Table 6.4) commented that the quality of the designs varied a lot, but that that the best ones would definitely be suitable to be used as input for training simulator specifications. The subjects that did not complete the course did not stop because they had difficulties with the method (see § 6.3.5).

The responses to the questionnaires show that the subjects liked working with the MASTER-TPD method and did not find it very difficult. The fact that the planned procedure could be executed without major changes confirms that subjects were able to use the MASTER-TPD method and prototype tool. Striking, however, was the subjects' constant demand for feedback. Some subjects seemed to have expected feedback every week, and expressed that not knowing whether they were on the right track made them very insecure. For example (translated from Dutch):

- "I also think that getting feedback on what you made is very important."
- "I would just like to know whether what I do is correct. Could you help me with that? Then I do not have to be so insecure and struggling so much during the rest of the course..."

There are several possible explanations for this phenomenon:

1. Subjects had to work with an unknown domain. The subjects had no experience with simulators, and no knowledge about the specialised domain. However, background information was given during meetings and in the reader. Moreover, the experts that were available for advice through e-mail did not receive many questions about the domain (see Table 8.3-A).
2. Designing a simulator training programme is a difficult task. However, the results of the third questionnaire show that subjects did not think this task was too difficult for them (see Table 6.6).
3. The results expected of a design task are not well-defined, i.e. it is not very clear beforehand what the results should look like. There is no 'solution', only a defined design process that can lead to a good, well-motivated design. Moreover, there may be different, but equally good designs (see also § 3.2.2.2). Some subjects found this hard to deal with, as is illustrated by the following two quotes from e-mail (translated from Dutch):
   - "I sometimes have the feeling that I'm just fooling around and I have absolutely no idea whether it is good or bad. I miss feedback..."
   - "...The reason that I get back to this [i.e. question for feedback, red.]: I have discussed the work a little with a fellow student. Our work was very different..."
4. The subjects' attitude. The subjects are students who may have had more interest in passing the course than in designing a good simulator training programme. Asking for immediate feedback and suggestions is easier than elaborating a design and spend time on discovering yourself how it can be improved.
5. The consequences of failure. More than in many other academic courses the subjects were responsible for their own work, and failure meant the loss of a substantial amount of study credit points. Moreover, the university lecturers were not entirely clear on how assessment was going to take place exactly until later in the course.

Results of the second study indicate that the last point may have played a large role, see the discussion in § 7.4.1.5.

The results of this evaluation study suggest that designing simulator training programmes is, for an important part, 'just work': with the help of the MASTER-TPD method and tool, all subjects who spent enough time at the task were able to finish it. An important factor influencing the quality of the results seems to be time on task: in general, more time spent on the design task leads to a better training programme design. There is no evidence of the influence of other factors, like teaching experience or background. In Chapter 9, I will discuss whether these results are expected to be representative for the prospective users of the MASTER method.

6.4.2 The web-based discussion groups

In this section I briefly discuss the experiences with the web-based discussion groups because they were not used during the second evaluation study. The discussion of other forms of support that were available in both studies is postponed to Chapter 8.

The two discussion groups were meant to exchange experiences and discuss problems amongst each other. The discussion groups did not work very well, partly caused by the small number of participants, especially in the smaller group. The discussions were also hindered by the fact that subjects worked at their own pace. Some were lagging behind, because of illness or other reasons. To fulfil the demand for two contributions a week, they sometimes reacted weeks later when their reactions were of no interest anymore to the rest of the group (N.B.: the demand for two contributions a week was dropped later on).

Another reason for the lack of success of the web-based discussion groups could be that the subjects could not see each others' designs. Two remarks from subjects illustrate this (translated from Dutch):

- "I have experienced the discussion group as useful, especially in the beginning because we could still talk at a more global level about how you do something like this, and because you can ask for and offer each other help with technical problems. But further on in the design, you are working at such a specific level at your own scenarios that the discussion group can offer little support..."
- "...You never see the whole design of the other person, which makes it difficult to give or get constructive feedback...".

Web-based discussion groups may be useful when they are organised in such a way that subjects can take a look at the designs of their peers at the same time and/or when discussions can be planned at moments that all subjects are working on the same issue or step (i.e. synchronous discussions). However, in the format that they were used in this study they are not useful. See Section 7.4.3.3 for further discussions about synchronous and asynchronous web-based discussions.

6.4.3 The use of iteration

6.4.3.1 The number of iterations

The log files show which steps of the method the subjects worked on and in which order. From this information, the number of iterations has been derived, i.e. how often subjects have gone back to a previous step. To derive the number of iterations, some minor repairs had to be done: sometimes the entry marking the end of a certain step was lacking or the steps were deeply nested (N.B.: this happens when users keep windows open that they are not currently working on). Note that the number of iterations in Table 6.7 may be an underestimate. For some steps the log files contain time stamps of the start and the end, but for other steps the log files only show when objects are added or removed from corresponding tables in the database (i.e. when DE steps were used, see § 5.2.1). If subjects have gone back to one of these steps without making any changes, it does not show in the log files.

Table 6.7: Number of iterations

	Number of iterations	Ranking number of iterations	Time spent on design (from Table 6.5)	Average quality ranking (from Table 6.3)
Subject 1	18	6	39:38	5
Subject 2	16	5	121:26	10
Subject 3	25	8	54:06	2
Subject 4	13	4	50:12	4
Subject 5	27	9	55:12	3
Subject 6	10	3	99:53	8
Subject 7	5	1	12:31	1
Subject 8	6	2	81:48	6.5
Subject 9	19	7	44:45	6.5
Subject 10	37	10	62:50	9

Table 6.7 shows that the number of iterations ranges from 5 to 37 (average: 17.6 iterations; standard deviation: 10.0 iterations). During the presentations at the last meeting subjects explained that they had changed their opinion about several aspects during the design period, e.g., that they had changed the training strategy or the classification of what should be done with the simulator and what not. There is also evidence of this in e-mails to experts and in the log files of the web-based discussion groups. Closer inspection of the subjects' notes made clear that all subjects also describe how and why they iterated on one or more occasions.

6.4.3.2 The subjects' opinions about iteration

In an open question in the third questionnaire subjects were asked whether they thought they had often gone back and forth between steps to reconsider decisions or make changes in previous work. All subjects answered that they had iterated during their design work. They did not always like to do so. Remarks in the questions to experts and in the log files of the web-based discussion groups show that this was partly due to the problems with the tool (see below). Some subjects, however, also resented the feeling that they had to 'do the same thing again'. The following remarks illustrate this (translated from Dutch):

- "I have improved step two based on the feedback and have come to the conclusion that I now have to do everything in Step 4, 5 and 6 again. Does this bother you also?"
- "The improvement of Step 2 was a lot of work, I really have to do everything again because my work in Steps 4, 5 and 6 is no good anymore. Is it perhaps possible to look at my new Step 2 this week and to give feedback? I would rather not start working on four, five and six before I know whether this is really an improvement or it needs to be more specific"

In fact, the subject that made the first remark did not strictly follow the MASTER-TPD method afterwards. She postponed the specification of training activities (originally in Step 2.1) to Step 5, where instead of simulator scenarios she defined lessons that consisted of a number of training activities that would take place with the simulator and elsewhere (N.B.: information from her presentation during the last meeting).

On the other hand, some subjects seem to have no problems with iteration. This is illustrated by the following reaction to the complaints above (translated from Dutch):

- "[X] writes that the revisions in Step 2 have large consequences for the Steps 4, 5 and 6. But do you really have to do everything all over again? The ideas that you had (about kinds of scenarios and assessments, etc.) still exist, don't they?"

Another subject explained what she had done in Step 1.2 (Determine Training Sequence) in the global design phase, and then wrote (translated from Dutch):

- "...In the detailed design phase I plan to cluster training goals into better units (I think that the first three training goals already belong in one unit, I have not done that yet)."

In many respects the prototype tool makes it easier to iterate, compared to designing a training programme design on paper or with general software such as spreadsheets or text files.

Results and notes are stored for each step and remain available at all times, the relationships between the results of different steps are stored (e.g., which training goal a certain training activity relates to), etc. However, there seems to be room for improvement of this aspect. Seven of the ten subjects are not satisfied with the user friendliness of the tool with regards to iteration. Four subjects state that they tried to limit iteration because of these problems. They give the following reasons:

- The Course Map does not show all the training activities, but only those that will take place with the simulator. This makes it hard to keep an overview.
- Sometimes sequencing information gets lost when the Course Map is reconstructed.

The latter never occurred during testing and is suspected to be caused by subjects not using the tool as intended. The prototype tool is a customised version of DE 2.0 (1997), see Section 5.2. This software package is based on the use of hierarchies: training activities are placed under training goals; scenarios and assessments under training activities, etc. When scenarios or assessments are placed in the Course Map without connections to the hierarchy, the sequence gets lost when a new Course Map is reconstructed. If these problems occur, they can be locally repaired by dragging and dropping the elements into place. The results in Table 6.7 show that all subjects iterated despite possible problems with the tool.

6.4.3.3 Iteration and the quality of the designs

Figure 6.2 shows that more iteration does not correlate with a better design[11]:

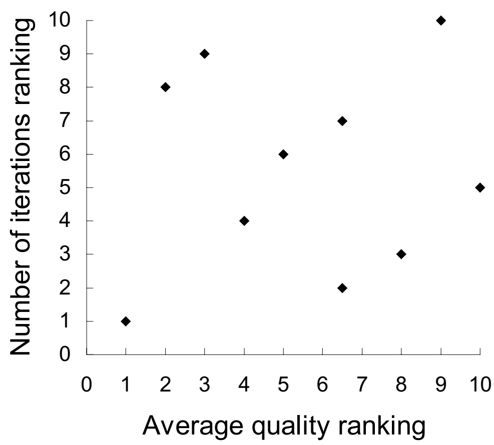


Figure 6.2: Number of iterations (ranking from Table 6.7) versus the quality of the resulting training programme designs (average ranking from Table 6.3).

In fact, between the best three designs, there are large differences in the number of iterations: Subject 10 has iterated more often than all other subjects, Subject 6 less often than average, and Subject 2 stands somewhere in between (see Table 6.7). Inspection of the log files shows that Subject 2 and Subject 6 follow the sequence of steps twice, i.e. in the global design phase and then again in the detailed design phase, only occasionally going back to earlier steps within one of these phases. Subject 10 jumps back and forth between steps far more often. Inspection of the subjects' notes shows that the number of iterations does not correspond to the amount of rethinking or reviewing done during the design period: especially Subject 2 elaborately describes in her notes for every step how she changed her mind, and/or elaborated on her work. She just seems to do so in one go, i.e. she reviews her work of the global design phase in the detailed design phase, but sticks to the sequential order of the steps in the MASTER-TPD method for both phases.

Apparently, there are various effective design styles. Some subjects work out a good concept and then go through the method step-by-step, going back to previous steps only when neces-

[11] SRCC Ranking number of iterations/Average quality ranking = 0.12.

sary. Others come to a good design by going back and forth between steps more often. Maybe they leave more decisions open, go to the next step(s) to elaborate the design partly, get new ideas, and then go back to work on the initial concept again, etc. The results show that a lot of iteration can be a part of an ineffective design style as well: the designs of the subjects that rank second and third on the number of iterations (Subject 3 and Subject 5) belong to the worst three designs. In those cases, a lot of iteration may be sign of lack of overview.

Figure 6.3 shows that there is also no obvious relationship between the number of iterations and the time spent on the design task[12]:

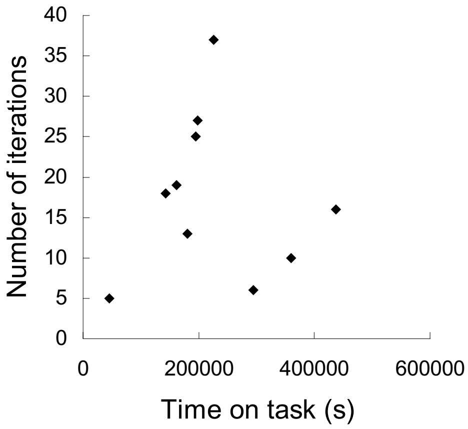


Figure 6.3: Number of iterations (from Table 6.7) versus the time (in seconds) spent on training programme design (from Table 6.5).

6.5 Conclusions and discussion

Conclusions regarding the usability of the MASTER-TPD method

The first research question addressed in this chapter was whether the subjects can design a suitable training programme with the MASTER-TPD method. The results of this evaluation study show that the MASTER-TPD method can be used by novice users with some background in education but no knowledge or experience in the field of training with simulators or designing specification of instructional products. The results of this study indicate that it is possible that some background in instructional design and development theory is required. For two subjects this was -at least partly- a reason to stop with the course. On the other hand, only one of the remaining two subjects who did not have a background in education failed. The other one passed and delivered one of the best designs (i.e. Subject 10, see Table 6.3). In Section 9.2, I will discuss the results of a short evaluation study with target users which seem to indicate that a strong instructional design background is not required when users are adequately supported.

The results of this study further suggest that in-depth knowledge about the domain is not required: the subjects did get information about the domain in the reader and during meetings, but this was definitely not sufficient to turn them into domain experts. The first reactions of domain experts to their designs (see § 6.4.1.4), however, do not indicate that they made grave errors in their interpretation of the training goals. The domain experts did not have time to study the designs in detail, so it is still possible that that would have revealed lack of domain knowledge. It is also possible that in-depth knowledge about the domain is only necessary when the training programme is further elaborated and implemented. In practice, training simulator specifications will usually be designed by a team that includes domain experts. The fact that some information may not be available, e.g., when the specifications for the operational system are still under development or when the task descriptions can still change due to

12 SRCC Ranking number of iterations/Time spent on design = -0.03.

personnel reorganisation, will be a bigger problem. If domain experts are not directly involved, it may be wise to discuss intermediate versions with domain experts as early as possible to avoid misconceptions.

In this study only one part of the MASTER method was used. The other two parts are structured in the same way, however. A systematic analysis of the training needs leading to training goals, covered by the first part of the MASTER method, may be more widely known and already more often applied (see, e.g., Swift, Martindill, and Allender, 1998; Wallace and Northham, 1998). The third part, the systematic definition of simulator specifications based on the preliminary training programme, is new. It is not clear to what extent designers would need more specific technical knowledge about simulators and simulator parts, such as vision and motion systems and simulation models. Moreover, this part of the MASTER method is currently least developed. More research will be necessary to evaluate the usability of this part of the MASTER method and to determine the background required for future users.

Conclusions regarding the web-based discussion groups
The web-based discussion groups were not useful In the format that they were used in this study (see § 6.4.2). Therefore, they have not been included in the second evaluation study.

Conclusions regarding iteration during the design process
The results of this evaluation study show that all subjects iterated, i.e. went back to previous steps to review or elaborate on work done earlier, although some subjects were clearly frustrated when they had to iterate. The number of iterations is not correlated with the quality of the results. Apparently there are different design styles -with more or less iteration- that can lead to good training programme designs and specifications. On the other hand too much iteration can also be part of an ineffective design style. To get more grip on this issue I have defined different kinds of events that are expected to evoke iteration (see § 7.2).

Limitations of this study
This first evaluation study was executed with a limited number of subjects and, because it was not a laboratory study, some factors could not be controlled, e.g., the amount of time that subjects spent on the design task. The results show that the time on task had influence on the quality of the resulting design. Furthermore, the information that the log files provide about the behaviour of the subjects is limited: it is not clear whether they were working all the time that the tool was running on their computer, nor whether they were always reading help files when they had help windows open. Finally, for some steps iteration was not measured if subjects did not make any changes there. To exclude the effect of time spent on the design task, a second evaluation study has been executed in a controlled setting where subjects worked with the MASTER-TPD method and the prototype tool for five full days under supervision. The design and main results of this second evaluation study will be described in Chapter 7.

Chapter 7
The second evaluation study

In Chapter 7, I describe the design and most important results of the second evaluation study, where subjects worked on their designs for a fixed amount of time and under supervision. The research questions addressed in this chapter are:

8. How is iteration used in the design process?
9. Can novice designers design simulator training programmes with the MASTER-TPD method in an adequate way?

These research questions are also addressed in Chapter 6 and Chapter 9.

7.1 Introduction

In the first evaluation study the amount of time spent on the design task varied a lot and had an important influence on the resulting training programme designs. Therefore, a second study has been executed in a controlled setting where subjects worked on their training programme design under supervision for a fixed amount of time.

The second important result of the first study was that, although all subjects iterated during the design of a training programme, the number of iterations did not correlate with the quality of the results. Yet it is clear that iteration during the design process cannot and should not be avoided (see § 3.2.3 and § 5.4.1). To get more grip on this issue I have defined different kinds of events that are expected to evoke iteration. In the second evaluation study several of these triggers have been introduced on purpose in order to study their effect. Apart from this, the goals of the second evaluation study are the same as for the first study, leading to the following four goals:

1. Evaluate the usability of the MASTER method for the specification of training simulators.
2. Evaluate the effects of different forms of support in the prototype tool.
3. Study how instructional designers iterate during the design process.
4. Evaluate how designers react to different triggers for iteration during the design of a training programme.

In Section 7.2 possible triggers for iteration are discussed. In Section 7.3 the differences between the design of the first and the second evaluation study are described. The most important results of the second study are given in Section 7.4 and the consequences for the list of triggers for iteration are discussed in Section 7.5. Only the data that are relevant for the research questions are presented in this thesis. See for a complete description Verstegen, Barnard, and Pilot (in press).

7.2 Triggers for iteration

The design of training simulator specifications is an inherently iterative process (see § 3.2.3). Iteration makes managing the design process more complex, but it is not necessarily a bad thing to happen. In fact, it should improve the final result of the design process, for example if designers iterate to correct errors or to reconsider alternative options (see also § 5.4.1). However, in the first evaluation study no relationship between the amount of iteration and the quality of the subjects' training programme designs was found (see Figure 6.2). To get more grip on the role of iteration in the design process, more insight is required in the events and circumstances that can or should cause iteration. In this section I propose a list of possible triggers for iteration, partly based on the results of the first evaluation study and partly based on observations, experiences and reflection. Subsequently, I discuss to what extent these different kinds of iteration are supported by the process-oriented support implemented in the MASTER-TPD prototype tool (as described in § 5.4.2).

7.2.1 Five triggers for iteration

In Section 5.4.1, I described four different types of iteration defined by Wagenberg (1992):

1. Iteration as repetitive activity: repeating the same design activity.
2. Iteration to correct errors: reviewing the concept design and correcting errors.
3. Iteration to improve the design: fundamental revisions to get to a better design.
4. Mutual iteration: this form can also be seen as a form of decomposition and will, therefore, not be taken into account here.

In the context of designing simulator training programmes procedures for the acquisition of (expensive) instructional products, that are obligatory in large organisations (see § 3.3.6), are an obvious trigger for iteration as repetitive activity. Iteration to correct errors will occur, especially because designers are likely to get new information about the domain that they are working with and because the project's constraints often change during the design period (see § 3.2.1). Iteration to improve the design can occur when the designers get new ideas or insights themselves, but also when new arguments or ideas come up in discussions with others, either stakeholders or peers. Five triggers that can evoke iteration during the design of simulator training programmes are described below (see Figure 7.1).

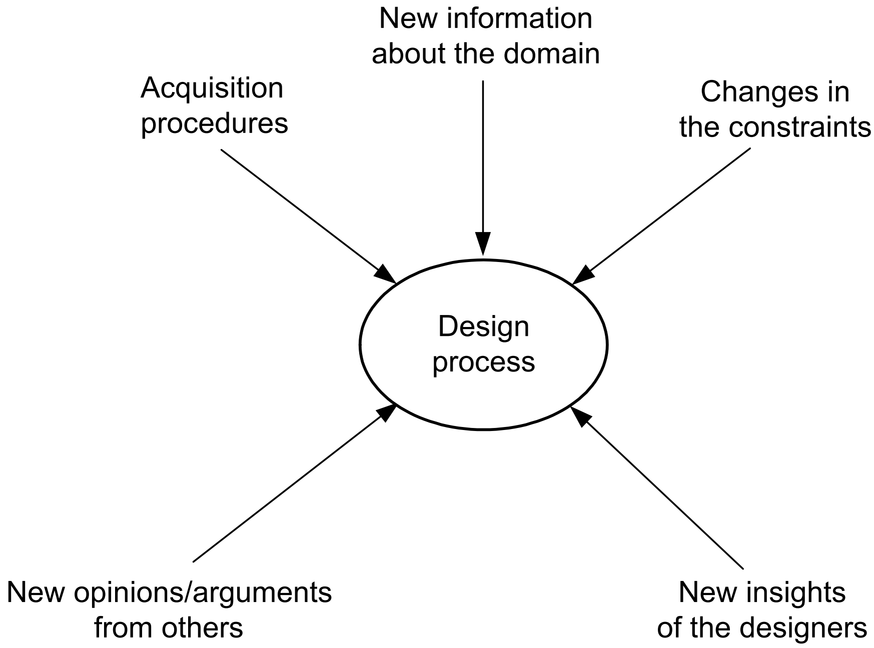


Figure 7.1: Five triggers for iteration.

Trigger 1. New information about the domain
New information about the domain, for instance, new information about the tasks of operators, the missions that the operational system will be used for, the interface of the operational system, or the background of trainees that will be trained.

Trigger 2. Changes in the constraints
Financial, technical, organisational and other constraints can change anytime during the design process, e.g., the budget that is available for the design and/or acquisition of instructional products, or the number of instructors that will be available for the delivery of training. The organisation's policies may change, dictating, for example, that field training should be minimised because of environmental pollution, or that instructional products must be bought 'off-the-shelf' when possible.

Trigger 3. New insights of the designers
Sometimes, further elaboration gives more insight into the consequences of earlier decisions. An example may clarify this: when the designers are specifying training activities, they might realise that the training strategies selected earlier do not really suit the type of training goals they are working on. The MASTER-TPD method prescribes that, in this case, they should go back to Step 1.1 to change the training strategies and go through the intermediate steps to see whether this revision has consequences for their design, e.g., for the sequencing of training goals in Step 1.2.

Trigger 4. New opinions/arguments from others
Discussions with other parties involved in the design process can cause iteration as well. Other people may bring forward arguments that the designers were previously unaware of. A

domain expert, for example, could argue against simulator training because trainees never experience the stress of a war situation in a simulator. Or an instructor might argue for more controlled training scenarios because then it is easier to give effective feedback. Even when the designers do not agree with these arguments, they will have to spend some time to defend their decisions and explain why they have not changed their design.

Trigger 5. Acquisition procedures within an organisation
As the acquisition of training simulators is expensive, it is usually regulated by procedures that enforce an iterative design process, for instance by requesting milestone documents (see for an example § 3.3.6.1). To obtain resources for the design and acquisition of training simulators, designers usually have to present and defend their choice for this type of instructional product early on. This means that they will have to make a first draft quickly, taking preliminary decisions based on quick, global analyses and incomplete information.

As discussed in Section 5.4.1, acquisition procedures may force designers to go through the whole design process several times, going into more and more detail. This fits in with the idea that the MASTER method can be used at different levels of detail for different purposes (see § 4.2.2 and § 5.4.1).The other four triggers described can occur at any moment during the design and may thus enforce iteration within one design cycle. I will discuss possible triggers for iteration again in Section 10.2.

7.2.2 Process-oriented support for triggers

In this section I discuss whether the triggers listed in Section 7.2.1 have played a role during the first evaluation study, and if and how they are addressed by the process-oriented support implemented in the MASTER-TPD tool (as described in § 5.4.3).

Trigger 1. New information about the domain
In the first evaluation study information about the domain was provided during meetings. Subjects could also take the initiative themselves and send questions to the domain expert or look for information in the reader. Some of the process-oriented guidelines explain when (i.e. in which steps) to look for additional domain information, for example:
- In Step 4.1 (Make Assessment Points): "Ask domain experts for information about the criticality of knowledge and skills for the execution of the future tasks of the trainees".
- In Step 5.2 (Describe Content of Scenarios): "Ask domain experts for additional domain specific information to fill the slots, if necessary".

Trigger 2. Changes in the constraints
There was no explicit change in the constraints during the first evaluation study since the planned intervention about the costs of the subjects' design was not executed (see § 6.3.6.4). However, it is possible, and even likely, that subjects found new information about financial, technical or other constraints in the reader or asked the experts questions about constraints by e-mail. Some of the process-oriented guidelines direct the designers' attention to possibly important constraints and encourage them to check whether the information they have is still correct. For example:
- In Step 3.1 (Construct Simulator Course Map): "Go for maximum simulator coverage when you want to automate instruction and feedback as much as possible, e.g., because you know that the number of instructors that will be available for training will be very limited".
- In Step 3.1 (Construct Simulator Course Map): "Start with designing a minimal set of specifications, when you know that the available budget is small".

Trigger 3. New insights of the designers
The third trigger for iteration, new insights cannot be planned beforehand. Creativity and design styles play a role here. The results of the first evaluation study showed that frequent iteration can be part of an effective design style, but also of an ineffective design style. Some

of the process-oriented guidelines describe the relationships between steps, thus trying to support situations where designers are likely to get new ideas, for example (see § 5.4.3.1 for more examples):
- In Step 4.1 (Make Assessment Points): "Planning assessment will force you to reconsider which knowledge and skills are prerequisite for others. This may give you reasons to reorder the training goals. Go back to Step 1.2 to review the training sequence. Look at your notes before you make any changes."
- In Step 5.1 (Make Training Scenarios): "Making scenarios may give you reasons to make changes in your assessment plan. Go back to Step 4.1 to add or delete assessment points, or to Step 4.2 to change the goals and/or kinds of assessment. Look at your notes before you make any changes."

The warning that appears when only a small number of training activities is allocated to simulator training is a more explicit form of advice to look for other, possibly more efficient solutions:
- In Step 2.2 (Determine Suitability for Simulator Training): "It seems that the majority of your training can be provided without a simulator. Do you really need a simulator? Or can the remaining part be trained during field training or on the job?"

Trigger 4. New opinions/arguments from others
New opinions and arguments from others occurred in the first evaluation study in the form of discussions during face-to-face meetings, contact with experts through e-mail and contact with each other in the web-based discussion groups. Some of the process-oriented guidelines draw the designers' attention to moments when it may be opportune to consult others. For example:
- In Step 1.1 (Select Training Strategy): "Ask instructional design specialists for advice on suitable training strategies for specific trainees or training goals"
- In Step 4.1 (Make Assessment Points) + Step 4.2 (Define Goal and Kind of Assessment) "Ask instructional design specialists for advice on assessment strategies, and the link between assessment plans and training strategies".

Trigger 5. Acquisition procedures within an organisation
The last trigger for iteration is related to the acquisition procedures that are adopted within an organisation. During the first evaluation study this was mimicked by asking the subjects to first go through all the steps quickly to make a global training programme design and then go through all the steps in more detail. The division between global and detailed design is supported by explicit descriptions of these design cycles in the Manual, in the Training Cards, and in the process-oriented guidelines (see for an example § 5.4.3.4).

7.3 Design of the second evaluation study

In the second evaluation study triggers for iteration were planned beforehand in the form of the following events (further described in § 7.3.6.3):
- Contact with domain expert: subjects participated in a group conversation with a domain expert who provided new information about UAVs and thermal images (Trigger 1).
- E-mail about resources: the subjects received an e-mail from the project manager with new information about the budget available for the design and acquisition of the training simulator (Trigger 2).
- Peer review: the subjects were asked to review the work of another subject and to give helpful comments. The expectation was that subjects would get new ideas both from looking at somebody else's solution to the same design problem and from the comments they received on their own work (Trigger 4).

- Discussion with peers: the subjects participated in a chat session (in two groups) and discussed about predefined questions related to the subjects' design task in order to learn from each other's opinions and arguments (Trigger 4).
- Scheduled global design phase (day 1) and detailed design phase (day 2-5) to mimic acquisition procedures within an organisation (Trigger 5).

Note that the third trigger, i.e. new insights of the designers themselves, cannot be planned.

Furthermore, the subjects worked under supervision and the time available for design task was kept under control. Otherwise, the design of the second evaluation study was similar to the design of the first evaluation study. Therefore, I will describe only those aspects that were different below.

7.3.1 Research questions

The second evaluation study addresses the same research questions as the first (see § 6.3.1), except for Question 2d (Do subjects use the web-based discussion groups to get support from each other?), because the web-based discussion groups were not available in this study. Two extra research questions regarding iteration were defined for this second evaluation study:

- Do subjects iterate when expected, i.e. after the events introduced to evoke iteration? Do subjects iterate at other moments?
- What is the subjects' opinion about the interventions planned to evoke iteration?

I had planned to look at differences between novice instructional designers and subjects with a few years of experience in instructional design and research. However, there are not enough data available because the data of one of the experienced subjects cannot be used due to technical problems (see § 7.3.5).

7.3.2 Setting

In the second evaluation study the subjects received the same assignment and worked with the same MASTER-TPD method and tool as in the first study. Subjects were asked to imagine that they were temporary employees of TNO hired to work on this project for a customer. They worked individually, except when discussions were deliberately introduced. A realistic task description was provided by e-mail at the beginning of the study (see Appendix D). Just as in the first study, different kinds of experts could be approached for advice through e-mail at any time. In this second study the web-based discussion groups were not available, since the results of the first study show that these were not used much and not appreciated.

The second study was executed in a more controlled setting. Five subjects worked in a computer room at the University of Twente under supervision of a TNO research assistant. The other four subjects worked on laptops (separate from their own computers) in their own rooms at TNO Human Factors, under the supervision of the author of this thesis. During the evaluation study different events that were expected to evoke iteration were introduced at predefined moments (see § 7.3.6.3).

To exclude the influence of time on task on the quality of the resulting training programme designs the available amount of time was fixed. During the first evaluation study the subjects that produced acceptable designs (i.e. passed the course) spent between 45 and 121 hours on the design task (see § 6.4.1.2). For the second evaluation study a time period of one full week, i.e. five successive working days was chosen under the assumption that this would be enough for subjects to design a first draft of a simulator training programme (obviously, more time would be required to elaborate the training programme to the level of detail required for simulator specification or the implementation of training scenarios). The study lasted for a total of 35 hours (see also § 7.3.6).

In the limited time available there was no time for the face-to-face meetings and demonstrations regarding the domain that were carried out in the first study. Instead, the domain was briefly introduced by the supervisor. The description given in Section 6.2 was provided on paper and background information about the domain could be found in the reader. Subjects were advised to approach the domain expert (by e-mail) for more information, if necessary. There are no indications that this limited introduction to the domain has hindered the subjects in executing their task (see § 7.4.1), but it may have forced them to ask more questions about the domain (see § 8.2.3).

7.3.3 Materials

Just like in the first study, subjects received:
- a printed version of the introduction about the MASTER-TPD method and tool as provided at the beginning of the study (text with screen prints),
- the MASTER-TPD glossary (see Appendix B), and
- a reader with background material on ISD, the MASTER method in general and the TPD phase in particular, and about thermal images and Unmanned Aerial vehicles (UAVs).

Additionally, the subjects in this study received:
- an explicit description of their task by e-mail (see Appendix D),
- a description of the domain (i.e. the text of § 6.2 of this thesis), and
- assignments regarding the two design phases and the other interventions were provided by e-mail and on hand-outs.

The full text of all e-mails can be found in the appendices of Verstegen et al. (in press).

Before the evaluation study started, the MASTER-TPD tool (see § 5.2) was installed on computers in a computer room at the University of Twente and on laptops for those subjects that worked at TNO Human Factors. The subjects at the University of Twente could e-mail questions to experts using the button 'Experts' on the Notepads (see Figure 5.3). Due to technical problems this was not possible with the laptops used at TNO Human Factors. These four subjects used normal e-mail software to send e-mails to experts. All subjects used normal e-mail software and their own e-mail address to receive e-mails. For the chat-sessions the subjects used Microsoft NetMeeting, version 2.1 (1997).

7.3.4 Data gathering methods

Just as in the first study the following data were collected by the MASTER-TPD tool:
- log files with the subjects' actions with the tool with time stamps (see § 5.2.1),
- the subjects' training programme designs, and
- the subjects' notes on the Notepads.

The final versions were saved at lunch time on the last day, as the afternoon was used for presentations. On the other days a back-up of the subjects' work was made at lunch time and at the end of the day (except for lunch time on the first day because subjects started just before lunch, see Table 7.2).

During the second evaluation study three questionnaires were issued:
- questionnaire 1 before subjects started working on their task (see § 7.3.6.1), concerning the subjects' background,
- questionnaire 2 after the chat sessions on the fourth day, concerning the subjects' need for additional support not yet provided by the MASTER-TPD method and tool (Note that due to circumstances this questionnaire was only completed by seven of the eight subjects), and
- questionnaire 3 at the fifth day before the presentations concerning the subjects' opinions about the MASTER-TPD method, its usability and usefulness.

Questionnaire 1 and questionnaire 3 were adapted versions of questionnaires used in the first evaluation study. All questionnaires can be found in Verstegen et al. (in press).

Other data collected during this second evaluation study are:
- the questions sent to experts by e-mail,
- the files with the subjects' comments on each others' designs (see § 7.3.6.3),
- log files of the chat sessions (see § 7.3.6.3),
- audio-tapes of the discussions with the domain expert (see § 7.3.6.3),
- video-tapes of the subjects' presentations at the last day (see § 7.3.6.4),
- the slides that subjects used during their presentations, and
- ratings of the presentation by two experts on the quality of the presentation, their impression of the quality of the training programme design, and the quality of the explanation/defence of design decisions.

The video-tapes, audio tapes and slides did not provide additional information that was not available from other sources. Therefore, they have not been formally analysed.

Just like in Chapter 6, the results of this study will be presented in tables and figures. I have computed correlations only as a way to find possibly interesting results, since the validity of correlation coefficients is not guaranteed with such a limited number of subjects. The results of these computations are reported in footnotes. The Spearman Rho Correlation Coefficient (SRCC) has been used for data on an ordinal scale (i.e. rankings) and the Pearson Product-Moment Correlation Coefficient (PPMCC) for data on an interval scale.

In the log files the times are given in seconds. For all the calculations the original measures in seconds were used. For readability purposes these times have been translated into hours and minutes in the tables (rounding seconds to the nearest minute).

7.3.5 Subjects

Two categories of subjects participated in the second evaluation study:
1. Seven students: five students of Educational Science and Technology (third year or higher), one subject who had just graduated in Applied Communication Science (having done courses in Educational Science and Technology), and one student Educational Psychology (final year).
2. Two TNO colleagues with a background in primary school teaching, a degree in Educational Sciences and a few years experience in instructional design and research.

The two TNO colleagues completed the assignment in work time, the other subjects were paid for their participation in this study.

One of the TNO subjects had serious technical problems: his work of the first day got lost because the MASTER-TPD tool had not been installed correctly on his laptop. He did work on until the end of the study, but indicated clearly that the had not been able to complete his task due to lack of time. Therefore, this subjects' data have not been included in the analyses that are reported here. Table 7.1 gives an overview of the background of the remaining eight subjects based on the results of the first questionnaire:

Table 7.1: Subjects' background (Subject 2-8)

Background	yes	no
Followed course(s) in instructional design	8	0
Followed course(s) in selection and/or design of instructional products	7	1
Followed course(s) on interface design/other kinds of design	6	2
Experience with (instructional) design tools	5	3
Experience in designing instruction	3	5
Experience in teaching	1	7
Experience as instructional scientist	3	5
Experience with training simulators	0	8

The subjects can be regarded as novices since none of them had experience in the design of training simulator specifications or the use of the MASTER method. They did, however, have theoretical knowledge relevant to the design task they were asked to perform during the evaluation study: they had followed one or more courses on instructional design, and most of them also on other kinds of design and on the selection and/or design of instructional products. The students from the University of Twente had experience with instructional design tools that were used during courses. The one remaining TNO colleague had experience in teaching, designing instructional products and research. Two students had some research experience participating as an assistant in research executed by university lecturers, and two students indicated that they had designed instructional products during university courses.

7.3.6 Procedure

The second evaluation study took place from July 3 to 7, 2000 (Monday to Friday). Subjects worked full time (i.e. eight hours a day minus half an hour lunch break and two coffee breaks of 15 minutes), for a total of 35 hours, including a 1.5 hours introduction on the first day and 4 hours for the presentation of the designs on the last day. Before the study started, all subjects received an invitation with background information, times and location. The complete text of all e-mails and hand-outs can be found in the appendices of Verstegen et al. (in press). Table 7.2 gives an overview of the procedure:

Table 7.2: Procedure of the second study

Activity	Including	Day	Time
Introduction	Explanation goals and planning Brief introduction MASTER method in general Introduction MASTER-TPD method and tool First questionnaire E-mail with task description (Appendix D)	1	9.00 -10.30 AM (or Friday before)
Global design phase	E-mail describing global design	1	10:45 hrs
	Four reminders indicating time schedule global design phase	1	13:00, 14:00, 15:00 and 16:00 hrs
	Reminder 'almost end of day'	all	16:30 hrs
Detailed design phase	Encouraging e-mail about results of first day E-mail describing detailed design	2	9:00 hrs
	Intervention 2: peer review	2	13:30 hrs
	Intervention 3: contact with domain expert	3	13:30 hrs
	Intervention 4: message from customer	4	9:00 hrs
	Intervention 5: discussion in chat session	4	13:30 hrs
	Second questionnaire	5	14:30 hrs
Finalising stage	E-mail with instruction for presentations	5	9:00 hrs
	Finalising detailed design Preparation presentations Third questionnaire	5	9:00 - 12:30 hrs
Presentations	Each subject 15 minutes	5	13:30 - 17:00 hrs

7.3.6.1 The subjects' task

The subjects' task was to design a simulator training programme with the MASTER-TPD method and tool in the context of designing specifications for a training simulator to be used for training the analysts of thermal images from a UAV (see § 6.2). The input for this task was a list of 54 training goals and a description of the target trainees (see Appendix C). The subjects could ask advice from the following experts through e-mail:

- Simulator-training-specialist: for questions about the MASTER-TPD method, and other questions about the design of simulators.
- Tool-specialist: for questions about the tool.
- Domain-specialist: for questions about the domain, i.e. interpretation of UAV thermal images.

- Instructional-design-specialist: for questions regarding instructional design and development theory.
- Project-manager: for all other questions.

The introduction took about 1,5 hours and was given on Friday afternoon to the four subjects that worked at TNO Human Factors and on Monday morning to the five subjects that worked at the University of Twente (in both cases by the author of this thesis). All subjects started on the design task after the coffee break on the first day, after receiving the background materials described in Section 7.3.3. The subjects were asked not to talk about their design task during breaks and to work on their own, except when explicitly asked to discuss with each other. Furthermore, they were told that the supervisor would only help in emergencies and that all other questions should be asked by e-mail.

7.3.6.2 Global and detailed design phases

The rest of the first day subjects spent making a global design of the training programme. They were asked to go through all the steps of the MASTER-TPD method quickly in order to design a first draft, making short notes on the Notepads. Subjects spent 5.5 hours in the global design phase (i.e. 8 hours minus 1.5 hours introduction and 1 hour lunch and coffee breaks). At the start of the global design phase subjects received an e-mail with a short explanation about the global design phase. They were told that making a global design was important to get an overview of the decisions that they would have to make during the design of a training programme and to get a rough idea of what their own design was going to look like. Furthermore, they were told that TNO and the customer would like to look at their global designs to check whether they were on the right track. During the day, they received reminders by e-mail roughly every hour, indicating which step they should be working on (more or less) in order to finish their global design at the end of the day.

At the beginning of the second day the subjects received an encouraging e-mail stating that both TNO and the customer thought that they were on the right track. A second e-mail contained a short explanation about the detailed design phase. Subjects were again advised to make notes on the Notepads with each step to use, for example, during the preparation of their presentation. The e-mail message also explained that it is normal that a design process is not a linear process, and that, therefore, it is possible in the MASTER-TPD method and tool to go back to earlier steps to make changes at any moment.

In the detailed design phase subjects had a maximum of 3.5 days to elaborate their design further in detail. They worked at their own pace. Each day at 16.30 hours they received an e-mail to remind them that they had 30 minutes left. One subject (Subject 3) of the Soesterberg group started late on the third day due to a work appointment and compensated this time in the evening of that day. She could not be present during the third intervention (described in § 7.3.6.3). The supervisor summarised the information provided by the domain expert to her. Two subjects at the University of Twente started a bit later on Friday morning, because they had an exam (Subject 8 started one hour later; Subject 9 started 15 minutes later). None of these subjects reported having problems with the amount of time available to complete the design task.

7.3.6.3 Interventions

Five interventions to evoke iteration were executed:

1) Explicitly scheduled global and detailed design phases (encouraged, not obligatory)
Just like during the first evaluation study, the subjects were asked to first go through all the steps quickly in order to make a global training programme design and then go through all the steps in more detail. Subjects were not obliged to follow this advice, but this iteration was encouraged by e-mails describing the usefulness of first making a global design, and re-

minders during the first day (see § 7.3.6.2). Furthermore, the division between global and detailed design is supported by explicit descriptions of these design cycles in the Manual, in the Training Cards, and in the process-oriented guidelines (see § 5.4.3.4 for examples).

2) Peer review (individual, obligatory)
Subjects were asked to review the work of another subject and to give helpful comments. At both locations the supervisors explained what was expected beforehand, and this explanation was also handed out to the subjects on paper. It contained indications of what to focus on: Did your colleague follow the MASTER-TPD method correctly? Do you agree with the decisions that were taken? Do you think that the arguments on the Notepads are convincing? How does this work differ from your own work? After the explanation subjects were asked to go to the computer of one of the other subjects (using a rotation scheme). Each subject individually looked at another subject's work either directly in the different steps in the tool (just as they had done their own work) or using the overview provided by the Report-function (see § 5.2.1). Comments were typed into a text file that was saved on the desktop. The maximum time for this intervention was one hour. Some subjects took less time while others said they had not had enough time to look at the whole design. In summary, each subject gave comments and received comments, and the comments were available as data for further analysis.

3) Contact with domain expert (groups, obligatory)
Subjects listened to a short presentation from a domain expert and then asked questions. The domain expert was a TNO staff member who had been involved in the analysis of training goals that served as a basis for the case description (see § 6.2 and Appendix C). Contact was made by speaker phone in two groups: first the domain expert spoke with the subjects that worked at TNO Human Factors and then with the subjects who worked at the University of Twente.

4) Message from customer about resources
At the start of the fourth day subjects got an e-mail with new information about the resources of the customer. The e-mail stated that the TNO project leader had visited the customer the day before and had heard that there were already contacts with several companies interested in the UAVs and in the training that the company was going to offer. The customer had also explained that it would probably be very difficult to find good instructors. Therefore, it was important that the training could be provided with a minimum number of instructors, even if that meant that the simulator would be more expensive.

5) Discussion in chat-session (groups, obligatory)
Subjects participated in a chat-session using their own computer and Microsoft NetMeeting, version 2.1 (1997). At both locations the supervisors gave an explanation of what was expected beforehand, and this explanation was also handed out to the subjects on paper. The maximum time for this intervention was one hour. Afterwards subjects could read a text file of the other group's discussion that they received by e-mail. To make sure that the subjects would see a reasonable amount of new arguments, the two groups discussed about different topics. The subjects working at the University of Twente discussed the following questions:
- What should be trained with the simulator? What not? And why?
- Which arguments can you use to defend your design to TNO and to the customer?

And the subjects working at TNO Human Factors discussed the following questions:
- How will the training design change now that there is more money available?
- In which ways can the required number of instructors be reduced?

7.3.6.4 Presentations

At the end of the study, all subjects presented their design at a group meeting in the presence of a panel of two simulator training experts. The four subjects who worked at TNO Human

Factors travelled to the University of Twente for the presentations on Friday morning. They compensated for this time on Thursday evening and/or by working on their laptops in the train.

In the instructions, sent by e-mail beforehand, subjects were asked to explain briefly but clearly their training programme design. They were told that the experts would be mainly interested in the explanation/defence of important design decisions, such as the choice of what is covered with simulator training and what not. All subjects had 15 minutes for their presentation (including questions). They used slides and a projector. Two experts with experience in the area of training with simulators and simulator specification (i.e. two TNO colleagues not involved in any of the other analyses) judged the quality of the designs as they were presented (see § 7.3.4).

7.4 Results of the second evaluation study

In this section the results regarding the following research questions will be discussed:

§ 7.4.1 Can subjects design a suitable training programme with the MASTER-TPD method?
§ 7.4.2 How is iteration used in the design process? How frequently do subjects iterate? Does more iteration lead to a better training programme design?
§ 7.4.3 Do subjects iterate when expected, i.e. after the events introduced to evoke iteration? Do subjects iterate at other moments?
§ 7.4.4 What is the subjects' opinion about the interventions planned to evoke iteration?

The discussion of the other research questions is postponed until Chapter 8, where the results of both evaluation studies regarding these questions are presented simultaneously and compared.

7.4.1 Usability of the MASTER-TPD method

7.4.1.1 Quality of the training programme designs

To judge the quality of the end products, i.e. the training programme designs delivered by the subjects at the end of the evaluation study, these were ranked them from the best (8) to the least good (1) following the same procedure and by the same experts as in the first evaluation study (see § 6.4.1.1). Ratings were also obtained from a third rater who had been a participant in the first evaluation study and had, therefore, a good understanding of the MASTER-TPD method and tool and of the subjects' task (Note that these three raters were not involved in judging the quality of the subjects presentations which are reported in § 7.4.1.2). The average scores were ranked again to come to an 'Average quality ranking' to use for comparison with other results below (see Table 7.3).

The concordance between the raters is reasonably high (Kendall coefficient of concordance: W = 0.67, Average ranking correlation = 0.51; statistically significant[13]), but not as high as during the first study (see Table 6.3). It is probable that the differences in the quality of the design were much smaller in this study because the time spent on the design task was more or less the same for all subjects (and in the first study the time on task correlated with the quality of the resulting training programme designs, see § 6.4.1.2). When the differences in quality are relatively small, the raters' individual preferences can have a greater influence on the results. Indeed discussions between the raters afterwards indicate that they had used the same criteria (that had been given in the instruction, see § 6.4.1.1) but with slightly different priorities. Because the quality rankings are less concordant than in the first study I have also looked at relations between the rankings of single raters and other data presented in this chapter.

[13] Friedman ANOVA: Chi Sqr. (n = 3; df = 7) = 14.1; $\alpha < 0.05$.

There were, however, no occasions were those comparisons gave a different view on the results[14].

Table 7.3: Quality of training programme designs[a]

	Ranking Rater 1	Ranking Rater 2	Ranking Rater 3	Average	Average quality ranking
Subject 2	1	2	1	1.3	1
Subject 3	8	7	6	7	8
Subject 4	7	3	3	4.3	4
Subject 5	2	4	2	2.7	2
Subject 6	5	6	7	6	6
Subject 7	6	8	5	6.3	7
Subject 8	4	1	4	3	3
Subject 9	3	5	8	5.3	5

[a] Rankings from best (8) to least good (1).

Finally, I want to report some of the raters' general impressions:
- In the raters' opinion all subjects had taken their task seriously. They felt that the quality of all eight training programme designs was acceptable. According to the two raters that evaluated the results of both studies (Rater 1 and 2) the average quality of the designs of this second study was at least as high as in the first study (and the best of those were highly valued by domain experts, see § 6.4.1.4).
- The best designs show creativity on the part of the designer (Rater 1).
- Sometimes the explanations in the notes do not correspond with the design (Rater 2).
- Most subjects still get to a rather standard 'boring' design: theory followed by practice, a design that could be used in almost any domain. The fact that many of them choose for a mastery learning concept does not show a good understanding of the learning process: this is a task that cannot be learned by simply practising until perfection (Rater 2).

7.4.1.2 The presentations

At the last day subjects presented their training programme design to each other and to a panel of two experts with knowledge and experience in the field of training with simulators and simulator specification. The experts scored, amongst others (see § 7.3.4), their impression about the quality of the training programme design on a four-point scale: bad-reasonable-good-very good. Based on average scores the subjects were ranked from best (8) to least good (1).

Table 7.4: Impression quality of designs during presentations

	Score Expert 1[a]	Score Expert 2[a]	Average scores[a]	Ranking based on scores[b]	Average quality ranking (from Table 7.3)
Subject 2	4.0	4.0	4.0	8	1
Subject 3	3.0	2.5	2.8	3.5	8
Subject 4	3.0	3.0	3.0	5	4
Subject 5	3.0	2.5	2.8	3.5	2
Subject 6	3.0	2.0	2.5	2	6
Subject 7	2.0	2.0	2.0	1	7
Subject 8	4.0	3.5	3.8	7	3
Subject 9	3.0	3.5	3.3	6	5

[a] Scores on a four-point scale: very good (4) to bad (1). Rater2 chose in-between scores sometimes.
[b] Rankings from best (8) to least good (1). Equal intermediate rankings were given for equal grades (Expert 2) or equal averages.

[14] There is a relation between the ranking based on the quantity of notes that subjects made (see Table 8.2-B) and the quality rankings of Rater 2 (SRCC = 0.67) and Rater 3 (SRCC = 0.76, i.e. significant), and this is not the case for Rater 1 (SRCC = -0.05). This does not, however, change the general conclusion that good performers make more notes than less good performers (see § 8.2.2).

Table 7.4 shows that these rankings differ considerably from the average quality ranking based on prints of the designs (from Table 7.3)[15]. This can mean two things:
- the differences between subjects in presentation skills overshadowed the differences in the quality of the designs, or
- these experts have different opinions about the quality of training programmes then the raters in § 7.4.1.1.

In both cases, it supports the hypothesis that the quality differences are relatively small.

7.4.1.3 Time spent on the design task

From the log files the amount of time that subjects were logged into the tool has been derived in the same way as for the first study (see § 6.4.1.2). One subject (Subject 9) forgot to shut the MASTER-TPD tool down before the start of the presentations on Friday afternoon. Therefore the end of her last session was set to the time of her last action in the log file manually. Note that subjects may have spent a part of the time that they were logged into the tool on reading background information, sending and reading e-mails, etc.

The eight subjects were logged into the MASTER-TPD tool roughly between 25 and 30 hours (average 27 hours and 58 minutes; standard deviation: 1 hour and 57 minutes). A considerable part of the differences in time on task is caused by the fact that some subjects logged out during coffee breaks (twice 15 minutes) and lunch breaks (30 minutes) and others did not. Small differences remain because some subjects stopped a bit earlier on the fourth day (see below), whereas others continued working on their design during the morning of the fifth day (see § 7.3.6.2). There are, however, no indications that these differences had any influence on the quality of the resulting training programme designs[16].

Even though the subjects had less time than during the first study, all subjects designed a first draft of a simulator training programme and presented their design at the end. Two of the eight subjects said during their presentation that they had not been able to completely finish the last step of the MASTER-TPD method within the given time. A third subject said that she had not had enough time to reconsider which part of the training should take place with the training simulator after the message with new information about the resources available for simulator design and acquisition (i.e. Intervention 4). On the other hand, three other subjects claimed on the morning of the fourth day (before the coffee break) that they had finished their task. The supervisors asked them to have another look at their designs to review and/or elaborate their work. All subjects worked at least until half past three, i.e. at least an hour after the chat sessions (Intervention 5).

7.4.1.4 Subjects' opinion about the usability of the MASTER-TPD method

The data presented in Table 7.5 are results from the third questionnaire issued at the end of the evaluation study just before the presentations. The subjects were positive about the MASTER-TPD method: they think that it is useful and that it supports the design and acquisition of training simulators. Most subjects (5) would use the method again if they had to design a training programme or specifications for a training simulator, although some comment that they would combine it with other methods. One subject saw no advantages of this method compared to others. In their comments three subjects said that working with the method was sometimes boring, mainly because there was -within the limited time available- not much time for other activities, such as reading background material, discussions, etc. A fourth subject stated that there was not enough time for reflection.

---

[15] SRCC Ranking average score Impression quality during presentations/Average quality ranking = -0.70.

[16] SRCC Time on task/ Average quality ranking = 0.12; Time on task/Quality ranking Rater 1 = -0.17; Time on task/Quality ranking Rater 2 = 0.12; Time on task/Quality ranking Rater 3 =0.38.

Table 7.5: Opinions about the MASTER-TPD method (Subject 2-9)[a]

Question	Mean score	Standard deviation
You have worked with the MASTER-TPD method this week. Did you find that:		
# Difficult----------------Easy	3.1	1.0
# Not useful--------------Useful	3.9	0.8
# Boring -----------------Exciting	3.1	0.8
The MASTER method is meant to support the design and acquisition of training simulators. Did you find this support:		
# Insufficient------------Sufficient	3.8	0.5
# Not useful-------------Useful	3.9	0.4
Suppose you would work as instructional designer in a company: Would you use this method if you had to design a training programme or specifications for a training simulator?		
# Definitely not---------Yes, certainly	3.4	0.7
Would you advice a colleague to use this method?		
# Definitely not---------Yes, certainly	3.5	0.5

[a] Scores on a 5-point scale from 1 (left side) to 5 (right side).

7.4.1.5 Conclusions

The first research question addressed in this chapter was whether the subjects can design a suitable training programme with the MASTER-TPD method. In this second evaluation study the subjects performed their task in a more structured setting under supervision and within a fixed amount of time. Even though they had less time then the average during the first evaluation study, all subjects were able to complete the assignment and all designs were judged to be acceptable. This is promising: the subjects in these evaluation studies are novices, but when they work seriously on the design task with the MASTER-TPD method and the accompanying support they can make a reasonably good draft of a simulator training programme without major problems. No doubt an expert instructional designer specialised in the design of training simulators could do better (see, e.g., the comments of Rater 2 in § 7.4.1.1), but experts are often not available in practice (see § 3.2.1).

The usability of the MASTER-TPD method is confirmed by the fact that the subjects did not think that working with it was difficult (see Table 7.5). Moreover, it proved possible to execute the planned procedure without major changes. In contrast to the first evaluation study, there were no indications that the subjects felt insecure. One subject suggested feedback on completed steps as a possible improvement in the second questionnaire (see § 5.4.2), but none of the subjects asked for feedback during the study. A probable cause for this difference is that the subjects in this study participated in work time or were paid for their participation. This means that failure would not have serious consequences for them, whereas in the first study failure meant failing an important course and not receiving the study credit points for that course (see also the discussion in § 6.4.1.4). A second reason might be that the supervisors were available to help with acute technical problems. This only happened a few times and the supervisors never gave feedback on the design process or product, but just the fact that they were present may have had a soothing effect.

Another difference with the results of the first study is that the quality rankings of the three experts show considerable differences. It is probable that the quality differences were much larger in the first study due to large differences in the amount of time spent on the design task. In this study the effect of time on task was ruled out and, probably, the quality differences are much smaller. When the differences in quality are relatively small, the raters' personal preferences can have a greater influence on the results.

7.4.2 The use of iteration

7.4.2.1 The number of iterations

The number of iterations was derived from the log files in the same way as for the first study (see § 6.4.3.1). Table 7.8 shows that all subjects iterate. The number of iterations ranges from 10 to 30 (average: 18.6; standard deviation: 7.9). There is also evidence of iteration in the e-mails to experts, in the subjects' notes, and in the subjects' presentations.

Table 7.6: Number of iterations

	Number of iterations	Ranking based on number of iterations[a]	Average quality ranking (from Table 7.3)
Subject 2	28	7	1
Subject 3	13	4	8
Subject 4	24	6	4
Subject 5	12	2.5	2
Subject 6	20	5	6
Subject 7	10	1	7
Subject 8	12	2.5	3
Subject 9	30	8	5

[a] Rankings from best (8) to least good (1).

The results in Table 7.6 show that the best two subjects do not iterate many times (Subject 3 and Subject 7), but neither do two of the three worst performers (Subject 5 and Subject 8). On the other hand, the subject that got the lowest rating does iterate many times (Subject 2). The other three subjects with a high number of iterations have an average performance (see Table 7.6). One of the reasons could be that the average quality ranking is not a very trustworthy measure for quality since the three raters gave different judgements[17] (see Table 7.3). However, in the first study, where the raters' judgements correlated highly, there was also no relationship between the number of iterations and the quality of the results. Figure 7.2 shows that the number of iterations does not correlate with the quality of the results[18].

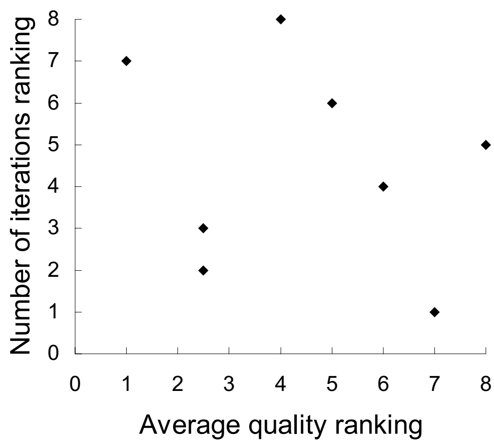


Figure 7.2: The ranking based on the number of iterations (from Table 7.6) versus the average quality ranking (from Table 7.3).

7.4.2.2 The subjects' opinion about iteration

The third questionnaire contained a set of open questions about iteration during the design process. Six of the eight subjects said that they had iterated a lot. The other two indicated that

17 There are also no relations between the ratings of individual raters and the number of iterations.

18 SRCC Ranking number of iterations/Average quality ranking = -0.23; Ranking number of iterations/Quality ranking Rater 1 = -026; Ranking number of iterations/Quality ranking Rater 2 = -0.28; Ranking number of iterations/Quality ranking Rater 3 = 0.38.

they would have iterated more if the tool would have been more user friendly (see below). When asked whether they had iterated because some of the guidelines advised them to do so, only one subject said 'yes, sometimes'. Five subjects answered negative and two subjects did not answer this question. The reasons for iteration they mentioned are:
- new insights or discovering later on that the initial choices were not the best (7 subjects),
- new information e.g., from answers from experts by e-mail or from the e-mail about the resources available for simulator design and acquisition (3 subjects), and
- made mistakes initially because misunderstood the purpose of one step of the MASTER-TPD method (1 subject).

This last subject had misunderstood the purpose of Step 2.1 and then found that she could not continue in Step 4 because her training activities were not defined in enough detail. This also happened during the first evaluation study, see the discussion about the feedback provided to subjects in Section 8.3.2.1.

Most (7 subjects) said that the MASTER-TPD method facilitates iteration. Some subjects are not positive about the tool in this respect, for the same reasons as the subjects in the first evaluation study: the Course Map does not show all the training activities, but only those that will take place with the simulator; and sometimes sequencing information gets lost when the Course Map is reconstructed (see § 6.4.3.2).

7.4.2.3 Conclusions regarding iteration during the design process

Just like in the first study there is no correlation between the number of iterations and the quality of the resulting training programme designs. Actually, the number of iterations in this second study (range: 10-30; average: 18.6) is very similar to the number of iterations in the first study (range: 5-37; average: 17.6), even though the subjects in this study had less time to work on their training programme design. It is, of course, possible that the interventions stimulated the subjects to iterate more than they would have done otherwise. In any case, the results of the second study are further evidence for the hypothesis that, although iteration during the design process is unavoidable, the timing and amount of iteration depends on characteristics and preferences of the designer, and that different design styles -with more or less iteration- can lead to good results.

The subjects in this study were also not enthusiastic about having to iterate. However, in general, they objected much less than the subjects of the first evaluation study even though interventions to evoke iteration were executed. It is possible that subjects were less insecure, because they were hired for this task and did not depend on its results for academic credits. Maybe, also, it is easier to iterate in such a concentrated period of time when subjects can still clearly remember what they have decided yesterday and why.

In the list of triggers for iteration presented in Section 7.2.1 at least one trigger for iteration seems to be missing: iteration to repair errors that designers discover themselves (see § 7.4.2.2). Furthermore, closer analysis of the feedback given to subjects in the first study made clear that they sometimes needed to iterate because important input for later steps is not sufficiently elaborated (see § 8.3.2.1). Thus, it seems that the trigger 'New insights of the designers' can cause iteration in different ways:
- iteration because errors are discovered,
- iteration because important input for a later step is missing,
- iteration because later decisions require changes in earlier steps, and
- iteration caused by unrelated new ideas (the unexpected 'Aha' or 'Eureka').

On the other hand, the triggers 'New information about the domain' and 'new information about resources' seem to be similar in nature. They have similar consequences and can be supported in the same way. From a theoretical viewpoint they could be combined into one trigger called 'new information'. See Section 10.2 for further discussion of the list of triggers for iteration.

7.4.3 The interventions to evoke iteration

7.4.3.1 Iteration after interventions

The log files show whether subjects iterated after the interventions. For the first intervention (the scheduled global and detailed design phase) this was easy: the log files show that subjects went through the MASTER-TPD method once, in more or less one day, and then back again to the first step. For the other interventions, the decision of what to count as iteration caused by an intervention is a bit more complicated, since subjects may have other reasons to iterate at that moment (by chance) or may want to finish what they were doing first before iterating back to an earlier step. Therefore, the following criteria have been used to decide whether subjects iterated after an intervention:
- subject went back to a step that is relevant to the intervention, and did so
- within three actions in the log file, and
- within one hour after the intervention.

Table 7.7: Iteration after interventions

	1. Global and detailed design	2. Peer review	3. Contact with domain expert	4. E-mail about resources	5. Chat sessions
Subject 2	*	x	*	-	*
Subject 3	*	x	*	-[b]	*
Subject 4	*	x	-	x	-
Subject 5	*	*	-[a]	*	*
Subject 6	*	*	*	*	*
Subject 7	*	*	-	-	*
Subject 8	*	x	-[a]	*	-
Subject 9	*	*	*	-	-

* Immediate iteration to a step linked to the intervention.
x Iteration to a step linked to the intervention within three actions and within one hour.
[a] Subjects were working in relevant step (Step 5.2) and continued after intervention.
[b] Subject went forward to relevant step (Step 6.1).

Inspection of the log files shows that all subjects understood the idea of the two design phases and applied it. The global design phase, however, was sometimes shorter or longer than one day (see § 7.4.3.2). After the second intervention (peer review) four subjects immediately went back to a step that their 'colleague' had commented on. The other four subjects did not do this immediately, but shortly afterwards. After the contact with a domain expert on the third day five subjects iterated back to a relevant step (Step 1.1, Step 2.2 or Step 5.1) and two subjects continued working in Step 5.2 for which the new information from the domain experts was also expected to have impact. The message from the project manager with new information about the customer's resources at the (Intervention 4) made three subjects iterate immediately and one a bit later. One subject jumped forward to Step 6 for which this information was also expected to have impact. Finally, five subjects iterated after the chat sessions.

Limitations of the data about iteration after interventions
The iterations counted in Table 7.7 are not necessarily caused by the interventions, they can be coincidence. This is most likely for Intervention 2, the peer review (see § 7.4.3.2). On the other hand, subjects also iterate in between interventions, which means they also go back to previous steps because of other reasons, for instance, when they get a new idea and want to adapt their design accordingly. Another reason, not included in the list in Section 7.2.1, may be that they discover that they have to go back to a previous step to elaborate their work in order to get enough input to go on.

7.4.3.2 The interventions in further detail

Intervention 1: Explicitly scheduled global and detailed design phases
The log files show that all subjects went through two design cycles. However, three of the eight subjects did not get through all the steps on the first day (see Table 7.8). Two of them continued with the last steps on the second day (for about two hours) before going back to the beginning (Subject 4 and Subject 8). The third one started with the detailed design phase immediately (Subject 5). Two subjects got through all the steps in less time than estimated, and they already started with the detailed design phase on the first day (Subject 6 and Subject 7). Apparently, the reminders of the time schedule that were sent on the first day did not hinder subjects working at their own pace.

Table 7.8: Global and detailed design

	Number of subjects
Global design not finished at the end of day 1	3
Global design finished exactly at the end of day 1	3
Global design finished before the end of day 1	2

Intervention 2: Peer review
Two TNO experts (the author of this thesis and a colleague) analysed the content of the reviews. Almost all the comments were directly relevant to a particular step. Two subjects gave a few general comments that were not relevant for the subjects' task and are, therefore, not included in the analysis. The raters also made a difference between affirmative statements like "I agree with you on..." or "In your design I like the way that..." and criticisms or recommendations for improvement (see Table 7.9). Especially, comments from the second category are likely to cause iteration.

Table 7.9: Amount of comments

	Comments from:	Criticisms and Recommendations	Affirmative statements	Total number of comments
Subject 2	Subject 3	10	2	12
Subject 3	Subject 4	2	2	4
Subject 4	Subject 1	9	-	9
Subject 5	Subject 6	12	4	16
Subject 6	Subject 8	6	2	8
Subject 7	Subject 5	5	5	10
Subject 8	Subject 9	9	9	18
Subject 9	Subject 7	15	4	19
Total		*68*	*28*	*96*
Average		*8.5*	*4*	*12*

The subjects gave between 4 and 19 comments (average: 12.0; standard deviation 5.3). Table 7.10 gives an overview of the number of comments regarding the different steps (see Figure 5.4 for an overview of the steps of the MASTER-TPD method). These data have been used to decide whether subjects iterated after this intervention (see Table 7.7).

All subjects iterated to steps that they had received comments on after this intervention, but this may be (partly) coincidence: Table 7.10 shows that most of the comments concern the first steps (Step 1.1, Step 1.2, Step 1.3 and Step 2.1). At the time of the intervention, the second day after lunch, subjects were likely to be working on those steps: they had just gone back to the beginning again at the start of the detailed design phase that morning. There are two reasons why the comments mainly concern the first steps of the MASTER-TPD: most subjects had not done a lot of work in the later steps during the global design phase so there was not much to comment on, and a few subjects commented that they did not have time to look at the last steps of their colleague's work within the hour that was available for this intervention.

Table 7.10: Comments per step

Step	Sub-step	Number of comments
Step 1	1.1	19
	1.2	26
	1.3	13
Step 2	2.1	20
	2.2	5
Step 3	3.1	1
Step 4	4.1	7
	4.2	2
Step 5	5.1	3
	5.2	-
Step 6	6.1	-
	6.2	-
	6.3	-

Intervention 3:Contact with domain expert

On the third day, after lunch, subjects had contact with a domain expert. The domain expert first gave a description of the UAV/thermal image analysis domain and then subjects had the opportunity to ask questions. Striking at both sites was that there were not many questions: for the subjects at TNO Human Factors the contact lasted 14 minutes and they asked two questions (Note that one of these four subjects could not be present, see § 7.3.6.2). For the subjects at the University of Twente the contact lasted 16 minutes and they asked six questions. These subjects came up with new questions after the contact with the domain expert had been finished. The supervisor advised them to ask any remaining questions to the domain expert by e-mail. The supervisor at the University of Twente also reported that after this intervention some of the subjects were frustrated by the thought of having to adapt their design again. One subject complained explicitly that this information should have been available at the beginning of the week. After the contact with a domain expert five subjects iterated back to a relevant step (i.e. Step 1.1, Step 2.2 or Step 5.1) and two subjects continued working in Step 5.2 for which the new information from the domain experts was also expected to have impact.

Intervention 4: Message from customer about resources

Even though not all subjects iterated according to the log files, this intervention seems to have had impact. During the presentations several subjects stated that this message had influenced their design. Two subjects explained explicitly how the tasks of instructors had been minimised in their training programme design. Another subject explained that she had planned to execute all training activities with the simulator at first. When she was making training scenarios she already realised that this was not always the best solution, and after the message from the project manager she felt that she would have had to go back again to reconsider which part of the training should take place with the training simulator. She did, however, not have time to do this anymore.

Intervention 5: Discussion in chat sessions

The maximum length of the discussion was set to one hour. The five subjects at the University of Twente were finished with their discussion after half an hour. The five subjects at TNO Human Factors had to be interrupted when the hour had passed. There are two reasons why this intervention may have been less successful in stimulating iteration:

- the subjects were not very enthusiastic about the chat session (see § 7.4.3.3), mainly because it was hard to keep up and follow the discussion without a moderator, and
- the intervention took place almost at the end of the design period.

Three raters scored the number of 'meaningful' contributions in the same way as for the subjects' notes (see § 8.2.2) and came up with very different results. Since the results provided no information with regards to the research questions, they have not been further analysed.

7.4.3.3 Subject's opinion about interventions

The subjects liked the peer review (average score 4.0 on a five-point scale; standard deviation: 0.6), although one subject commented that this activity took place too early in the week. Five of the eight subjects would have liked to have more contact with their colleagues during the study. The subjects were less positive about the chat session (average score 3.1; standard deviation: 1.2). Indeed the discussions were rather confused and difficult to follow: four to five people participated in the chat sessions, and there was no moderator nor facilities to structure the discussion visually. Pena-Shaff, Martin, and Gay (2001) found similar problems: an electronic chat environment was suitable for brainstorming, personal discussions and social interactions, but it was often difficult to follow the thread of the conversation. In their study, participants used chat sessions to reach consensus about the group's views that had to be presented later on the day. In this study, subjects did not have such a strong incentive: the chat session was used to reflect on questions related to training programme design and to get the opinions of peers on these questions. Pena-Shaff et al. found that asynchronous bulletin boards were more suitable for this kind of reflective articulation of ideas. However, the experiences with asynchronous discussion groups in the first study were also not very positive (see § 6.4.2), possibly because there was no moderator and subjects were not rewarded for their participation.

7.4.4 Conclusions

The most successful trigger for iteration in this study was the explicit scheduling of design cycles. This was a strong intervention: although subjects were not forced to go through a global design phase first, this was encouraged by explanations in e-mails, in the Manual and in the process-oriented guidelines (as described in § 5.4.3). This is not unrealistic. In practice, this kind of iteration is often enforced by organisational procedures (see § 3.3.6). The peer review also stimulated all subjects to iterate, and was highly appreciated as well.

Most subjects also iterated after the contact with the domain expert. The two least successful interventions were the last two (new information about resources and discussion in chat sessions). Maybe the timing of these interventions played a role: they took place later in the week (see Table 7.2) and subjects may not have had enough time to adapt their design according to the new information they received. It is also possible, though, that it was less clear to them what the consequences of these interventions should be. The comments provided by other subjects in the peer review were directly related to the subjects' work in specific steps. For the last three interventions this was not the case: new information or new ideas were provided without direct instructions as to how and where they could influence the design of simulator training programmes. There are at least two ways to support subjects regarding this aspect:
- providing general guidelines, or
- coupling scheduled meetings or discussions with design cycles.

These are further discussed in Section 10.2.

7.5 Discussion

7.5.1 Comparing the two evaluation studies

The second evaluation study confirms that the MASTER-TPD method can be used to design preliminary training programmes and specifications for simulators by users with some background in instructional design and development theory, but no knowledge or experience in the field of simulators or the specification of instructional products. Note, however, that there was additional support available in both evaluation studies: an introduction about the method, guidance and storage facilities and the help about the method provided by the MASTER-TPD tool, background theory in the reader and a method expert available to answer questions by e-mail (see Table 8.4 for the kind of questions that were asked). In Section 9.3, I describe a different approach used in projects with customers, where a similar method was applied in workshops with all stakeholders and with experienced instructional designers as facilitators.

In the first evaluation study it appeared that the amount of time spent on the design task had an important influence on the quality of the resulting training programme designs. In this study the time on task was kept under control, and the results show that the effect of time on task was indeed ruled out. There are indications that, in consequence, the differences in the quality of the resulting training programme designs are much smaller than during the first study.

Just as in the first study all subjects iterated, although some subjects were frustrated because not all information was immediately available and tried to avoid iteration. It is, however, typical for specification of training simulators that changes in the available resources occur (cf. Intervention 4), and that not all domain information is immediately available (cf. Intervention 3), either because domain experts are not present or because the information simply is not there when the operational system has not yet been bought or does not yet exist (see § 3.2.1).

The subjects in this second study gave no signs of the insecurity that was so obvious during the first study (see § 6.4.1.4), probably because they were paid for their participation, irrespective of their results. In practice, failure will have large consequences: careers and large sums of money are at stake, not to mention the dangers of badly trained personnel: damage to operational equipment and possibly even the loss of lives. Moreover, outside of research settings, the design process will probably be even more complex due to continuously changing financial, technical and other constraints, changes in personnel, interference by other stakeholders, etc. Therefore, feelings of uncertainty similar to those in the first study are expected. In fact, a short evaluation with real target users indicated that they feel a need for confirmation of the correctness of their decisions (see § 9.2).

The limitations of this second study are partly the same as those of the first study: a limited amount of subjects and some restrictions on the data that were gathered (see § 6.5.3). The effects of time on task were ruled out in this study, and the amount of time available seemed sufficient for the case that was used as design task. However, the subjects worked on this task in a very condensed period of time. In reality, designers will probably have to combine a design task like this one with other tasks, more like in the first study (see § 6.3.2). The fact that there is a strict deadline is not unrealistic, but the control and guidance during the study with, for example, assignments and reminders by e-mail would not be provided.

7.5.2 Discussion regarding iteration in the design process

The results of both studies confirm that iteration is inherent to the design process: all subjects iterate. However, some subjects iterate more than others. There are a number of possible reasons for this. It is possible that personal preferences or design styles play a role. It is also likely that the capabilities of the subjects play a role. Weak designers will make more errors and will have less insight into the possible consequences of early decisions. Good designers might have to iterate less to repair errors, possibly because they make a more thorough problem analysis beforehand. On the other hand, good designers may use iteration for another purpose: to check whether they have designed the most optimal solution or to consider possible alternatives. This would be in line with the findings of descriptive research which show that experienced designers do indeed spend more time and effort on a problem analysis, generate several alternative solutions, and do not commit themselves until later on (see § 3.2.2.3).

Not reacting to important triggers for iteration will almost certainly lead to spending time and effort on a non-optimal design that will not be feasible or will not be accepted in the organisation. However, iterating too much will make it almost impossible to keep an overview of the design process and to keep the design consistent. In order to further improve the process-oriented support future research should investigate in more detail when iteration is desirable and when not, and develop precise recommendations on when and how to iterate. For this purpose, the instructional design process needs to be closely followed and the reasons for iterating or not iterating should be registered. This will not be easy to do as it needs to be done in a realistic design process over a realistic period of time (so that triggers for iteration can occur and have effect) and without disturbing the design process itself. One option is to participate and observe design teams in real projects (ethnographic studies), or to introduce support for iteration and investigate the effects, in the tradition of formative development research (van den Akker, 1999; Reigeluth and Frick, 1999). A first attempt in that direction are the case studies presented in Section 9.3.2. Another option is to combine empirical studies similar to the ones reported in this thesis with shorter periods of registering 'think-aloud' protocols, preferably at significant moments in the design process. The disadvantages of this kind of research are that it is time-consuming, that it is difficult to organise and control, and that it is hard to isolate critical variables and compare the results between studies.

I will propose a list of possible measures to support iteration in Section 10.3. Suggestions for future research will be further discussed in Section 10.5.

Chapter 8
Results of two evaluation studies: further analysis

8

In Chapter 8, I use the results of both evaluation studies to discuss the effects of the different forms of support that were offered to the subjects. The research question addressed in this chapter is:

10. Do subjects use the different forms of support that are offered?

Furthermore, I will discuss the information that the two evaluation studies provided about the MASTER-TPD method in more detail.

8.1 Introduction

In this chapter I take a closer look at the results of both evaluation studies to discuss the research questions that were not addressed in Chapters 6 and 7. The questions regarding the different forms of support are addressed in the following sections:

§ 8.2.1 How much do subjects use the different types of help offered by the tool? Is the quality of their training programme designs related to the use of help? Is the number of iterations related to the use of help?
§ 8.2.2 Do subjects make notes? Is the quantity of notes related to the use of (different types of) help?
§ 8.2.3 Do subjects approach experts for support? If so, which kind of questions do they ask?
§ 8.2.4 What is the subjects' opinion about the different forms of support?

Subsequently, in Section 8.3, the remaining research questions regarding the MASTER-TPD method are addressed. I first discuss the subjects' opinion about the approach of the whole method, before going into detail discussing issues regarding the specific steps of the MASTER-TPD method:

§ 8.3.1 What is the subjects' opinion about the structured step-by-step approach of the MASTER-TPD method?
§ 8.3.2 Are subjects able to complete the steps of MASTER-TPD method? Which steps are difficult?

In Section 8.4, I discuss the most important differences between the subjects and settings used in the evaluation studies and training simulator specification in practice.

In this chapter the data of both evaluation studies are presented simultaneously and compared. The data presented here have also been reported in Verstegen, Barnard, and Pilot (2003) and Verstegen, Barnard, and Pilot (in press) respectively. To distinguish between the two studies I will in this chapter refer to the first study (described in Chapter 6) as study A and to the second study (described in Chapter 7) as study B. I will also use the A and B to refer to specific subjects, for example, Subject A7 is the seventh subject from the first study. Tables are split into an A part and a B part when the results of the two evaluation studies cannot be presented in one table. Finally, when the results are presented in figures, the figure on the left concerns study A and the figure on the right concerns study B.

8.2 Different forms of support

Just like in the previous chapters measures of time are presented in hours and minutes for readability purposes (rounding of seconds to the nearest minute). For all calculations the original measures in seconds have been used. Again the results will be presented in tables and figures. Correlations have been computed only as a way to find possibly interesting results and are reported in footnotes. For study B only the results for subjects B2 to B9 are reported, as subject B1 had severe technical problems (see § 7.3.5).

8.2.1 Help offered by the MASTER-TPD tool

From the log files the amount of time spent on different types of help offered by the MASTER-TPD tool can be derived: the log files contain time stamps for the opening and closing of each help window. It is not certain, though that subjects were always reading the help information when a help window was open. Inspection of the log files showed that subjects sometimes had several help windows open at the same time and sometimes left help windows open for a long time, occasionally until the end of the session. In other cases, subjects opened and closed the same help window several times within a relatively short period of time. Therefore, the following restrictions have been used in the computation of time spent on help:

A. Only one help window can be active. When another help window is opened the previous one is considered to be closed.
B. The maximum length of a help call is five minutes. After five minutes the help window is presumed to have disappeared to the background and to be no longer in use.
C. For the Manual, only the time spent on specific pages is counted, not the time spent on navigating between pages.

The MASTER-TPD tool offers different types of help (see § 5.2.2). The Manual contains all help information. Different parts of the information can also be accesses directly in the Training Cards, and under the buttons on the Notepad. To further analyse the use of help, three kinds of help information are defined:

- Help-Method: The time that subjects spent on information concerning the MASTER-TPD method and its steps which can be found in the Training Cards and in the Manual.
- Help-Tool: The time that subjects spent on information concerning the operation of the tool which can be found under the Tell-me-how button and in the Manual.
- Help-Guidelines: The guidelines that can be found under the Advice and Guidance buttons and in the Manual.

Table 8.1-A and Table 8.1-B show how the total time spent on help was divided over the three different kinds of information (i.e. adding up times of different ways of accessing the same information).

Table 8.1-A: Use of help study A

	Time spent on help	Ranking time on help [a]	Help: Method	Help: Tool	Help: Guidelines	Total time on task[b]
Subject A1	1:40	6	25%	40%	35%	39:38
Subject A2	8:09	10	45%	9%	46%	121:26
Subject A3[c]	1:51	7	32%	14%	54%	54:06
Subject A4	1:00	4	52%	19%	29%	50:12
Subject A5	4:15	9	40%	19%	42%	55:12
Subject A6	1:12	5	38%	35%	27%	99:53
Subject A7[c]	0:22	1	69%	23%	9%	12:31
Subject A8	3:44	8	28%	33%	39%	81:48
Subject A9	0:58	3	23%	8%	69%	44:45
Subject A10	0:47	2	49%	9%	41%	62:50
Average	*2:25*		*40%*	*21%*	*39%*	*62:14*

[a] Ranking from the highest (10) to the lowest (1).
[b] From Table 6.5.
[c] Subject A3 and Subject A7 performed worst and failed the course, see Section6.4.1.1.

Table 8.1-A gives the results for study A[19]. The subjects in study A spent somewhere between 20 minutes and 8 hours on help (average 2 hours and 25 minutes; standard deviation: 2 hours and 23 minutes). They looked at all three kinds of help information, spending least time on information about MASTER-TPD tool, and more time on information about the MASTER-TPD method and on the guidelines. Closer inspection of the subjects' notes showed that all subjects also regularly mention guidelines, which means that they have read and used them. The amount of time spent on information regarding the MASTER-TPD method may be somewhat exaggerated since the Training Cards that contain this kind of help pop up automatically when subjects move to a particular step.

[19] Please note that the restrictions used for the computations in this table were not used in the earlier report of the results of the study A in Verstegen et al. (2003).

This was the first time that the subjects used the MASTER-TPD method. They will probably spend less time on help regarding the method and its steps once they get more experienced and have worked with it for a longer period, e.g., during the whole simulator acquisition process or in more than one project. This is not expected to be true for the guidelines, since users will encounter different problems and bottlenecks at different moments and will, therefore, need different kinds of advice. The results for study B, which are presented in Table 8.1-B, support the conclusions:

Table 8.1-B: Use of help study B

	Time spent on help	Ranking time on help[a]	Help: Method	Help: Tool	Help: Guidelines	Total time on task[b]
Subject B2	4:09	7	48%	14%	39%	27:46
Subject B3	2:26	4	49%	23%	28%	24:39
Subject B4	1:42	2	37%	11%	52%	27:33
Subject B5	3:46	6	57%	6%	37%	25:43
Subject B6	1:29	1	42%	11%	47%	29:02
Subject B7	2:25	3	42%	20%	39%	30:15
Subject B8	3:37	5	59%	10%	31%	29:20
Subject B9	5:42	8	73%	6%	21%	29:23
Average	*3:10*		*51%*	*13%*	*37%*	*27:58*

[a] Ranking from the highest (8) to the lowest (1).
[b] All subjects spent more or less the same amount of time on the design task in this study, see Section7.4.1.3 for an explanation of the differences.

In study B subjects had less time to complete the same assignment (see § 7.3.2). They did, however, spend more time on help: roughly 1.5 to 5.5 hours (average: 3 hours and 10 minutes; standard deviation: 1 hours and 25 minutes). In study B subjects spent most time on information about the MASTER-TPD method, followed by the guidelines. Also in this study, subjects spent least time on information about MASTER-TPD tool.

Closer inspection of the log files of both studies showed that subjects who had access to the help information through buttons on the Notepad used the Manual very little or not at all[20]. This is not a problem since the Manual contains no information that cannot be found in the Training Cards or under the buttons Tell-me-how, Advice and Guidance. Apparently, users prefer the faster direct access with buttons, but they are willing to look for the information in the Manual if buttons are not available.

During both studies the general impression was that subjects found the assignment so difficult that they needed all available information to complete it. Tables 8.1-A and 8.1-B show that there is much variation in the amount of time subjects spent on help. Since there were no indications of technical problems accessing the help information, the hypothesis is that all subjects spent as much time on help as they needed, and that the time differences are caused by individual differences in the amount of time needed to (re)read the help information.

Figure 8.1 shows that there is no relationship between the amount of time spent on help and the quality of the resulting training programme designs[21].

[20] In study A three subjects (i.e. Subjects A8-A10) used a slightly different version of the MASTER-TPD tool with a different interface (see § 5.2.2). They could access all help information only through the Manual (i.e. Subjects A8-A10). Obviously, these subjects used the Manual a lot. The other subjects who could get to the information through buttons on the Notepads hardly used the Manual at all. There are no significant or meaningful differences in the amount of time spent on help or the time spent on different kinds of help information between the two groups of subjects (see Table 8.1-A).

[21] For study A: SRCC (Spearman Rho Correlation Coefficient) Time on help/Average quality of design = 0.18. For study B: SRCC Time on help/Average quality of design = -0.50.

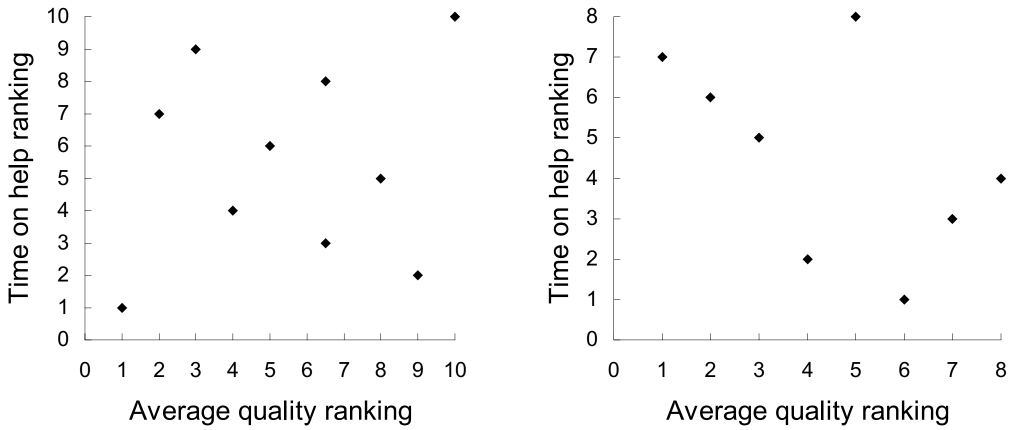


Figure 8.1: Time spent on help (ranking) versus the quality of the resulting training programme designs (average quality ranking):
- on the left in study A (data from Table 8.1-A and Table 6.3), and
- on the right in study B (data from Table 8.1-B and Table 7.3).

Some of the guidelines encourage the users to iterate. Figure 8.2, however, shows that there is no relationship between the number of iterations and the amount of time spent on guidelines[22] (nor for that matter with the total amount of time spent on help[23]). This is not strange if the time differences reflect individual differences in the required amount of help. Then all subjects would have just spent as much time on guidelines as they thought they needed. In study B most subjects claim that the guidelines did not make them iterate more (see § 7.3.2.1), but this is hard to check. The reasons for iteration that they mention are also included in the guidelines. Furthermore, Table 8.1-A and Table 8.1-B show that all subjects have looked at guidelines. Thus, the guidelines may still have helped subjects to decide when and how to iterate.

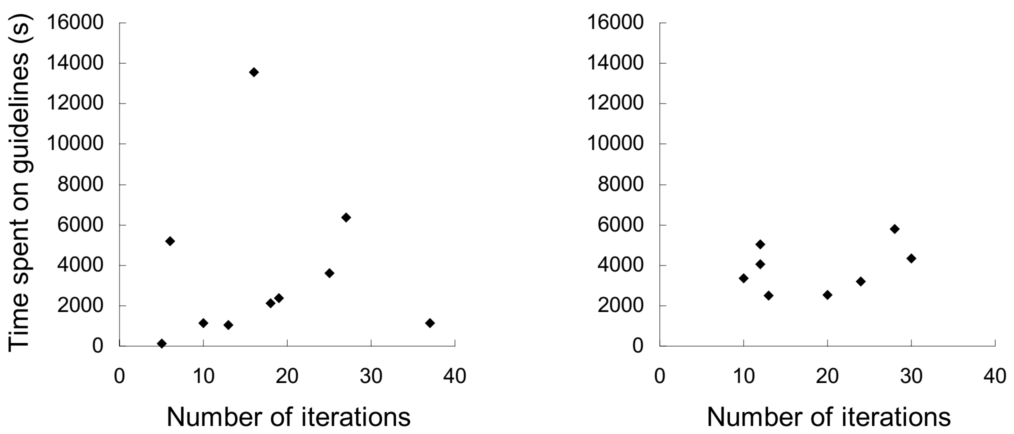


Figure 8.2: Time spent on guidelines (in seconds) versus the number of iterations:
- on the left in study A (data from Table 8.1-A and Table 6.7), and
- on the right in study B (data from Table 8.1-B and Table 7.6).

Limitations regarding the data on the use of help

There are some aspects that may have affected the reliability of these data. Even with the restrictions used, it is not certain that subjects were actually reading the help information when they had help windows open. If some subjects consistently closed help windows immediately after reading the information, and others did not, this will have caused differences in

[22] For study A: PPMCC (Pearson Product-Moment Correlation Coefficient) Time on guidelines/Number of iterations = 0.05. For study B: PPMCC Time on guidelines/Number of iterations = 0.32.

[23] For study A: PPMCC Time on help/Number of iterations = -0.02 For study B: PPMCC Time on help/Number of iterations = 0.40

the time measures presented in Tables 8.1-A and 8.1-B. Furthermore, in study A it was impossible to prevent subjects from printing help texts. It is not likely that this happened, though. The MASTER-TPD tool does not have printing options (but copy and paste is possible), and none of the subjects indicated that they had printed help texts when asked in the fourth questionnaire (Note that some subjects did say that they had printed out the training goals, intermediate versions of their designs, their own notes and the discussions from the web-based discussion lists). Finally, it is possible that these results are influenced by the limited number of subjects and the relatively limited time scale. The use of help may change when subjects get more experience with the MASTER-TPD method and tool (see above).

8.2.2 Notes

During both evaluation studies the subjects made notes on the Notepads with each step. Two TNO staff members counted the number of 'meaningful' fragments, i.e. sentences or parts of sentences. Notes are considered 'meaningful' if they describe the way the method was executed, or certain aspects of the design itself, e.g., reasons for decisions, explanations, doubts, possible alternatives, etc. Phrases that are considered 'not meaningful' are, for example, comments on the software or the organisation of the course, and explanations about the domain.

The same two raters analysed the number of notes for both studies. They were involved in the analysis of other data. The procedure for determining the number of notes was as follows:

1. The notes accompanying the final versions of the subjects' training programme designs were printed anonymously, i.e. without the subjects' names.
2. The two raters counted the number of meaningful fragments individually, with the instructions described above.
3. The two raters compared their individual scores, and discussed situations where they disagreed about what should be counted as one phrase or more than phrase. Small differences remained, due to differences in opinion about which phrases should be considered meaningful[24].
4. The average amount of meaningful phrases was computed. Based on this score the subjects were ranked from highest to the lowest amount of meaningful phrases. Where the averages were equal, subjects were given an equal intermediate ranking.

Tables 8.2-A and 8.2-B give an overview of the results:

Table 8.2-A: Notes study A

	Average meaningful phrases	Ranking number of notes[a]
Subject A1	198	9
Subject A2	632	10
Subject A3	156	5.5
Subject A4	178.5	7.5
Subject A5	65.5	1
Subject A6	178.5	7.5
Subject A7	99.5	3
Subject A8	127	4
Subject A9	156	5.5
Subject A10[b]	67.5	2

Table 8.2-B: Notes study B

	Average meaningful phrases	Ranking number of notes[a]
Subject B2	182	4
Subject B3	190	5
Subject B4	111	1
Subject B5	149	3
Subject B6	204	7
Subject B7	196.5	6
Subject B8	117.5	2
Subject B9	333	8

[a] Ranking from highest to the lowest, i.e. for study A from (10) to (1) and for study B from (8) to (1).
[b] Subject A10 was not a native Dutch speaker; she had some difficulty expressing herself in (written) Dutch, and this may have influenced the number of notes she made.

[24] For study A: PPMCC Number of notes Rater 1/Number of notes Rater 2 = 0.997. For study B: PPMCC Number of notes Rater 1/Number of notes Rater 2 = 0.94.

A relationship between the amount of meaningful notes and the judgements about the quality of the training programme designs would be not surprising, because these judgements were influenced by how well subjects explained and defended their design decisions (see § 6.4.1.1). Table 8.2-A shows, for example, that Subject A2, who was the best performer, also made three times as much notes as anybody else. There is no linear relationship[25], but in study A an upward trend can be discerned when Subject A10 (average quality ranking 9), who performed quite well but might have made less notes because she is not a native Dutch speaker, is left out (see Figure 8.3, left hand side). In study B the four subjects who performed better made more notes than the four subjects who performed less good (see Figure 8.3, right hand side).

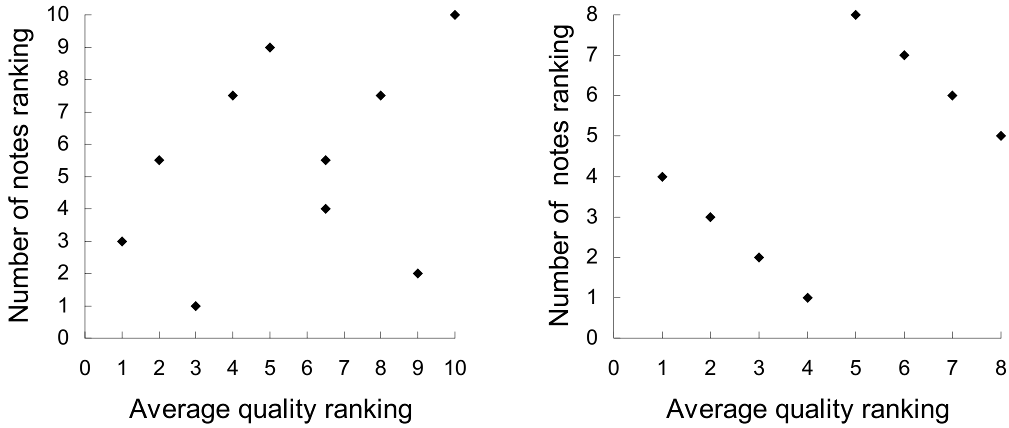


Figure 8.3: Number of notes (ranking) versus the quality of the resulting training programme designs (average quality ranking):
- on the left in study A (data from Table 8.2-A and Table 6.3), and
- on the right in study B (data from Table 8.2-B and Table 7.3).

In fact, in study B the relation between the number of notes and the quality of the subjects' designs follows the same upward trend as in the study A for two of the three raters (Rater 2 and 3), as depicted in Figure 8.4. The strange looking division in two groups seems to be mostly caused by the quality rankings of Rater 1. Therefore, the conclusion is that, in general, better performers seem to make more notes.

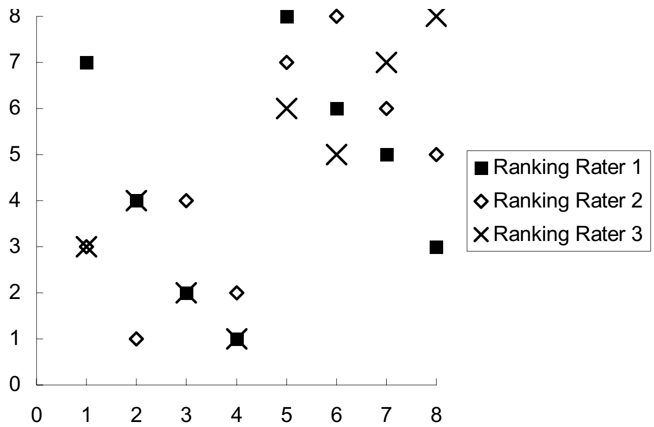


Figure 8.4: For study B: Number of notes (ranking) versus the quality of the resulting training programme designs as ranked by three different raters (data from Table 8.2-B and Table 7.3).

[25] For study A: SRCC Ranking number of notes/Average quality ranking = 0.36. For study B: SRCC Ranking number of notes/Average quality ranking = 0.52.

The results of study A do not indicate a relationship between the time spent on the design task and the number of notes for the other subjects[26]. For study B such a relationship was not expected since all subjects spent more or less the same amount of time on the design task (see § 7.3.2)[27].

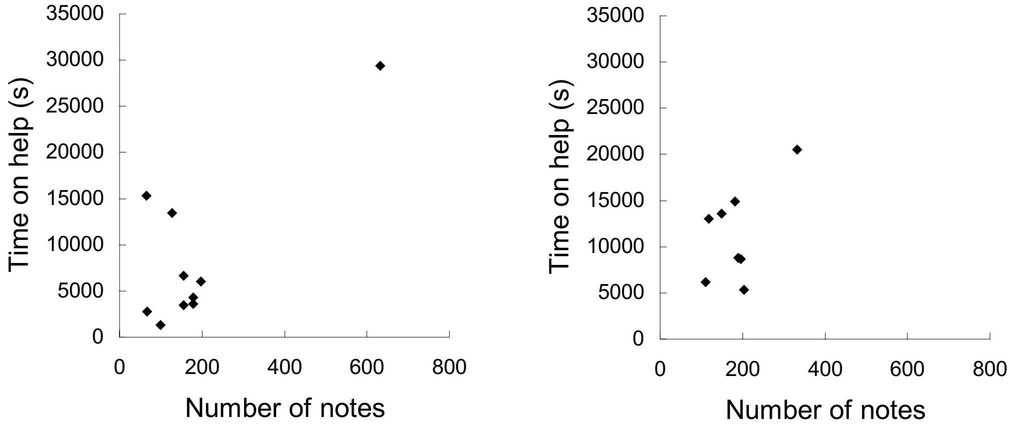


Figure 8.5: Time spent on help (in seconds) versus the number of notes:
- on the left in study A (data from Table 8.2-A and Table 8.1-A), and
- on the right in study B (data from Table 8.2-B and Table 8.1-B).

Different types of help advise the users to make notes. Figure 8.5 shows, however, that there is no relationship between the time spent on help and the number of notes that subjects make[28]: The analysis of the relationship between the amount of time spent on guidelines and the number of notes gives a similar result[29]. The process-oriented guidelines and the Hints on the Notepads give the most explicit advice, indicating also what topics should be addressed in the notes. Closer inspection, however, revealed no relationship between the amount of time spent on the process-oriented guidelines under the Guidance button and the number of notes[30]. The fact that there is no clear relationship with the amount of time spent on different help forms can mean a number of things. It is possible that subjects were already motivated to make notes or that the data on the time spent on help were not reliable enough to show any effect (see § 8.2.1). Probably, however, all subjects got enough information about when and how to make notes from the general advice to make notes in the descriptions of the MASTER-TPD steps, from the more explicit statements in the Hints on the Notepads and from whatever amount of time they spent on reading the process-oriented guidelines.

8.2.3 Questions to experts

During both evaluation studies the subjects could send e-mails to ask different kinds of experts for advice. Some e-mails concern more than one issue. Two TNO staff members (the author of this thesis and a colleague with experience in using the MASTER-TPD method and tool) indicated on prints of the e-mails what they considered separate issues, and categorised

[26] For study A: PPMCC Number of notes/Time on task = 0.66. This rather high correlation is caused by one subject, see (Verstegen et al., 2003).

[27] For study B: PPMCC Number of notes/Time on task = 0.27.

[28] For study A: PPMCC Time on help/Number of notes = 0.77 (significant). For study B: PPMCC Time on help/Number of notes = 0.56. Figure 8.5 shows that these high correlations are both caused by extreme scores of one subject.

[29] For study A PPMCC Time on guidelines/Number of notes = 0.79 (significant). However, this high correlation is due to the extreme scores of one subject. For study B: PPMCC Time on guidelines/Number of notes = 0.04.

[30] This comparison is not possible for three subjects in the study A (i.e. Subjects A8-A10) who used a slightly different version of the tool (see § 5.2.2).

them according to the coding scheme discussed below. Then the two raters got together to compare their results and to come to an agreement[31]. Tables 8.3-A and 8.3-B give an overview of the results.

Table 8.3-A: Questions study A

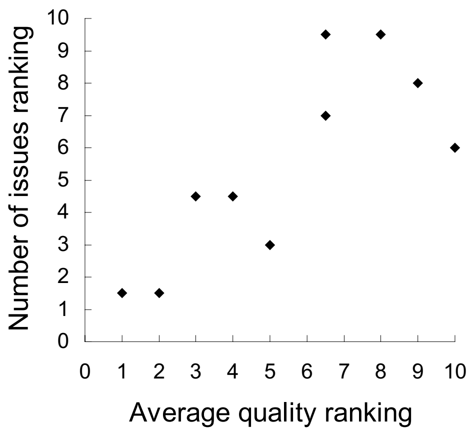
	Number of issues	Ranking number of issues[a]
Subject A1	2	3
Subject A2	11	6
Subject A3	0	1.5
Subject A4	8	4.5
Subject A5	8	4.5
Subject A6	37	9.5
Subject A7	0	1.5
Subject A8	37	9.5
Subject A9	13	7
Subject A10	21	8
Total	*137*	
Average	*14*	

Table 8.3-B: Questions study B

	Number of issues	Ranking number of issues[a]
Subject B2	22	7
Subject B3	5	1
Subject B4	7	2
Subject B5	21	6
Subject B6	25	8
Subject B7	14	4
Subject B8	12	3
Subject B9	18	5
Total	*124*	
Average	*16*	

[a] Rankings from highest to lowest number of issues addressed in e-mails to experts, i.e. for study A from (10) to (1), and for study B from (8) to (1).

During study A ten subjects sent a total of 62 e-mails to the experts that were available to answer their questions, addressing a total of 137 issues (average per subject: 13.7; standard deviation: 13.8). Surprisingly enough, the eight subjects in study B, who had less time for the assignment sent a total of 101 e-mails addressing a total of 124 issues, i.e. they asked more questions per person than the subjects in study A (average per subject: 15.5; standard deviation: 7.2). The subjects had been told beforehand that the experts would have to read and answer e-mails in between their other tasks. In both studies the fact that experts did not always answer immediately frustrated some of the subjects who claimed that they could not continue with their task without the answers to their questions. It is, however, a realistic situation that experts are not always available or not always able to provide the information that designers need (see § 3.2.1).



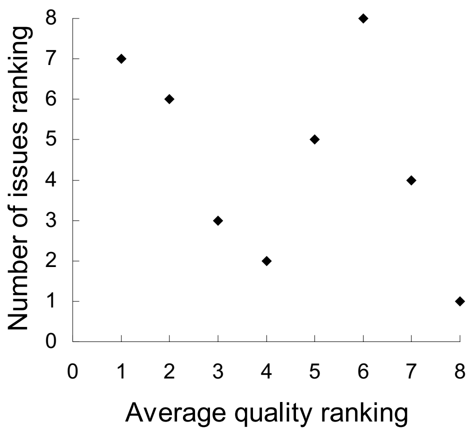


Figure 8.6: Number of issues addressed in e-mails to experts (ranking) versus the quality of the resulting training programme designs (average quality ranking):
- on the left in study A (data from Table 8.3-A and Table 6.3), and
- on the right in study B (data from Table 8.3-B and Table 7.3).

Figure 8.6 shows the relationship between the number of issues that subjects addressed in e-mails and the quality of their training programme designs[32]: In study A the five subjects

[31] Differences mainly concerned different interpretations of what should be seen as a separate issue and what not. On a few occasions the raters had interpreted the categories differently.

who performed better asked more questions to experts than the five subjects who performed less good and the two subjects who failed the course never asked any questions (see Table 8.3-A). This may be an indication that they did not work seriously on the assignment (especially since there was also a relationship between the amount of time spent on the design task and the quality of the resulting training programme designs in this study, see Figure 6.1). In study B there seems to be no relationship between the amount of questions asked to experts and the quality of the subjects' training programme designs. The hypothesis was that the differences in quality were relatively small in this study (see § 7.4.1.1). If this is true than it is not strange that relationships with other variables are less strong, or absent.

A part of the guidelines advises users to ask experts for advice on certain topics. Figure 8.7 seems to suggest that there was a relationship between the amount of time that subjects spent on the guidelines and the number of questions they ask in study B (right hand side), but not in study A (left hand side)[33]:

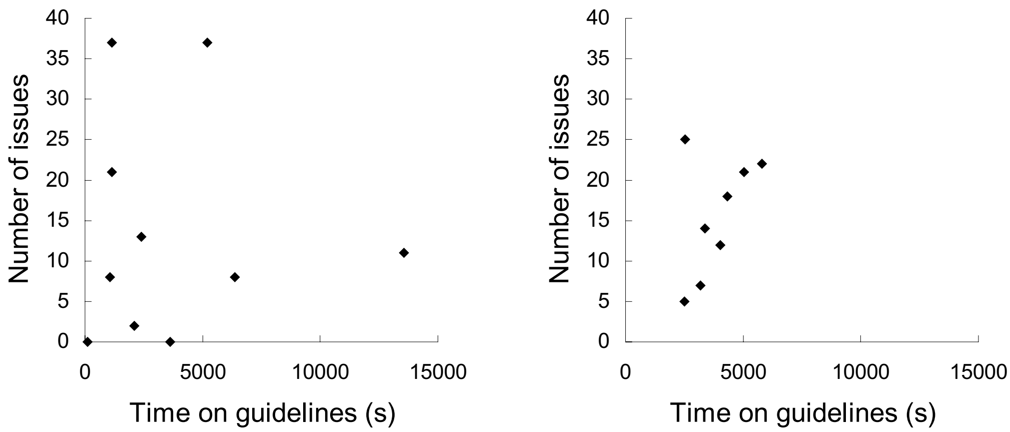


Figure 8.7: Number of issues addressed in e-mails to experts versus the time spent on guidelines (in seconds):
- on the left in study A (data from Table 8.3-A and Table 8.1-A), and
- on the right in study B (data from Table 8.3-B and Table 8.1-B).

Closer inspection of the results of study B, however, led to the conclusion that this is probably an artefact. The guidelines about experts are part of the process-oriented guidelines that can be found under the Guidance button, and in the Manual. Since the subjects hardly used the Manual, the relation should even be stronger for the time spent on information under the Guidance button and the number of questions that subjects ask. Figure 8.8 shows that this is not the case:

The guidelines about experts mainly address questions for experts regarding the domain, the MASTER-TPD method and instructional design. The absence of a relation between the amount of time spent on process-oriented guidelines and the number of questions might be explained by the fact that many of the subjects' questions concerned other issues, especially during study A (see Table 8.4).

---

32 For study A: SRCC Number of issues/Average quality of design = 0.78. For study B: SRCC Number of issues/Average quality of design = -0.41.

33 For study A: PPMCC Number of issues/Time on help = 0.06; PPMCC Number of issues/Time on guidelines = 0.00. For study B: PPMCC Number of issues/Time on help = 0.28; PPMCC Number of issues/Time on guidelines = 0.47.

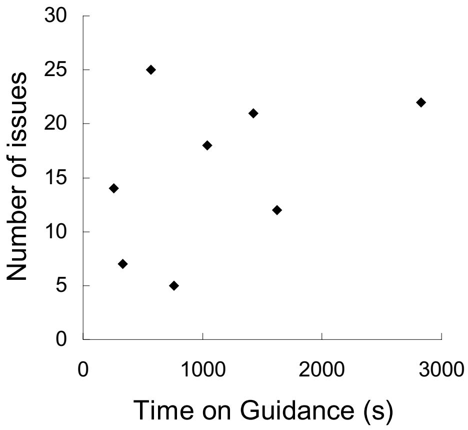


Figure 8.8: Number of issues addressed in e-mails to experts versus the time spent on the part of the guidelines that can be found under the Guidance button (in seconds) for study B.

To analyse the nature of the questions, the raters used a coding scheme with four main categories, with sub-categories:

- Organisation: All questions and comments regarding the organisation of the evaluation study, i.e. questions or comments regarding the time-schedule, meetings and requirements (Planning), questions about who to address a certain question to (Who?) and questions regarding feedback (Feedback). Note that only questions or comments concerning feedback in general have been scored here. Requests for feedback on a specific decision or work done in a specific step are scored for that step under Method.
- Method: Questions or comments regarding specific steps of the method.
- Tool: Questions or comments concerning the tool, i.e. technical problems such as error messages or crashes (Technical), issues concerning the operation of the tool in the different steps (Related to steps) or issues regarding the use of the different forms of support (Related to support).
- General: Questions or comments concerning the MASTER-TPD method or tool in general, or questions concerning background theory.
- Domain: Questions or comments concerning the domain that was used (i.e. the analysis of UAV thermal images).

Table 8.4 gives an overview of the results:

Table 8.4: Type of issues in e-mails to experts

Category	Sub-category	Study A		Study B	
Organisation	Planning	23	17%	9	7%
	Who?	7	5%	3	2%
	Feedback	10	7%	0	-
Method	Step 1: Design of Training	3	2%	3	2%
	Step 2: Specification of Training Activities	4	3%	9	7%
	Step 3: Construction of Simulator Course Map	1	1%	0	-
	Step 4: Planning of Assessment	16	12%	9	7%
	Step 5: Design of Scenarios	9	7%	10	8%
	Step 6: Design of Instruction	1	1%	1	1%
Tool	Technical	27	20%	13	10%
	Related to steps	18	13%	32	26%
	Related to support	7	5%	1	1%
General	About method or tool in general	4	3%	6	5%
Domain		7	5%	28	23%
Total		*137*		*124*	

About 25% of the questions concern the MASTER-TPD method, mainly Step 4 and Step 5, and in study B also Step 2 (for discussion about the difficulty of different steps, see § 8.3.2). Table 8.4 shows that in both studies more than a third of the questions concerned the

MASTER-TPD tool. In study A half of these questions concern technical problems. During study B this type of question did not always have to be asked through e-mail because supervisors were present to help with urgent technical problems. In both studies subjects also asked many questions about using the MASTER-TPD tool to execute specific MASTER-TPD steps.

In study A almost a third of the questions were of organisational nature, e.g., questions about meetings, scheduling problems, etc. It is not strange that this type of question was not very common in the controlled setting of study B. It is surprising, though, that none of the subjects during study B asked for feedback (see § 7.4.1.5 for discussion of this difference between the studies).

Another striking difference is the number of questions about the domain, i.e. UAVs and thermal image cameras (see § 6.2). In both studies the subjects had no prior knowledge about the domain and this knowledge was expected to be important (see § 3.2.2.2). In study A there were surprisingly few questions about the domain, whereas in study B about 25% of the questions concerned the domain. This is probably caused by differences in the design of the two studies:
- During study A the subjects got more domain information during face-to-face meetings, including demonstrations of different simulators and a thermal image camera.
- The reader concerns information regarding the domain, and although it was provided to subjects in both studies there was not much time available to read background information during study B.

8.2.4 Subjects' opinion about different forms of support

Information regarding the subjects' opinion about the different forms of support is available from the fourth questionnaire in study A and the second questionnaire in study B. These questionnaires have been returned anonymously. For study A only the results of the seven subjects using the same version of the MASTER-TPD tool (i.e. Subjects A1-A7) are reported here, since access to help information was different in the version used by the other three subjects (see § 5.2.2). During study B the questionnaire was only returned by seven out of eight subjects. This results in data regarding seven subjects for each study; the mean scores and standard deviations (SD) are given in Table 8.5. A more elaborate analysis of the subjects' experiences with the MASTER-TPD tool can be found in Steutel (1999) for study A and Zagers (2001) for study B.

Table 8.5: Subjects' opinion about forms of support[a]

	Mean score[b] study A	SD study A	Mean score[b] study B[34]	SD study B
Advice	4.1	0.5	4.7	0.5
Guidance	3.4	0.7	4.3	0.8
Tell-me-how	4.3	0.7	3.9	1.4
Training Card	4.0	0.7	3.9	1.1
Manual	3.7	0.6	3.1	0.7
Hints on Notepads	3.4	0.9	[c]	[c]
Experts	3.0	1.3	4.4	0.5

[a] Results for seven subjects in study A, and seven subjects in study B.
[b] Answers on a five-point scale from very bad (1) to very good (5).
[c] During study B the questionnaire did not contain a question about the Hints.

Table 8.5 shows that subjects were quite positive about all forms of support (all scores more than three on a five-point scale). The subjects in study A are least enthusiastic about the help they got from the experts. The subjects in study B are least enthusiastic about the Manual. In

[34] In the fourth questionnaire in study A, a 7 point scale was used; in the second questionnaire in study B a 5 point scale was used. To make it possible to compare the results the mean scores of study A have been recomputed to a 5 point scale.

study B subjects were also asked which forms of support they found the most and the least helpful. In response to this open question four out of these seven subjects mention the Manual as the least helpful for two reasons: there is no button to the Manual on the Notepad, and the Manual does not provide information that is not also included in one of the other types of help. Two subjects said that the Tell-me-how information was the least useful and one subject said the option to ask questions to experts was the least useful. Different forms of support were mentioned as the most useful: Advice (3 subjects), Tell-me-how (2 subjects), Training Cards (2 subjects) and Guidance (1 subject). When asked which forms of support they had missed three subjects mentioned more help or more direct help with regard to using the tool, one subject asked for feedback on completed steps, and one asked for examples.

8.2.5 Conclusions

In Section 8.2 the following research questions have been addressed:
- How much do subjects use the different types of help offered by the tool? Is the quality of their training programme designs related to the use of help? Is the number of iterations related to the use of help?
- Do subjects make notes? Is the quantity of notes related to the use of (different types of) help?
- Do subjects approach experts for support? If so, which kind of questions do they ask?
- What is the subjects' opinion about the different forms of support?

The general impression during both studies was that subjects found the assignment so difficult that they needed all available support to complete it. This is supported by the fact that subjects looked at all kinds of help information in the tool. The subjects are quite positive about all forms of support. There are no indications of problems with accessing help information in the tool, and there are no clear relations between the amount of time spent on help and the quality of the subjects' training programme designs, the number of iterations, and the number of notes subjects make. Therefore, the hypothesis is that differences in time spent on help are either caused by the way time was measured (see § 8.2.1) or, more probably, by differences between subjects in the amount of time they needed for (re)reading the help information.

Although taking notes is important in an iterative design process (see § 5.4.2), the expectation beforehand was that it would be hard to convince users to make notes. This proved to be not true during both evaluation studies: all subjects made notes, and often -but not always- those notes were sufficient to follow the subjects' reasoning. Note that it is possible that the subjects in these evaluation studies made more notes than real target users might make, because they are students who are used to writing assignments. There are indications that engineers, for example, have much more difficulty to make notes that are understandable for non-experts (Barnard and Boy, 2003). Target users might also be more reluctant to make notes because they can be held personally responsible if wrong decisions can be traced back to them. The study with real target users, reported in Section 9.2, was too short to evaluate this aspect, but the participants did say that making notes was, in their opinion, very important. In a real design process in a real environment, the users' notes will also be used by others, i.e. other members of the design team and/or people responsible for quality assurance. These people will no doubt ask for clarification if the notes are incomplete or unclear. They might even find the hints and guidelines that were implemented for the designers in the MASTER-TPD method useful to check whether the notes are complete.

In both studies the subjects asked questions to experts by e-mail. Some subjects were frustrated because the experts were not always able to answer immediately. This could be a sign that they find it hard to deal with uncertainty and incomplete information. There are other indications for this hypothesis: in study B some subjects were frustrated by the fact that not all domain information was provided at the very start (see § 7.4.3.2); and during study A subjects asked for constant feedback (see § 6.4.1.4). In practice, insecurity is expected to play a large role as well, since the target users are also novices and the consequences of bad design will be more serious than during these evaluation studies (see § 9.2). Since incomplete and insecure information and changing constraints are typical for this kind of design task (see § 3.2.1) it seems worthwhile to investigate ways to support users in dealing with uncertainty. One way could be to make assumptions explicit and to provide one, central location to store and review them (see, e.g., § 9.3.1). In any case, it is important that experts are easy to contact for questions, e.g. by phone or by e-mail, and that designers get new information as soon as it is available.

8.3 The MASTER-TPD method

In this section I address the following research questions:

§ 8.3.1 What is the subjects' opinion about the structured step-by-step approach of the MASTER-TPD method?

§ 8.3.2 Are subjects able to complete the steps of MASTER-TPD method? Which steps are difficult?

8.3.1 Subjects' opinion about the MASTER-TPD method

8.3.1.1 Opinions of the structured approach

Information regarding the subjects' opinion about the MASTER-TPD method is available from the third questionnaire in both studies (issued in both cases at the end of the design period, see Table 6.2 and Table 7.2). Table 8.6 shows that subjects think that the structured approach proposed by the MASTER-TPD method is useful. Most subjects also find it a pleasant way of working, except for two subjects from study A and one subject from study B (i.e. they answered 2 on a five-point scale). Subjects agree with the method's claims that using it will lead to better and better motivated specifications in a more efficient way, although some of them comment that it still depends on the user whether he or she executes the steps correctly and makes elaborate notes or not. Indeed the method offers advice and support and it guides users step-by-step through the design process and offers advice, but gives no guarantees.

Table 8.6: Subjects' opinion about MASTER-TPD approach

Question	Mean score study A	SD study A	Mean score study B	SD study B
The method proposes a structured way of working with pre-specified steps. Did you find this structured way of working: [a]				
# Not useful -------------- Useful[35]	4.5	0.5	4.0	0.5
# Not pleasant ----------- Pleasant	3.8	1.1	3.4	0.7
The MASTER method claims that using the method has the following advantages. Do you agree? [b]				
# Better simulator specifications	3.7	0.7	3.6	0.7
# Designed in a more efficient way	4.0	0.5	3.8	0.7
# Specifications better motivated	3.9	0.6	3.4	1.1

[a] Answers on a five-point scale from left (1) to right (5).
[b] Answers on a five-point scale from no, not at all (1) to yes, completely (5).

8.3.1.2 Strong and weak points

In both studies subjects were also asked to describe the strong points and the weak points of the method. During study A this open question was part of the second questionnaire issued

[35] In study A there is a significant difference between the subjects using slightly different versions of the tool ($t = -2.50$; $p < 0.05$). No conclusions are drawn because of the small number of subjects, and because the average scores for both groups are high (average 4.29 vs. average 5.00).

during the fourth meeting (see Table 6.2). During study B it was part of the third questionnaire issued at the end of the design period (see Table 7.2). Table 8.7-A and Table 8.7-B give an overview of what subjects mentioned as strong points in study A and B respectively:

Table 8.7-A: Strong points of MASTER-TPD method mentioned in study A

Strong points	Of the 13 subjects: [a]
Structured approach	12
Possibility to make revisions	1
Explicit decision what to train in simulator	1
Help offered by tool	7
Notepads	2
Theory integrated	1

Table 8.7-B: Strong points of MASTER-TPD method mentioned in study B

Strong points	Of the 8 subjects: [a]
Structured approach	7
Possibility to make revisions	3
Go more and more into detail	2
Suitable for beginners and experts	1
Concrete	1

[a] In study A, 13 subjects were still involved at this moment (three of them dropped out before the end of the study, see § 6.3.5); in study B there were 8 subjects.

In both studies almost all subjects mentioned the structured approach as a strong point of the MASTER-TPD method. They gave the following reasons: it takes you step-by-step to the goal, it defines what you should do and helps you to keep an overview, it makes sure you don't forget anything, and it forces you to think about important issues. The explicit decision of what should be trained with the simulator and what not is mentioned explicitly by one subject. On the other hand, some subjects (i.e. four in each study) also see limitations of the structured approach: there is less freedom for the designer, it limits creativity, there are many steps, you cannot continue with the next step when you get stuck in a given step, and when you change something in one step, you have to pay attention to the consequences for other steps (see Table 8.8-A and Table 8.8-B). During the presentations at the end of study A, one subject mentioned that he did not like having to work according to a prescribed method (N.B. this was an evening student, who worked as a primary school teacher). Another subject remarked that she liked the structure of the method, but she felt that it had sometimes hampered her creativity.

In study A the support that was offered was also often mentioned as a strong point: the help information in the MASTER-TPD tool (seven subjects) and the Notepads (two subjects). Especially in study B subjects seem to value the iterative approach: the possibility to make revisions, i.e. to go back and adapt the design (three subjects), and the fact that you work iteratively in more and more detail (two subjects). In study A only one subject mentioned this as an advantage of the MASTER-TPD method. A reason for this difference might be that the interventions in study B made clear why iteration is necessary. The subjects in study B were also less insecure and may have had a better overview of the reasons for iteration because they worked in a concentrated period of time.

In both studies subjects complained about the fact that training goals cannot be clustered in the tool. Inspection of the designs showed that all subjects in both studies found ways to cluster of training goals: some have added topics in the (linear) list in capital letters, the others describe the clusters in their notes. During discussions in one of the meetings in study A some subjects mentioned that they had printed the training goals to get a better overview, and had sorted them on paper. In fact, that training goals and training activities could not be clustered was a technical problem with the MASTER-TPD prototype tool, and not an intentional limitation. Especially in an iterative design process clustering is a powerful mechanism. Users could, for example, put clusters of training goals in a sequence in the global design phase and not bother with the sequence of individual training goals until they start working on a more detailed design; or define training activities for entire clusters first and only go into more detail for those clusters that will be (partly) covered with the simulator. During study A, these questions were part of the second questionnaire issued half way the design period (during the fourth meeting). This may have been a reason for the subjects' problems with the terminology and descriptions. According to five subjects it was sometimes not clear what should be done in a specific step or how detailed descriptions need to be. Three subjects also commented on the content or structure of the help forms. One subject complained about the lack of an elaborated example. Another subject complained about the quantity of information and the fact that some things are mentioned in more than one place. A third subject said that he would rather have all information in one text document instead of under different buttons in the tool (Note that this subject seems to have missed the fact that there is also a Manual with all help information available. Moreover, it is clear from the use of help forms in Tables 8.1-A and B and the opinions about forms of support, reported in § 8.2.4, that most subjects prefer the help information directly accessible under buttons). Three subjects that participated in study A also complained about technical problems, such as copy and paste functions and the location of the database. One subject missed an overview of the whole training programme in the later steps where the Course Map only shows that part of training that will take place in the future simulator.

Table 8.8-A: Weak points of MASTER-TPD method mentioned in study A

Weak points	Of the 13 subjects: [a]
Limitations structured approach	4
Not cluster training goals/training activities	3
Structure/content of help	3
No overview of the whole course	1
Terminology and description not always clear	5
Technical problems	3

Table 8.8-B: Weak points of MASTER-TPD method mentioned in study B

Weak points	Of the 8 subjects: [a]
Limitations structured approach	4
Not cluster training goals/training activities	3
Order of Step 4 and 5	2
Background in instructional design and development required	1
Link theory/practice not clear	1
Resources not taken into account	1
Simulator training not in instructional context	1

[a] In study A, 13 subjects were still involved at this moment (three of them dropped out before the end of the study, see § 6.3.5); for study B there were 8 subjects.

During study B the subjects had more comments on the MASTER-TPD method itself: one subject thought that simulator training should not be designed separately from the rest of the training. Another subject complained about the fact that the method does not explicitly take resources into account. And two subjects indicated that Step 5 (Design of Scenarios) should take place before Step 4 (Planning of Assessment), because you can plan assessment better when you have elaborated the lesson plan and the exercises (i.e. the training scenarios).

Finally, there are some contradictions: one subject thought that the method would be suitable for beginners and experts. Another subject, however, was adamant in saying that users of the MASTER-TPD method need a background in instructional design and development theory. Likewise, one subject had problems understanding the link between theory and practice and the way that the method would be applied in practice. On the other hand, one of the other subjects said that the method was very concrete and applicable and another subject praised the integration of theory and practice.

8.3.1.3 Discussion

The MASTER-TPD method prescribes a structured design approach. Since there are strong input-output relationships between steps, it is almost impossible for users to skip steps or to do them in a different order. Beforehand, some frustration about this enforced step-by-step approach was expected. However, almost all the subjects are positive about the structured approach of the MASTER-TPD method (see § 8.3.1.1), although some of them also see limitations. They think that such an approach hampers creativity. This is illustrated by the following comment in the questionnaire from one of the subjects in study B (translated from Dutch): "Your design remains very structured. You are limited severely in your creative thinking process, though. At the start you are full of nice ideas that you cannot use because you have to stick to the strict structure." This subject's problem might be solved by facilitating brainstorming at the very beginning of the design process in order to lay down creative ideas before they get pushed to the background by a strict method or other demanding tasks.

From a more fundamental point of view, however, the question is which role creativity plays in the design process. The ultimate goal of the MASTER-TPD method is to support the design of good specifications for training simulators and good training programmes. The results of the field study in Chapter 2 indicated that important instructional aspects are currently often forgotten. A structured approach helps to keep the design process efficient and to ensure that all the important aspects are addressed. The descriptive literature in Chapter 3 indicates that a structured approach may be particularly useful for novice designers. On the other hand, it is clear that creativity plays a role in design processes. One of the raters in study B commented on the fact that the subjects all came up with very standard 'boring' training programme (see § 7.4.1.1). The most important causes are probably the limited time available for the design task and the limited experience of the subjects. However, it seems worthwhile to investigate ways to encourage creativity, while at the same time using a systematic approach. I will come back to this issue in Section 10.5.

8.3.2 The MASTER-TPD steps

In this section I will discuss the steps of the MASTER-TPD method in more detail. First I present the different kinds of data that provide information about the difficulty of specific steps in Section 8.3.2.1. Subsequently, in Section 8.3.2.2, I discuss why some steps may have been more difficult for the subjects.

8.3.2.1 Data regarding MASTER-TPD steps

Information about the difficulty of individual MASTER-TPD steps can be derived from three types of data:
- the questions asked to experts in both studies;
- answers to questionnaires in both studies; and
- the feedback provided during study A.

Questions to experts regarding the MASTER-TPD method
Table 8.4 in Section 8.2.3 shows that most of the questions about the MASTER-TPD method concern:
- Step 2: Specification of Training Activities (more clearly in study B),
- Step 4: Planning of Assessment (both studies), and
- Step 5: Design of Scenarios (both studies).

All the questions about Step 2 in study B concern Step 2.1, more specifically what is expected from users in this step and the level of detail at which training activities should be specified. Closer inspection of the questions about Step 4 shows that in study A many questions concern working with the Course Map (with which subjects are confronted for the first time in Step 4.1) and the specification of the end criterion. In study B half of the questions concern terms that are used in the library of Step 4.2 (i.e. examples of goals and types of assessment). In both studies the questions about Step 5 concern the goal of this step, the required level of detail, and the format that is used to describe scenarios (i.e. both Step 5.1 and Step 5.2).

Questionnaires

In study A subjects gave an opinion about the difficulty of the MASTER-TPD steps in the third questionnaire issued during the fifth meeting at the end of the nine-week design period (see Table 6.2):

Table 8.9-A: Difficulty of MASTER-TPD steps study A[a]

Step	1.1	1.2	1.3[36]	2.1	2.2	3.1	4.1	4.2	5.1	5.2	6.1	6.2	6.3
Mean	3.5	2.3	3.5	2.7	3.6	4.2	2.3	3.2	2.3	2.2	3.3	3.1	3.1
SD	0.7	0.7	0.9	1.0	0.8	0.8	1.1	0.6	0.7	1.0	1.2	1.1	1.2

[a] Answers on a five-point scale, question: I think that this step is difficult (1) -- easy (5).

During study B the question was phrased differently. Subjects were asked to list the three easiest and the three most difficult steps, and the three most and the three least meaningful steps in the third questionnaire issued at the end of the design period (see Table 7.2). Table 8.9-B shows no unambiguous relationship between difficulty and meaningfulness. Step 2.1 that is considered very meaningful by six subjects is also considered difficult by four subjects. However for other difficult steps (e.g., Step 1.2 and Step 5.2) there is no high score on meaningfulness, and Step 2.2 that is considered very meaningful by all but one subject, is considered to be difficult by two subjects and easy by two others.

Table 8.9-B: Difficulty of MASTER-TPD steps study B[a]

Step	1.1	1.2	1.3	2.1	2.2	3.1	4.1	4.2	5.1	5.2	6.1	6.2	6.3
Most difficult		5		4	2		1	1	3	5	1	2	2
Easiest	6	2	1		2	4	4		1		3	1	
Most meaningful	1	3	1	6	7			2	2	2			
Least meaningful[b]		2	4			2	1	1			1	1	2

[a] Numbers indicate the number of subjects that mentioned this step (N.B.: not all subjects always list three steps).
[b] Three subjects said that all steps are meaningful and important.

Summarising the results of the questionnaires from both evaluation studies (represented in Table 8.9-A and Table 8.9-B) the subjects indicate that the following steps are difficult:

- Step 1.2: Determine Training Sequence (although also considered easy by two subjects in study B, see Table 8.9-B).
- Step 2.1: Define Training Activities.
- Step 4.1: Make Assessment Points (only clear in study A; in study B half of the subjects considered this step to be easy, see Table 8.9-B).
- Step 5: Design of Scenarios (both Step 5.1 and Step 5.2).

In both studies the subjects thought that the most difficult steps were also the most time-consuming (see Verstegen et al., 2003, and Verstegen et al., in press, respectively).

Feedback provided during study A

As explained in Section 6.3.6.3 the subjects in study A received individual feedback twice: after sending in their global training programme design and after sending in their detailed training programme design. The feedback focused on the subjects' execution of the MASTER-TPD method, and did not interfere with the subjects' design styles, design choices or reasoning. When a topic was addressed in the feedback to more than one subject, standard phrases were used. Table 8.10 gives an overview of the topics addressed in the feedback[37]:

[36] In this first study the three subjects using the slightly version of the tool find step 1.3 significantly less easy (average on a five-point scale 2.7 vs. 3.7; $t = 2.60$; $p < 0.05$). They are also significantly stronger in their opinion that executing this step will take a long time (average on a five-point scale 4.3 vs. 3.3; $t = -2.97$; $p < 0.05$) and have a significantly lower opinion of the help information with step 1.3 (average on a five-point scale 3.0 vs. 4.0; $t = 2.90$; $p < 0.05$).

[37] In both feedback rounds the feedback was sent by e-mail and started with a general part with some announcements and an encouraging statement that they were on the right track.

Table 8.10: Topics addressed in feedback study A[a]

Topics	First feedback	Second feedback
Step 1: Specify range of instructional products for all training goals	1	-
Step 2: More specific definition of training activities	7	-
Split-up training activities	2	3
You can specify > 1 training activity for a training goal	3	-
Try to specify training activities independent of instructional product	1	1
Step 3: Update Course Map after changes in Step 1 and 2	1	-
Step 5: Clarification of the term scenario and the goal of Step 5	3	3
More specific description of scenarios	-	6
Split up scenarios	-	2
Step 6: Specify instructional interventions for each scenario	1	-
Step 4, 5 and 6 concern only training covered with simulator	1	-
Tip: you can ask experts for advice	1	-
Advice to take more notes	2	1

[a] Numbers of subjects that received this feedback (out of 10 subjects).

The most important topic in the first feedback round was the definition of training activities in Step 2.1. Many subjects had formulated training activities in very general terms, such as 'study And practise' or 'perform real task'. Some subjects did not specify training activities for each training goal, but used one general definition for a whole group. This lack of detail makes it difficult to decide which training activities should be covered in the future simulator and which not. These subjects were advised to specify training goals on a more detailed level and/or to split them up into more than one training activity. An example from their own work was used, such as "…for example: practise how to focus the camera, instead of practise real task" or "for example: study how you can steer the camera with a joystick + practise steering the camera with the joystick, instead of just study and practise". If subjects had never specified more than one training activity for one training goal, this possibility was suggested to them. The training activities were also the main issue during the third non-obligatory meeting at the end of the third week of study A at which half of the subjects were present. We discussed what training activities are and what their function is in the design process. The subjects stated that the feedback that had been provided had helped them to get a clearer view on this issue.

Some of the topics from the first feedback round were addressed again during the second round. The most important topic in this round, however, was the design of training scenarios in Step 5. Often, the descriptions were not detailed enough; i.e. they did not contain enough information to specify the future training simulator. Two subjects tried to specify training scenarios that covered a whole range of training activities. Since the descriptions of these scenarios were quite long and still very abstract (in the context of training simulator specification), these subjects were advised to split them up and to define one or more scenarios for each training activity.

8.3.2.2 Difficult steps in the MASTER-TPD method

In summary, the data presented in Section 8.3.2.1 indicate that the most difficult steps of the MASTER-TPD method are:
- Step 1.2: Determine Training Sequence (only in questionnaires).
- Step 2.1: Define Training Activities (in questions, questionnaires and feedback).
- Step 4: Planning of Assessment (only in questions to experts).
- Step 5: Design of Scenarios (in questions, questionnaires and feedback).

Possible reasons are discussed below.

Step 1.2: Determine Training Sequence
Problems with this step only show up in the questionnaires and are probably largely caused by problems with the tool. Subjects wanted to group training goals and organise them in a hierarchy, but technical limitations of the tool make that impossible: training goals can only be represented in a flat list (see also the discussion about weak points in § 8.3.1.2).

Step 1.2 is also the first step where users are confronted with the details of the domain, because in Step 1.1 they select main training strategies and/or learning principles for the whole set of training goals. Therefore, it is also possible that sequencing training goals was hard because it was difficult for the subjects to understand the content, difficulty and criticality of the training goals of the new domain.

Step 2.1: Define Training Activities
Difficulties with the definition of training activities in Step 2.1 appear in all three kinds of data. The feedback provided during study A shows that one of the causes of these problems is that training activities are defined in general terms, too abstract to be useful as input for the next steps. In study B one of the subjects said that she had initially misunderstood the goal of Step 2.1 and had only understood it when she got to Step 4 (information from comments in the third questionnaire). One of the reasons may be that the guidelines for Step 2.1 indicate that training activities can be defined at a more abstract level during the first, global design phase. Subjects may not have realised that the description can also be too abstract.

Defining training activities is, however, a difficult task for more fundamental reasons as well. Firstly, it requires a change of perspective: from the training goals that describe what should be taught (instructor perspective) to how it is going to be learned (trainee perspective), i.e. which activities are trainees going to execute in order to achieve those training goals. Secondly, in this step users have to make the step from (instructional design) theory to the elaboration of a concrete training programme. Thirdly, the MASTER-TPD method prescribes that training activities should be described independently of instructional products. The idea behind this is that in most cases the same training activity can be executed with different instructional products, e.g., studying some piece of background theory can happen in the classroom or with CBT, and practising a specific task might be possible with a simulator and during field training with the operational system. Defining training activities independently from instructional products makes it possible to come back to this step when the designer's first choice is for some reason not feasible, in order to see whether the same activity can be executed with another type of instructional product. It is, however, quite difficult to define training activities independently of the instructional products, not only for the novice designers during these evaluation studies (as shown by the feedback rounds in study A, and the questions asked by subjects in both studies) but also, based on comments from TNO colleagues, for experienced instructional designers.

I have observed similar problems concerning the definition of training activities during contacts with domain experts and instructors in the context of other projects. They can define training goals, but when they are asked what trainees need to do to achieve those training goals, the answer is often something like: "Well, they have to practise the tasks". However, training activities do not always have to be identical to the real tasks (Farmer, Jorna, Riemersma, van Rooij and Moraal, 1999): different aspects can be trained separately or combined, in simplified or enhanced settings or even in an environment that is totally different from the real task environment. The quick detection of targets, for example, can be trained separately from the identification of targets and firing weapons, using a simple construction with lights that can flash on at different places in the environment.

Step 4: Planning of Assessment
It is possible that subjects had problems with the task of planning assessment in a simulator, which was new to them. It is more probable, though, that the problems with this step are for

an important part caused by technical limitations of the prototype tool. In Step 4.1 (Make Assessment points) users are confronted for the first time with the Course Map. The Course Map does not provide an overview of the whole training programme, but only of the part that will be covered with the simulator. One subject explicitly mentioned this as a weak point of the MASTER-TPD method (see § 8.3.1.2). Another problem is that the Course Map does not show where the training activities come from, i.e. it does not show the training goals that the training activities were specified for. Inspection of the designs showed that in study A half of the subjects kept on including non-simulator parts in their notes or in the descriptions of assessments and scenarios, e.g., describing scenarios that include instruction in the class room and studying background theory in syllabi as well as practising tasks with the simulator. In study B one subject did the same. This behaviour is not strange. It is good that designers strive for an integrated set of training activities with different instructional products to cover all the training goals (cf. integration of simulator training in the course, see § 2.4.3) and not surprising that they need an overview of the whole training programme to do so. The fact that the Course Map did not provide such an overview (e.g., with the non-simulator parts greyed out) was actually not an intended limitation, but a technical problem with the implementation of the MASTER-TPD tool.

In study A the questions that subjects asked by e-mail also concern the end criterion. The Manual and the Training Cards urge users to specify an end criterion but the tool offers no specific place for this (although it can, of course, be planned as a regular assessment point). In study B half of the questions concern terms that are used in the library of Step 4.2 (i.e. examples of goals and types of assessment).

Step 5: Design of Scenarios
The definition of scenarios builds on the training activities specified in Step 2.1, a step that is also difficult. If the definition of the training activities is not suitable or not detailed enough, it is hard to work on scenarios. Furthermore, the concept of scenarios and the format that is used with a process, environment and system description is quite specific for the field of training simulators and, thus, new to the subjects in these studies. Indeed the results of a small evaluation with target users better acquainted with simulator training show that they had much less difficulty with this step (see § 9.2.2). A third cause for this problem may be that subjects found it hard to specify scenarios for simulator training apart from the rest of the course. This is illustrated by the following remark of one subjects in study A (translated from Dutch):

> "Also I find it strange that you cannot describe how you think the theory lessons should be. I think that this cannot be separated from practice. Only indicating whether this should happen in the classroom, or with books etc., is very minimal. I know how I would like it, more or less, but do the designers of the simulator know this also? What I really mean is this: with the specifications that I have defined for the simulator, the theory should be taught in a certain way, if not then maybe a different simulator would have to be designed."

More detailed information about the MASTER-TPD method from study B has been used for work with a similar ISD-based method for a slightly different purpose, see for a brief discussion Section 9.3.1.

8.3.2.3 Discussion

The results of the evaluation studies show that subjects were able to complete all MASTER-TPD steps. The most difficult steps are:
- Step 1.2: Determine Training Sequence
- Step 2.1: Define Training Activities
- Step 4: Planning of Assessment, and
- Step 5: Design of Scenarios.

There are indications that technical problems have played a role for both Step 1.2 and Step 4. Improvements to the tool that make it possible to cluster training goals and training activities and to keep an overview of the whole course at all times (e.g., with the non-simulator part greyed out) may help to solve these problems. In Step 4.2 the terms in the library need to be better defined. The subjects' problems with designing scenarios in Step 5 may be partly caused by the fact that the concept of scenarios and the format that is used are quite specific for the field of training simulators and, thus, new to the subjects in these studies. A better definition of the term scenario is required, with a set of examples. However, these problems are less likely to occur when users are more acquainted with simulator training (see also § 9.2.3).

The definition of training activities is difficult because it is the step from (instructional design) theory to the elaboration of a concrete training programme. It requires defining training activities independently of instructional products, and it requires a change of perspective from what needs to be taught (instructor perspective) to how it is going to be learned (trainee perspective). According to the developers of the MASTER method it is also an important step: linking training activities to training goals provides better guarantees that all goals will be achieved, it creates room for alternative instructional methods to achieve training goals and to deviate from operational activities, it can make assessment more transparent, it can make it possible to design training that can be adapted to individual trainees, etc. (Farmer et al., 1999). It is the key to innovative learning solutions: as long as designers keep on defining traditional training activities, the result will be a traditional training programme of classroom-based theory lessons followed by supervised practise. Results from the third questionnaire of study B show that six out of eight subjects find this step one of the most meaningful (see Table 8.9-B). This is encouraging, since this step is essential if designers want to make full use of the possible advantages of simulators for training and instruction. The results of both studies show that more support for this step is necessary.

In the prototype MASTER-TPD tool the guidelines with Step 2.1 provide some examples of training activities in rather general terms. The idea was that the subjects would elaborate these further in detail using, for example, information about the training goals, the target trainees, and the future learning environment. Inspection of the subjects' training programme designs showed that they mostly used the examples that were provided literally, and only rarely came up with new ones. This was surprising since these subjects had theoretical knowledge about instructional design models, and should, at least, know of possible alternatives. The chance that target users without a background in instructional design theory will define innovative training activities is even lower.

Choosing the right level of detail for the definition of training activities proved to be a large problem. A better library of training activities, accompanied by worked-out examples may help to solve this. Indeed the prescriptive instructional design literature (see, e.g., Reigeluth, 1999, p.22; and for more references § 3.3.2), can provide input for the development of a training activity library that also includes innovative training activities. One problem with this approach is that the MASTER method does not actually define the level at which training activities should be specified. The reason for this is to keep the method flexible: as it is, users can choose to work on different levels of detail depending on, for example, demands posed by acquisition procedures, the available time and resources, or characteristics of the design task (Farmer et al., 1999, see also § 4.2.2). Of course, it would be possible to develop libraries at different levels of detail as well, but maybe it would be better to encourage designers to spend more time and effort on defining training activities themselves, or to ask instructional design experts for help. The definition of training activities seems to be a design activity that could benefit from measures to stimulate creativity, such as organising a brainstorm session with colleagues and/or stakeholders (see also § 10.5).

A more fundamental discussion concerns the concept of training activities itself. The MASTER-TPD method prescribes the definition of training activities for each training goal in a rather strict top-down manner. Others, however, plead for the selection of one combination of training activities (Smuling, Brants and Pilot, 1990) or one combination of instructional products (Verwijs, 1998) for the whole course without such a one-to-one relationship. In the IMAT model training activities are described at a lower level, in the form of sets of teacher activities and learner activities (de Hoog et al., 2002; Verstegen, Veldhuis, Staalstra and Hendriks, 2001).

The central role of training activities in the MASTER method, and the demand to define them independently of instructional products, is also a point of discussion. Mettes, Pilot and Roossink (1981) propose the concept of instructional function to bridge the gap between learning goals (or training goals in MASTER terms) and the instructional procedures to realise these goals. This concept is taken from the world of engineering. The designer of a vacuum cleaner, for example, first decides which functions the machine will have to fulfil: it has to collect, transport and store dust (Holleman and Pilot, 2003). Likewise, instructional functions are the general operations that have to be performed in order to evoke the necessary phases of the learning process, such as orientation, practice, assessment and feedback (Mettes et al., 1981). Functions can be realised by (a combination of) activities with (a combination of) instructional products. Important is that all functions are realised, not how or by whom they are realised. A function model should be based on a learning theory that is suitable for the kind of learning goals that learners should attain (Pilot, 1997). From the learning theory, the instructional designers deduce the required learning phases and the functions required in each of those phases. Mettes and Pilot (1980) designed a function model for the domain of problem solving in science education. Such a specific function model can be reused for other courses in the same (or in a similar) domain, and can thus be seen as a partly elaborated template or hybrid design and development model (see § 3.3.4). Smuling et al. (1990) describe a number of other models for higher education. Holleman and Pilot (2003) propose a more general function model based on a number of function models that have been developed for different kinds of learning goals and/or incorporating different learning theories.

8.4 Validity of the results

In many respects the situation during the two evaluation studies described in Chapters 6, 7 and 8 was similar to the situation that the MASTER method was developed for: novice designers designing specifications for a training simulator. The case that was used was realistic in size and complexity. In fact, it was derived from a real case for which a task and training goal analysis had been done within another TNO project (see § 6.2). There are, however, *three differences* that may have consequences for the validity of the results:

1. The subjects in the two evaluation studies have less knowledge about the domain and simulator training in general than the target users are expected to have.
2. The subjects in the two evaluation studies have more background knowledge about instructional design and development models and theory than the target users are expected to have.
3. In real design processes in a real environment designers often work in the context of a design team, and not individually as in these studies (although it is possible that one person is responsible for a part of the design process and works individually part of the time).

There are no indications that a lack of knowledge about the domain prevented subjects without much domain knowledge from fulfilling the design task (see § 6.4.1.4, § 7.4.1.1 and § 8.2.3). They had some problems with simulator-specific terms and activities such as the design of training scenarios, but these could be solved with some extra examples and explanations. It was not possible to compare the subjects' designs to designs made by domain experts, because these were not available. However, the two instructors/SMEs that judged the subjects' training programme designs in the first study were enthusiastic, especially about the best ones (see § 6.4.1.4). Of course, designers will always need information about the domain, which can be provided in documentation and interaction with domain experts during meetings and/or by e-mail (as in the studies described in Chapters 6 to 8), or by involving domain experts as members of the design team (as in the case studies described in § 9.3.2).

To shed more light on the three differences described above, I will discuss the results of some studies in which the real target users were involved in Chapter 9 and compare the results with those from the evaluation studies described in Chapters 6, 7 and 8.

Chapter 9
Target users and the MASTER method

9

At the end of Chapter 8, I concluded that there are some differences between the subjects that participated in the evaluation studies described in Chapters 6, 7, and 8, and the target users of the MASTER method. To get information about the validity of the results of those evaluation studies I describe the results of some studies with target users. From a different perspective I discuss the following two research questions:

8. How is iteration used in the design process?
9. Can novice designers design simulator training programmes with the MASTER-TPD method in an adequate way?

These research questions are also addressed in Chapter 6 and Chapter 9.

9.1 Introduction

In the evaluation studies described in Chapters 6 to 8 subjects worked on a realistic case and spent a realistic amount of effort on the design task. For the most part, the subjects in the evaluation studies had the same characteristics as target users and they worked in a very similar way. However, the subjects did have more background knowledge about instructional design and development models, and less domain knowledge than most target users of the MASTER method would have. Furthermore, they worked individually, and not in the context of a design team. To check the validity of the results presented in the previous chapters, I will compare the results of the evaluation studies described in the previous chapters with the results of studies with real target users. The studies described in this chapter were limited in time scale, and rather informal, because they took place in the context of other projects.

In Section 9.2, I present the results of a short evaluation study with the MASTER-TPD method and prototype tool with participants from the Royal Netherlands Army (RNLA). Subsequently, in Section 9.3, I describe the method that has been developed for the development of needs statements for the RNLA, the SLIM method. The SLIM method is based on the MASTER method, but is used in a different way: in workshops with all stakeholders under supervision of instructional design experts. I discuss the results of five case studies with the SLIM method, and describe how the SLIM method deals with some of the problems encountered during the main evaluation studies. In Section 9.4, I briefly describe possible ways to support the communication in a design team. Finally, in Section 9.5, I discuss the conclusions regarding the research questions in the light of the differences between the research subjects who participated in the main evaluation studies described in the previous chapters and real target users.

9.2 Evaluation study with target users

A short evaluation study with the MASTER-TPD method and prototype tool was executed in the context of a project for the RNLA. The goal of this study was to allow target users to briefly work with the method in order to gain insight into its usability and, more in particular, its usefulness for the development of needs statements for technically advanced instructional products. Although the MASTER-TPD method is not specifically oriented towards this early phase of the acquisition process, it is possible to go through the whole sequence of steps quickly, doing only global analyses and taking preliminary decisions, in order to get a first set of global specifications (see § 5.4.1).

9.2.1. Design of the study

The evaluation study took place at TNO Human Factors at the 10th and 11th of November 1999. Just like the subjects in the two evaluation studies described in Chapters 6 to 8 the participants worked through the MASTER-TPD steps individually, using the same research prototype tool (see § 5.2.1) and case (see § 6.2). They were asked to imagine that they were responsible for a part of the development of a needs statement. The first part, the Training Needs Analysis (TNA) had already been done by a colleague, and it was now their responsibility to design a global training programme for the analysts of thermal images from UAVs.

The participants were five domain experts from the RNLA, i.e. personnel likely to be involved in the design of needs statements for advanced instructional products. Two of the participants had already been involved in such a project. Three participants had teaching experience, but little or no experience in teaching with simulators or other technically advanced instructional products. Four of the five participants had experience in designing courses and lesson material. Finally, two participants were from the Army school that will be responsible for the training of UAV personnel and can, thus, be regarded as Subject Matter Experts (SMEs) for the domain that was used in the evaluation studies[38]. Note, however, that at the time of this evaluation study UAVs were not used by the RNLA, and the courses for UAV personnel did not yet exist. The acquisition of simulators and/or other instructional products for these courses had not yet started.

The first day started with an introduction to the MASTER-TPD method, followed by some guidelines for the use of the prototype tool. The participants also received a printed version of the introduction (the same one as used during studies described in Chapters 6 to 8). Then some information about the case was provided (together approximately one hour). Finally, the participants were asked to complete a brief questionnaire about their background (see Appendix A in Verstegen and Barnard, 2000). Subsequently, the participants worked through the MASTER-TPD method in the following way:

- One step and its sub-steps were explained by the evaluator (i.e. the author of this thesis).
- Then the participants worked on the step individually.
- Then there was a plenary discussion to discuss the step.

Going through the six steps of MASTER-TPD took 8 hours (divided over 2 days). The participants worked on laptops in a room at TNO Human Factors. During the whole study TNO staff was present to answer questions regarding the method and to solve problems with the tool. Afterwards, there was a more elaborate, plenary discussion and participants were asked to fill in a second questionnaire about the usability of the MASTER-TPD method (see Appendix B in Verstegen and Barnard, 2000).

9.2.2 Results

The results are mainly based on the discussions with participants, where possible supported by data from the second questionnaire. Quantitative analysis methods have not been used because of the limited time scale and informal nature of this evaluation study. For the same reasons no attempt was made to judge the quality of the resulting training programme designs. The discussions in between steps and at the end were lively and informative. Comments regarded the MASTER-TPD method, as well as the design of needs statements for simulators and other advanced training instructional products. A summary of the results is presented below, see for a complete description Verstegen and Barnard (2000).

A) Working with the MASTER-TPD method
With the support that was provided the participants were able to fulfil their assignment: based on the given set of training goals and the description of target trainees, they worked on a draft simulator training programme. The results of the questionnaire confirm that they did not judge this task as particularly difficult or unpleasant.

B) The structured approach
During the discussions the participants concluded that the MASTER-TPD method takes a systematic approach, and makes sure that nothing is forgotten. However, it does not guarantee good results: users can still take the wrong decisions. The results of the questionnaire show that the participants think that the structured approach is useful. Most of them also liked this way of working.

C) Iteration
Even during this short session, it was clear that the design process was not linear. For example: one participant realised later on that he wanted to put the training goals in a different sequence then he had done initially; another participant decided that some parts that he had

[38] In fact, they were the SMEs that judged the training program designs developed by subjects in the first evaluation study (see § 6.4.1.1). They only did this, however, after participating in the study described here.

assigned to simulator training did not have to be covered with the simulator after all. The participants explained, that in their work situation the stimulus for iteration is usually a discussion with other people that are involved, e.g., instructors or those responsible for the financial resources. Some of them really disliked having to change or review their design; others did not have any problems with it.

D) Teamwork
The development of needs statements (and specifications) is teamwork. The discussions did not make clear how tasks are divided and who takes the decisions in this process. The participants stated that it is important that the end-users (i.e. the school who will use the simulator/instructional product) are involved more than is currently often the case.

E) Active advice
There was a lot of discussion about more active forms of advice. Most participants would like an 'automatic' judgement from the tool, as illustrated by the following comments: "I thought that the computer would tell me whether I have made the right decision" and "Can it not warn me when I make a mistake?" Although these expectations are clearly exaggerated, it would be possible to develop more active forms of advice. Some forms that were discussed are: provide checklists at important decision points, and classify training goals in a taxonomy and provide heuristics for selecting suitable training activities based on this taxonomy.

F) Taking notes
Providing arguments for decisions is very important according to the participants. Maybe users should be forced to make notes with each step, or at least forced to provide arguments for important decisions.

G) Coverage of the MASTER method
The MASTER method has been designed to develop training simulator specifications. It does not cover the specification of other instructional products. The method focuses on instructional factors. In the acquisition process other factors are also important, such as: financial and other resources, the power of the different parties involved and organisational factors (see also § 3.2.1). These are not explicitly analysed or represented. On the other hand, one of the participants commented that this is the way he would like to work: "I want to design the optimal solution from an educational point of view first, so that I know what I want before I start the negotiations."

H) First check
Developing a needs statement (and later on detailed specifications) is a lot of work. It is wise to check first whether it makes sense to start this process, answering questions such as: will there be a reasonably large number of trainees? Will there be budget for the acquisition of simulators (or other instructional products)?

I) Level of detail
Some participants said that, in practice, training goals are not specified at the detailed level used in this case (see Appendix C) for a needs statement. Another participant, however, said that he would prefer to do this even though it does not usually happen. The conclusion of this discussion was that users should be able to determine themselves at which level of detail they want to work.

J) Weigh alternative solutions
The participants would have liked to weigh alternative learning solutions against each other. Especially for the needs statement phase the end result should be a minimal set of specifications (i.e. describing what is absolutely necessary) plus preferred extras (i.e. specifying facilities that would make training more effective and/or efficient) and possible alternatives and their consequences. For example describing a solution with a cheaper simulator in combination with using the operational system for a part of the training. This is possible with the

MASTER-TPD method (see § 4.2.2), but it is not clear how it should be done: should alternatives be developed in parallel, or should designers pursue one option and check later on what the consequences are if some facilities are not feasible? Moreover, there is no specific support for comparing alternative solutions.

K) Definition of training activities
Defining training activities (MASTER-TPD Step 2.1) was difficult for the participants. They asked for a good library with examples of training activities. In this way, the user can still define his own training activities, but also choose and reuse those from the library. More active forms of advice for this step were also discussed (see above). The selection of training activities for simulator training (MASTER-TPD Step 2.2) was regarded as a crucial step in the MASTER-TPD method, because it determines the further design.

L) Other comments regarding MASTER-TPD method
The most important other comments regarding the MASTER-TPD method were:
- Structuring training goals: training goals should be provided in a hierarchical structure (as result of the Training Needs Analysis); and in Step 1.2 (Determine Training Sequence) it should be possible to (re)cluster training goals into themes or lessons (see also § 8.3.1.2).
- Priorities in Step 1.3 (Allocate Training Media): in this step users are asked to allocate a range of applicable instructional products to each training goal; the participants would have liked to assign priorities to the different instructional products.
- Weights for training goals: participants indicated that they would like to assign weights to the training goals, because the most important or critical training goals should have more influence on the design than the less important or less critical ones.
- Overview of the course: the participants did not like the fact that training goals that have not been assigned to simulator training are no longer visible in the later steps (see also § 8.3.2.2).

9.2.3 Conclusions regarding the MASTER-TPD method

Background in instructional design theory
The participants in this study are target users of the MASTER-TPD method. They are beginners with respect to the MASTER-TPD method. They have no experience with the specification of training simulators, or with the use of systematic methods to design specifications of instructional products. Three out of the five RNLA participants have teaching experience and four of them have experience in designing courses and lesson material. For their jobs they have followed a course for course designer and/or the course for instructor. However, these courses are relatively short (15 and 22 days respectively), and thus the RNLA participants have much less knowledge about instructional design and development theory than the subjects in the evaluation studies described in Chapters 6 to 8. The results described above indicate that such an extensive theoretical background is not required for using the MASTER-TPD method, at least not when experts are present to answer questions about the method and the tool, as was the case in this study. Some caution is necessary with regards to this conclusion: due to the limited time scale, it was not possible to check the quality of the resulting training programme designs. Therefore, it seems advisable to include an instructional design expert in the design team or ensure that the team has access to such people for support.

Difficult steps
Also in this study defining training activities (MASTER-TPD Step 2.1) was explicitly mentioned as the most difficult step (see also § 8.3.2.1). The participants in this study also asked for more support for analysing and representing information about available resources, including a first check to see whether it makes sense to start a design process, and for comparing alternative learning solutions. Maybe extra steps should be included to focus the

users' attention to these important decisions. In such steps users could be advised to discuss their ideas with colleagues, expert instructional designers and/or other stakeholders. An alternative would be to develop dedicated decision support. In fact, the BOOT tool (van der Hulst, de Hoog, and Wielemaker, 1999), described in Section 3.3.7.2, is developed to compare alternative solutions. Contrary to the subjects in the other evaluation studies, the RNLA participants do not have difficulty with the design of prototypical training scenarios (MASTER-TPD Step 5). This was to be expected since they have more domain knowledge, are used to the concept of training scenarios and are likely to have trained -as trainees and/or as instructors- with training simulators themselves.

Iteration and level of detail
According to the participants in this study the strongest triggers for iteration are the acquisition procedures and discussions with other stakeholders. In this more pragmatic setting it became clear that one of the strong points of the MASTER-TPD method is its versatility: the same steps can be executed at different levels of detail, and for different purposes. Combined with the structured approach this ensures a systematic design process and documentation of all relevant information with the step that it is relevant for. The same versatility is, however, also a weak point. It is up to the users to decide at which level of detail they need to work, how many alternatives they should consider, when alternatives need to be further elaborated and when they can be discarded, if and how decisions are explained in notes, etc. Not unexpectedly, this proved to be difficult for novice designers in all evaluation studies. It means that, more than in other design processes, it is not clear what the result should look like. An architect knows which information the builders need in order to build the house he is designing, but specifications for a simulator can be written in different ways and on different levels of detail. A standard format for needs statements is still lacking.

Insecurity
In the first evaluation study described in Chapter 6 subjects found it difficult to deal with insecurity (see § 6.4.1.4). This was not the case during the second study, probably because the subjects in that study were paid to participate in the study irrespective of their results (see § 7.4.1.5). In practice, the consequences of failure will have large consequences for users of the MASTER-TPD method. Careers and large sums of money are at stake, not to mention the dangers of badly trained personnel: damage to operational equipment and possibly even the loss of lives. This may partly explain why the participants in this evaluation kept on asking for (automatic) feedback from the tool. Another reason may be that they have (too) high expectations of support from computer tools. This issue was also mentioned by subjects in the informal evaluation executed within the MASTER project (see § 4.2.4.2).

9.2.4 Conclusions regarding needs statements

The evaluation study described above was executed in the context of a project that TNO has executed for the RNLA. Starting point of this project was the question for a systematic method for the development of needs statements for technically advanced instructional products, such as CBT, training simulators and E-learning. The development of a needs statement is the first phase in the acquisition procedure of the RNLA (see § 3.3.6). The goal of this phase is described as: determine the present and future needs for instruction and training, within a limited amount of time and on the basis of information that can be incomplete and insecure; decide whether and why the acquisition of advanced instructional products will be necessary and roughly estimate how much (financial and other) resources should be allocated for this purpose (Verstegen, Barnard, van der Hulst, and Sabel, 2000). For an organisation, needs statements are the basis for the planning of financial and other resources. At present, needs statements for advanced instructional products are often written in an ad hoc way, not based on training needs and focusing on physical fidelity. In other cases, the acquisition of instructional products is postponed until the new operational systems have arrived, which leads to unacceptable delays in personnel training (Verstegen and van der Hulst, 2000).

Based on the evaluation study described above and discussions with other stakeholders within the RNLA, the conclusion was that the MASTER method could be a valuable basis for a systematic method for the development of needs statement. However, the MASTER method is not suitable as it is, because (Verstegen, Barnard, and van Rooij, 1999a):
- it is not specifically geared towards this early phase in the acquisition process,
- it only covers the specification of training simulators, not of other instructional products, and
- it is not adapted to the context of the RNLA organisation and not integrated with the other RNLA procedures.

For the needs statement phase, the MASTER method with its many steps and sub-steps was considered to be too detailed. However, the global structure of the method is a suitable framework. Like the MASTER method, existing RNLA procedures for the later phases are based on an ISD approach (see § 3.3.3). Therefore, integration should not be difficult if the terminology is adapted. Although the last part of the MASTER method is specifically geared towards the specification of training simulators, it appeared that the specification of other instructional products could happen along the same lines, especially in this early phase.

9.3 Case studies with the SLIM method

Based on the MASTER method a specific method has been developed for the development of needs statements for technically advanced instructional products: SLIM (Specifying Learning means[39] in an Iterative Manner) method. The SLIM method is adapted to the RNLA setting and procedures and focuses on the first phase in the acquisition process of the RNLA (see § 3.3.6). The SLIM method covers both Analysis and Design (cf. Figure 1.1) on a global level rather than in detail. Therefore, I will use the terms development team and development process in this section (cf. § 3.2.1). In Section 9.3.1, I will briefly describe the SLIM method. In Section 9.3.2, I present results of five case studies in which the SLIM method was used.

9.3.1 The SLIM method for development of needs statements

The SLIM method leads users through a global development process. This does not, however, mean that the development process will always be the same. Depending on, for example the complexity of the domain and the amount of information available, certain design activities will take more or less time, and some design activities will sometimes get less emphasis or will only be partly executed. The SLIM method consists of four main phases:

I.	Problem definition:	Inventory of the intended outcome/results, the demands, available resources and project risks.
II.	Analysis of training needs:	Global analysis of need for instruction and training.
III.	Design blue print & Selection of instructional products:	Design first draft of the course and select a suitable combination of instructional products.
IV.	Design of specifications & Cost estimation:	Definition of global specifications and estimation of the resources that will be required for the acquisition and use of instructional products.

Developing a complete and thorough needs statement takes time and manpower. Therefore, the SLIM method starts with an inventory of the situation in order to decide whether it is worthwhile to invest in such an undertaking. This phase is meant as an explicit analysis of the design problem, as advised by prescriptive literature (see § 3.3.3.3) and observed in the behaviour of expert instructional designers (see § 3.2.2.3). Expert designers also generate several alternative solutions, and do not commit themselves to one particular solution until

39 In this thesis the term instructional products is used.

later on. This kind of behaviour is encouraged by asking users to sketch a number of possible solutions up front, in general terms, during the first phase (see below). In the second phase a global analysis of the needs for education and training is executed (cf. Training Needs Analysis in the MASTER method, see § 4.2.1), and in the third phase a first draft of the course or training programme is designed (cf. Training Programme Design in the MASTER method, see § 4.2.1). Finally, in the fourth phase global specifications for the required instructional products are defined (cf. Training Media Specification in the MASTER method, see § 4.2.1), and a rough estimate of how much they would cost is calculated. Three kinds of resources play a role here: resources for the acquisition of instructional products, resources for the use and maintenance of those products in future training, and resources for the development of (detailed) specifications.

The intention of the SLIM method is not to develop final and fine-grained specifications, but to start thinking about how education and training will probably take place and which kind of instructional products will probably be needed. On this basis a reasonably realistic cost estimation can be made, taking several alternatives into account. Figure 9.1 provides an overview of the four phases of the SLIM method.

The SLIM method has been applied during workshops with all stakeholders and with two experienced instructional designers as facilitators (see § 9.3.2.1). Iteration is encouraged by organising a sequence of workshops a couple of weeks apart. In between workshops the participants can reflect on the results so far, collect additional information and discuss with their colleagues. New information or discussion points are input for the next workshop, and thus taken into account immediately. Inviting all stakeholders ensures that all arguments will be taken into account, and that the selected solution will be accepted within the organisation. In fact, this format can be seen as a combination of a systematic and a relational or communicative approach, as advised by Kessels (1999, 1993), amongst others (see also § 3.3.7.1).

The SLIM method also deals with some of the problems with the MASTER-TPD method encountered during previous evaluation studies. Six of these are discussed below:

A) Dealing with resources
The SLIM method starts with an explicit inventory of demands, resources and risks (Step I-2)[40]. These are stored on three lists that are made available at all times. After each step users are reminded explicitly to keep these lists up to date.

B) Generating alternatives
As discussed above, users are asked to sketch a number of possible solutions in general terms up front. This is part of Step I-1 and is meant both to make the participants' preconceived ideas explicit and as an inventory of alternative solutions. Comparing alternative solutions comes back as an explicit step half way the design phase (see below, under point D). For now, it is up to the workshop facilitators to make sure that the participants keep on considering feasible alternatives during the rest of the design process (see § 9.3.2.1).

C) Explicit decision points
After each step an explicit decision point is introduced. A general checklist with recurring questions or considerations has been developed (e.g., "Are your assumptions still correct?"), and at important decision points more specific questions are asked (e.g., after the selection of instructional products users are prompted to consider possible alternative solutions).

[40] As proposed by target users (see § 9.2.2) and one of the subjects in the second evaluation study (information from the third questionnaire).

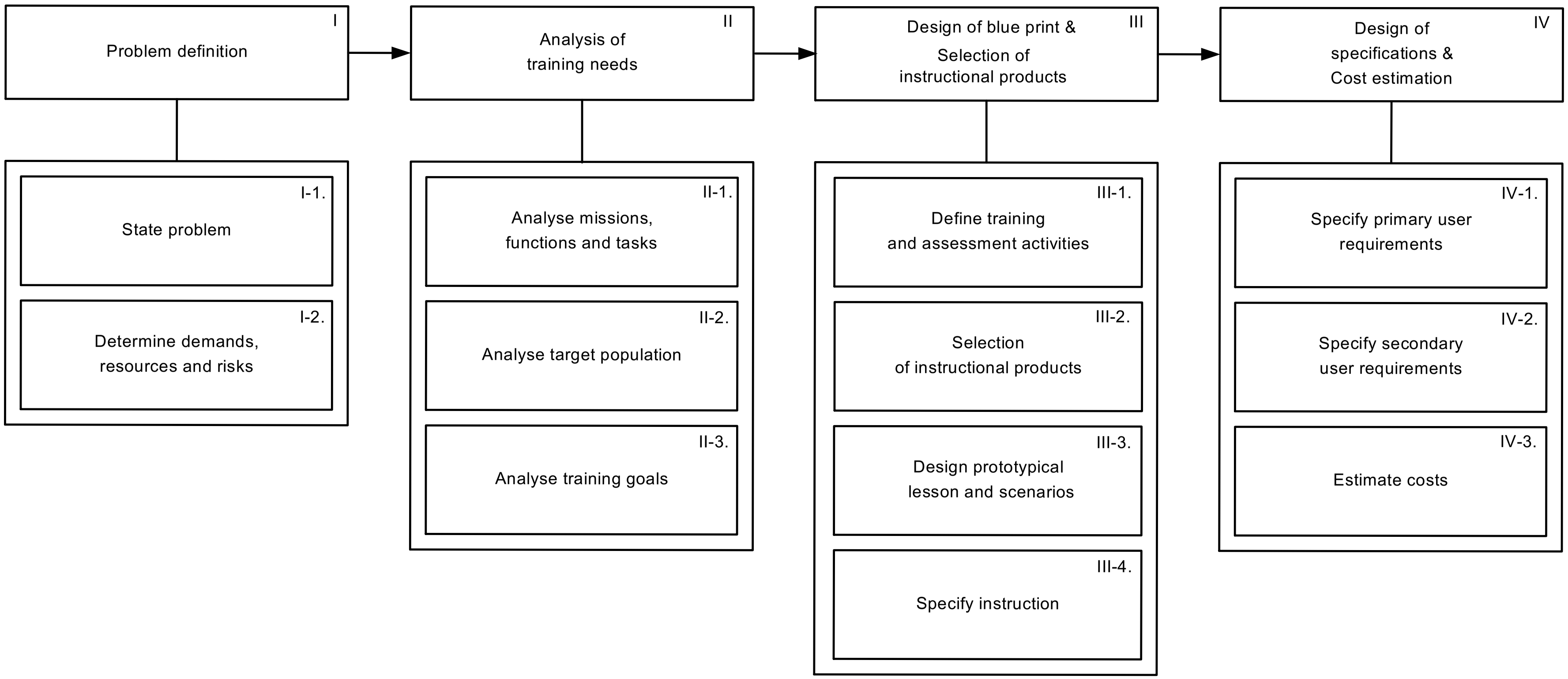


Figure 9.1: The steps of the SLIM method for the development of needs statements for technically advanced instructional products.

D) Prioritising solutions for individual training activities
To encourage users to consider alternatives they are explicitly asked to select a wide range of suitable instructional products for each training activity, and to prioritise them from most desirable to least desirable (this is part of Step III-2). Subsequently, users select a combination of training means with which all training activities can be executed, though maybe not all in the most preferred way. Users are advised to envision alternative combinations and to define their advantages and disadvantages, before making a choice. Later on in the acquisition trajectory these alternatives can be reconsidered, for example when the resources do not permit the acquisition of (a part of) the initially chosen set of instructional products (Note that, like in the rest of this thesis, the term instructional product is used for any product that is made to help a trainee acquire some knowledge or skills, including not only technical systems such as CBT or simulators, but also text books, exercises or assignments, and lesson plans to be used by a teacher in a classroom).

E) Assessment
Evaluations with the MASTER method indicated that assessment planning without an overview of the whole course is difficult, and that there is some overlap between assessment and debriefings[41]. In the SLIM method another point of view has been chosen: assessments are regarded as a special category of training activity and are planned in the same step. Subsequently, they are elaborated in the same way, i.e. a range of suitable instructional products is chosen and a 'scenario' for the assessment is described in Step III-3.

F) Level of detail
The SLIM method is meant to be used at a global level of detail. All support is focused on helping users to go through the steps in a limited amount of time. At the same time it ensures that the information collected during the first phase of the acquisition process can be reused later on when specifications are developed at a more detailed level with a more detailed ISD-based method (e.g., the MASTER method) as depicted in Figure 9.2:

9.3.2 Five case-studies with the SLIM method

The SLIM method has been applied in five case studies (Boot, Verstegen, and Veerman, 2002; Janssen, Boot, and Verstegen, 2003; Melis and van Berlo, 2003; Verstegen, Veerman, and van der Arend, 2001). The main results of these case studies are described in this below.

9.3.2.1 Design of the case studies

In the five case studies workshops were organised with all stakeholders and with two experienced instructional designers as facilitators. During the workshops the roles were divided as follows:
- The workshop leader was fully responsible for the management of the development process: explaining the goals of the SLIM method, organising the development process, structuring and guiding the discussions, making sure that no available information is neglected and that all feasible alternatives are taken into account, deciding when to go to the next design activity and also when iteration was necessary.
- The second facilitator took notes and was responsible for the documentation of the results as well as the development process.

[41] One subject in the second main evaluation study described in Chapter 7 did not understand the difference between the planning of assessment in MASTER-TPD Step 4 and the planning of the debriefing in MASTER-TPD Step 6.3 (information from the third questionnaire). This can be confusing, because assessment does not always have to be planned as a separate event: sometimes ongoing assessment is planned with each exercise and the results are given by the instructor during the debriefing.

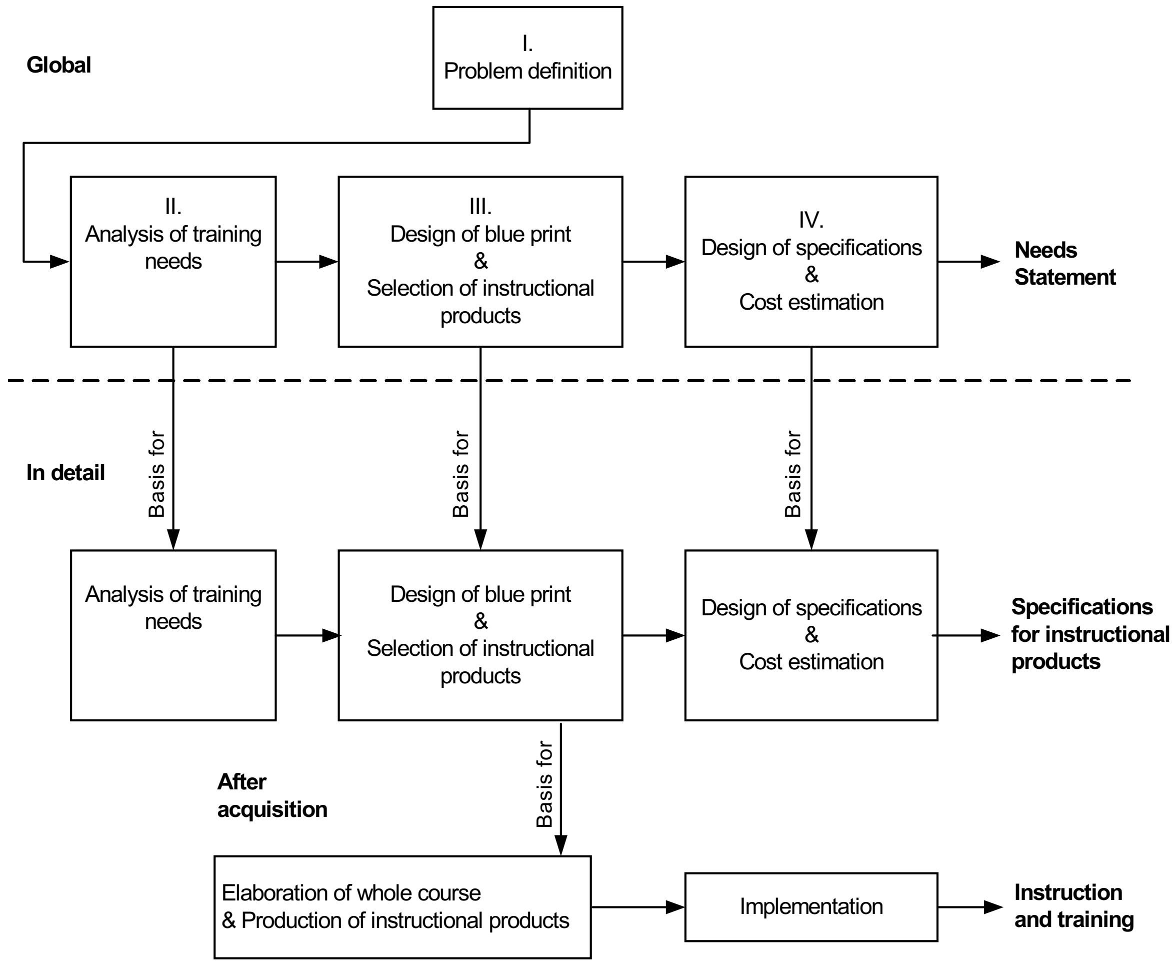


Figure 9.2: Reuse of information collected with the SLIM method.

- The role of the workshop participants was to provide the necessary problem and domain information, and to take decisions.
- Only on request, the facilitators brought in their own knowledge and experience in the field of instructional design and development.

During most sessions the author of this thesis took the role of workshop leader (with the exception of three sessions divided over two different cases). The role of the second facilitator was fulfilled by different TNO colleagues.

The results presented below are based on the written reports that were made of all the sessions, the discussions with the participants during the sessions, and the experiences of the facilitators. In the context of this thesis it is interesting to investigate:

a. the role of iteration, and the way it was managed during these workshops, and
b. the design of specifications for instructional products in a team setting, rather than individually.

I will come back to these questions in the discussion in Section 9.3.2.6.

9.3.2.2 The five cases

The SLIM method has been applied in five different cases for different customers. In three of the cases, the resulting needs statement describes a CBT application, in one case a desktop-based simulation and in one case a simulator. All the cases were complex, but for different reasons. In one case the main source of complexity was the fact that the operational system did not yet exist and that there was not much information available about the tasks that trainees would have to be trained for. In another case the tasks were relatively easy, but the complexity from an educational point of view was caused by the very large number of trainees to be trained for different subsets of tasks. In two cases, the same instructional products would have to be used by clients from two rather different organisations who were using the operational system in slightly different ways; there were also some differences in organisational culture, capabilities of instructors and target trainees. And in one case, the course addressed during the workshop was meant to be just an example to investigate the potential role of simulation-based CBT, to replace (part of) the practice with simulators and/or operational systems.

Since the complexity of the cases differed, the amount of time required to execute the SLIM method differed as well. The case studies took 4 to 7 days divided over 2 to 4 sessions. In between sessions the participants went back to their workplace to collect more information and discuss the results with their colleagues. In some cases, the workshop participants took away 'homework', i.e. further elaboration of steps to be done independently in between sessions. However, in some cases participants were not capable and/or willing to do this. It should be noted that the level of detail of the results differed as well, mainly because the amount of time available for workshops was limited by project boundaries.

9.3.2.3 The SLIM method applied to different cases

During the workshops it became clear that in all cases participants misinterpreted some terms, or used them in slightly different ways. When miscommunications hindered the process, terms were redefined on the fly or replaced by terms that were better known by the participants. In some cases the method was slightly customised as well, e.g., combining steps three and four in the design phase in a less complex case, or leaving out the fourth phase because it was not the responsibility of the workshop's participants.

In the experience of the facilitators the method was a valuable instrument in structuring the development process and maintaining focus and consistency. The participants valued the structured approach of the SLIM method, but clearly preferred not to be bothered with details regarding the SLIM method or instructional design and development theory in general. The facilitators experienced the ad-hoc customisation as an advantage of applying the SLIM method in a workshop setting in direct contact with the stakeholders, but also as an extra burden. Applied in this way, the SLIM method proved to be robust and versatile enough for all five cases.

Note that in this format, the participants take all the final decisions. This means that the SLIM method works towards a solution that suits the organisation and is accepted by all stakeholders, a solution that can be -but is not necessarily- innovative. The acquisition of instructional products had to solve existing or expected problems of the organisation. Sometimes, new types of instructional products were required, e.g., because the participants expected that they would have much less opportunity to let trainees practise with the operational system in the future. In one case, the explicit goal of the workshop was innovation (i.e. simulation-based CBT, see above). In another case, however, the participants decided that for a new course, classroom based lessons were most suitable, much like the classroom based lessons in existing courses at the school. Another observation was that the participation of all stakeholders is vital. In one of the cases not all stakeholders were present during the workshops, which led to uncertainty and speculation about 'what they would want', and extra time investment in reporting the results of the workshops, getting feedback and adapting the results accordingly.

9.3.2.4 The role of the facilitators

The role of the facilitators proved to be both demanding and important. When the workshop leader knows the SLIM method very well (as was the case for the author of this thesis), it is possible to adapt terminology on the fly, and to decide when and how the SLIM method can be customised without violating the important principles underlying the method. The case studies made clear that the workshop leader needs to be familiar with instructional design and development models in general and, in detail, with the SLIM method, in order to be able to organise and manage the design and to keep an overview. The interaction can be quite unorganised, sometimes almost chaotic, since the workshop participants do not (and are not meant to) pay attention to the SLIM method and the management of the development process. This means that the second facilitator, who is responsible for the report, also needs to be familiar with the SLIM method in order to be able to summarise and reorganise information that may come up at different moments during the workshop according to the steps of the method.

The workshop format proved to be an efficient way to collect all available information, make the different opinions and arguments of stakeholders explicit, and come to joint decisions. However, it also introduces a social aspect to the design task. In most cases the participants had different backgrounds and conflicting interests, and there were some overpowering personalities. Participants had different positions in the organisation's hierarchy, and often existing problems or power struggles played a role as well. Managing these group dynamics is extra task for the workshop leader. It proved to be important to state the roles of the facilitators and the workshop participants clearly

Again, in three cases, the participants had very high expectations of the knowledge available in the area of instructional design, illustrated by remarks such as: "So, just tell us which training goals need CBT", "I'm sure there has been done a lot of research about how to teach these tasks", and "There must be ready-made solutions described in literature" (cf. the results of the evaluation study described in § 9.2.2 and the comments of users during the MASTER project described in § 4.2.4.2). Like other instructional design tasks, however, the development of a needs statement is an ill-structured problem for which there is, usually, more than one good solution (see § 3.2.2.2), and several prescriptive instructional design models that could be applied (see § 3.3.2). Which one is selected, then depends largely on other factors such as the available budget, the amount and capabilities of the instructors, etc. This was an eye-opener for many participants.

9.3.2.5 Iteration and dealing with uncertainty

In principle, the goal was to go through the entire SLIM method at least twice in two explicit design cycles (cf. the design cycles in the evaluation studies described in Chapter 6 and Chapter 7). In practice, a more pragmatic approach was applied: during the first workshop the SLIM method was carried out step-by-step as far as the available time allowed. At the beginning of the next session the report of the previous session was discussed step-by-step and the information was corrected or elaborated where necessary. In all cases there was also iteration within sessions. This was usually caused by the fact that the required input for a later step was incomplete or not elaborated into enough detail or because of new information from sources or people not present at the workshop. Sometimes, participants phoned colleagues during the workshop for missing input information. The participants were less inclined to iterate in order to improve their design or to consider alternative solutions. In one case, the participants proposed to split of the first phase (Problem Statement) into a one-day workshop, followed by an interval to collect additional information.

Uncertainty played a role in all cases, but for different reasons. In the case where the operational system did not yet exist there was much uncertainty about the tasks, especially about the division of tasks between team members and the learning difficulty of the tasks. In another case the source of uncertainty was an ongoing reorganisation with -as yet- unknown consequences for the division of tasks over personnel. In a third case, the participants' task was to develop a needs statement for CBT to be used in addition to a simulator for which the specifications were being developed simultaneously by other people. It was not yet clear what the capabilities of the simulator would be, and indeed not even whether the simulator would be bought or not. In the fourth case, the main goal of the workshops was to explore the possibilities for using CBT in the school. The domain that was taken, was just an example and it was not clear whether CBT would be necessary or (financially) feasible for this course. And in the fifth case, much uncertainty was caused by the fact that not all stakeholders were present during the workshop (see also § 9.3.2.3).

9.3.2.6 Discussion

In the context of this thesis, the following questions regarding the case studies of interest:
- What is the role of iteration, and how was it managed during the workshops?
- What is different when specifications for instructional products are developed in a team setting, rather than individually?

The case studies show that iteration is necessary even during the development of a needs statement (which is only the first phase of the acquisition trajectory, see § 3.3.6). Especially in this early phase in the acquisition process often not all information is available or information is insecure, e.g., when it is not yet clear which operational system will be procured. Another reason for uncertainty can be that it is difficult to foresee whether ideas will be feasible within the given technological possibilities and resources, especially since these are often partly unknown or the information about them is insecure. During the case studies, the participants were encouraged to make assumptions and take preliminary decisions based on whatever information was available. Assumptions were written down explicitly, and reported under a separate heading in the workshop reports. At the beginning of each session, the participants decided whether these assumptions were still valid. By going through the report of the previous session step-by-step, the facilitators and participants checked where corrections, revisions or elaborations were necessary. On top of this explicit form of iteration, the workshop leader also decided to iterate during the sessions whenever necessary.

Working with a team, rather than individually, seemed to make the process more complicated. There were almost continuous 'interventions' caused by discussions between workshop participants or by new information brought in by one of them. Information or opinions from other people, collected in between sessions, also caused iteration. The SLIM method provides a framework that ensures that all important issues are systematically addressed. It also helps to manage the iterative design process. When participants bring in new information, the workshop leader decides to discuss it immediately or to come back to it at a later moment. The second facilitator makes sure that it is documented with the right step in the report. Working with a team also introduces a social aspect to the development process. It is possible that GroupWare systems could support this aspect of the development process during the workshops. Other forms of software can support the development team asynchronously (e.g., in between workshops), see for example Section 9.4.

During the SLIM workshops the different aspects of the design task were split up between the facilitators and the participants. The workshop leader took care of managing the design and development process (Aspect C in § 3.2.3) and, to a large extent, also of the communication (Aspect D in § 3.2.3). The workshop participants were responsible for gathering information (Aspect A in § 3.2.3) and solving the design problem (Aspect B in § 3.2.3), supported by the structured approach of the SLIM method. The second facilitator took care of storing all the collected information. This proved to be a successful way to involve these participants who were novices in instructional design. Based on the experiences during the case studies, prospective users are recommended to ensure that facilitators have at least a background in instructional design and development theory and experience with both systematic and communicative design approaches.

9.4 Support for design teams

In reality, designers often work together in a team (see § 3.3.7.1). Prescriptive methods contribute to the communication process by providing a shared terminology, an understanding of the steps to be taken in the development process and a set of intermediate products (e.g., problem definition, training goals). However, working together requires communication and co-ordination, i.e. the exchange of information and the division and planning of tasks (Verstegen, 1997). Team members depend on each other: they need information from each other and some tasks need to be done before others. Sometimes, different parts of the design process are the responsibility of different team members or different teams. Personnel changes during the long simulator acquisition process can be another complication, especially in military settings with job rotation schemes. Finally, communication can be difficult because team members often have different backgrounds and different interest (see § 3.2.1).

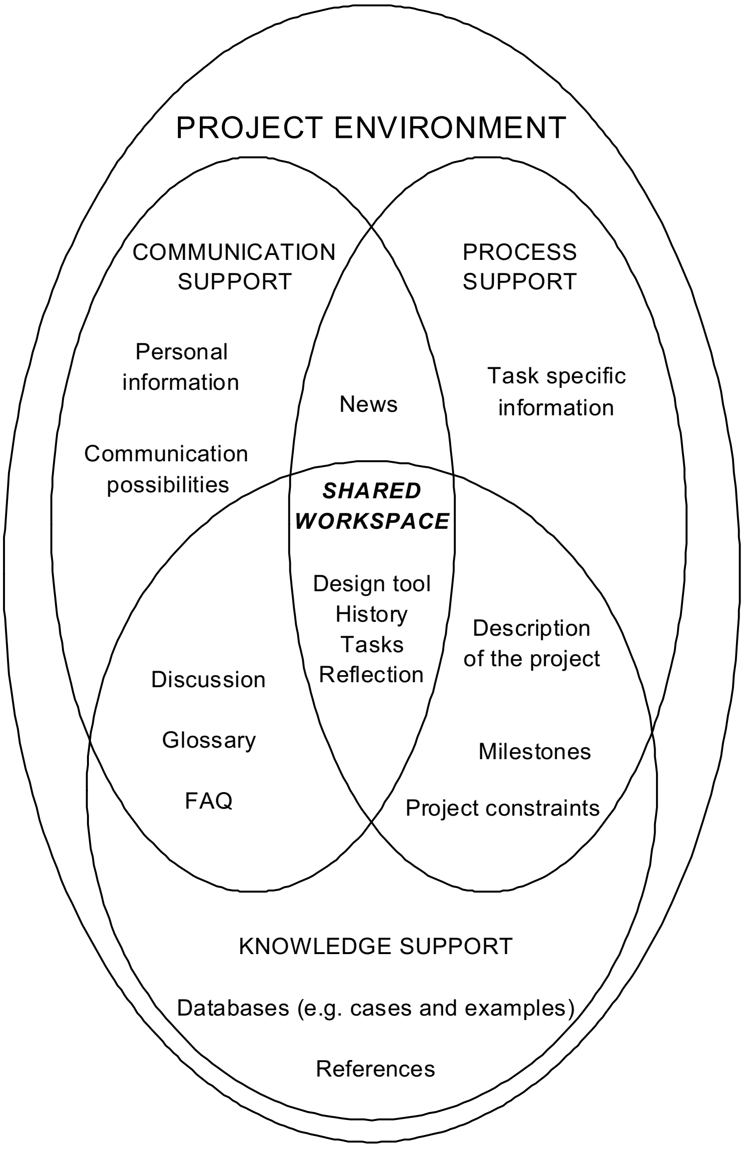


Figure 9.3: An appropriate structure for a project environment like MASTER+ (From Zagers, 2001, p. 89; used with permission of the author).

In the context of the evaluation studies described in Chapters 6 to 8, Zagers (2001) investigated ways to support working in a design team and proposed a shared workspace as depicted in Figure 9.3. Zagers proposes a web-based project environment that provides three categories of support:
- communication support,
- process support, and
- knowledge support.

At the core of the project environment is the shared workspace that contains the design tool that the team has chosen to use, a project history, a dynamic list of tasks and responsibilities, and a place for reflection with room for experiences of the design team, feedback from experts, peers and stakeholders, etc. Around the shared workspace are other facilities. The design process and team communication are further supported by 'News': a list of announcements about anything that concerns the design team. Further knowledge and communication support is provided by a list of Frequently Asked Questions, a glossary (related to the chosen design tool) and a place to store information from both face-to-face and computer-mediated discussions (Discussion). Different kinds of project-specific information can be seen as part of process and knowledge support: a description of the project, including the client, the target group, the goal and the expected result, a definition of project milestones, and a description of the project's constraints, e.g., the financial, time and technical constraints. More background knowledge can be provided in the form of references to books, web sites or people, and databases with examples derived from previous projects or from instructional design literature. The project environment foresees a place for personal information about the members of the design team and facilities to support communication, such as e-mail, bulletin boards and video-conferencing. Finally, more background information regarding specific design tasks supports individual designers in executing the tasks they were made responsible for.

9.5 Conclusion

In Section 8.4, I concluded that there are three differences between the subjects that participated in the evaluation studies described in Chapters 6, 7, and 8, and the target users of the MASTER method:

1. The subjects in the two evaluation studies have less knowledge about the domain and simulator training in general than the target users are expected to have.
2. The subjects in the two evaluation studies have more background knowledge about instructional design and development models and theory than the target users are expected to have.
3. In real design processes in a real environment designers often work in the context of a design team, and not individually as in these studies (although it is possible that one person is responsible for a part of the design process and works individually part of the time).

As expected, the results of an evaluation study with target users, reported in Section 9.2, shows that they do not have problems with terms or design activities that are specific for the field of simulator training, such as the design of training scenarios. Furthermore, the results of this study indicate that an extensive background in instructional design and development theory is not required to use the MASTER-TPD method, at least not when experienced instructional designers are present to answer questions about the method and the tool, as was the case in this study (see § 9.2.3).

The results of the case studies presented in Section 9.3 indicate that working in a team makes the process more complicated and more iterative. One way to deal with this kind of complications is to apply a systematic instructional development method in workshops with all stakeholders and with instructional design experts as facilitators, as described in Section 9.3.2.1. The facilitators then take over important aspects of the design task, such as managing the design process and the communication within the team (see § 9.3.2.6). When the team does not always work together in the same place additional facilities to support communication and co-ordination are proposed (see § 9.4).

The studies presented in this chapter do not systematically investigate the differences between working individually and working in a design team. In Section 3.2.3, I discussed different kinds of information that can influence the design process (see also Figure 10.1). From a different perspective this information will have to be provided by people in different roles. Based on the results of the five case studies, Table 9.1 gives an overview of roles that can be involved in the design of simulator specifications:

Table 9.1: Roles that can be involved in the simulator design process.

Roles
Instructor
Domain expert
Task analyst
Instructional designer
Instructional design and development expert
Trainee
Project leader
Expert in simulator training
Technical expert
Manager of the school
Controller/financial expert
Developer of CBT, books, etc.

People fulfilling these roles either have information themselves, or know how and where to get it. Financial experts, for example, may not know how much financial resources are available for the specification, acquisition and use of a (future) training simulator. They will know, however, who decides about budgets and which procedures have to be followed to claim and allocate budgets. Some of the roles will be fulfilled by people within the design team, others will be outsiders. Often, one person will have several roles, e.g., instructor and domain expert, or instructional designer and expert regarding instructional design theory. For other roles more than one person may be required. For example, several domain experts with different fields of expertise might be involved, or representatives of different levels of management. With all these different kinds of information and roles involved, it is obvious that designing simulator specifications requires a lot of communication and communication is likely to cause iteration. Therefore, it is relevant to study communication and the relationship between communication and iteration. I will come back to this issue in more detail in § 10.5.

Chapter 10
Conclusions and discussion

In Chapter 10, I summarise the answers to the research questions described in Section 1.4. Subsequently I propose a list of triggers for iteration and describe how different facilities can contribute to support the management of an iterative design process. Then I discuss to what extent the results of the research described in this thesis can be generalised to other settings, and give some directions for further research. I conclude with a brief reflection on the research presented in this thesis.

10.1 Conclusions regarding the research questions

In Section 10.1, I summarise the answers to the research questions described in Section 1.4. Note that in the context of simulator training, the terms trainee (rather than student or learner) and instructor (rather than teacher) are used (see § 1.2.2). In line with this, the MASTER method for the specification of training simulators that I have used for evaluation studies uses the terms training goal (instead of learning goal) and training activity (instead of learning activity). To avoid confusion I use the same terms in this thesis when I refer to settings where training simulators are used, even though the terms learning and learner might be more correct from a theoretical point of view (see § 1.5 and § 4.1).

Below the research questions are addressed one by one:

1. Which factors determine the (kind and level of) fidelity of training simulators?

From an educational point of view the minimum level of fidelity should be determined by the training goals that are covered with the training simulator (see § 2.1). For those tasks or task aspects that are trained with the simulator, at least functional fidelity should be guaranteed: the stimulus and response elements should be the same as in the real environments, so that trainees learn the behaviour that they need in operational tasks under similar conditions (see § 1.2.2). Limited functional fidelity restricts the range of tasks or task aspects that can be trained with the simulator. That means that trainees will not achieve all training goals at the end of simulator training. This is not necessarily a problem if there are enough opportunities to train for the rest with the operational system.

Only in three of the 44 simulators in the field study is there evidence that the specifications for the simulator have been deduced from training goals (see § 2.3.1.3). There are strong indications that specifications for training simulators usually focus on technical factors and physical fidelity, or that simulators are acquired (or delivered with the operational system in a package deal) without further specification of training. The results of the field study cannot prove this hypothesis, since documentation regarding the original specifications and the specification process was almost never available. However, experience, discussions with stakeholders and descriptive studies from other authors all point in this direction (see § 2.4.1 and § 2.4.4).

2. Do training simulators have the necessary facilities to support training and instruction?

Training simulators can incorporate a range of facilities to support training and instruction that are not available in the operational system: facilities to measure and register the trainees' performance, to provide demonstrations and (remedial) instruction, to provide augmented cues that speed up the learning process (and are gradually removed), to detect errors, to provide guidance and feedback, to adapt training to individual trainees, etc. The simulator can take over parts of the instructor's task, for example by providing (a part of) the feedback automatically, and by supporting the instructor in other tasks, e.g., by facilitating the monitoring and assessment of trainees.

The field study presented in Chapter 2 shows that this kind of facilities is not well developed in many training simulators. Facilities to support training and instruction are often poor and a heavy task remains on the shoulders of the instructors, whose support facilities on the Instructor Operator Station (IOS) are often quite limited too (see § 2.3.4.3 and § 2.4.2). The reason is probably that these issues did not get enough attention during the design of simulator specifications. This is not surprising if specifications are not derived from training goals (see above). Another reason could be that those responsible for the specification of training simulators often do not have enough knowledge about training and instruction, and the possible advantages of training with simulators (see § 2.4.4 and § 3.2.1). With extra facilities to support training and instruction the effectiveness of simulator training can be considerably improved with, compared to fidelity issues, relatively small investments.

3. Which issues should be addressed during the design of functional specifications for training simulators from an educational point of view (product requirements)?

A part of the issues that should be addressed in functional specifications for training simulators concerns the (part of the) course that will be covered in the future training simulator: simulator coverage, integration in the course and the characteristics of the target group(s) of trainees. Other issues concern the training process itself: individualisation of training, instruction and assessment. Finally, some issues concern knowledge management and instructor support: facilities for instructors, quality assurance and evaluation of training, editing facilities for training programmes and scenarios, and documentation of specifications and the specification process. More detailed descriptions of these issues can be found in Section 2.4.3.

4. How are functional specifications for training simulators designed?

Design tasks are ill-structured and inherently iterative problems. This is also true for the specification of training simulators: the domains are complex, every project is different, it is a long and time-consuming process, there are different kinds of -often contradictory- demands that can change over time, etc. (see § 3.2.1). The outcome is not only determined by what would be best given the tasks to be learned and the target trainees. More practical complications, such as conflicting interests, limited resources, personnel changes and timing problems also play a role. Different kinds of expertise are required for the design of functional specifications, such as knowledge about the operational domain, of the technological possibilities, of the market and expertise regarding training and instruction. The latter is often brought in by domain experts who have become instructors and course designers. These probably have experience with simulator training (both as trainees and instructors), but not with the specification of training simulators. They also do not have an extensive background in instructional design and development theory (see 3.2.1).

Thus, the people responsible for the design of functional specifications for training simulators do often not have enough resources, expertise and experience to execute a thorough analysis of training goals and alternative training solutions (see § 2.4.4). Moreover, they will have to design preliminary specifications with incomplete and insecure information, e.g., when they have to write a needs statement for a training system while the operational system is not yet acquired or fully developed. They will have to review and alter their decisions frequently when new information becomes available (see § 3.2.3). Reacting to these 'disturbances' can make the design process chaotic, but not reacting to them will certainly lead to solutions that are out of date or too expensive, and to the specification of a training simulator that does not fulfil the training needs or will not be accepted in the organisation. Waiting until all information is available is also not an option: training needs to be implemented and available before the operational systems are delivered. Functional specifications for the required simulators and other instructional products are necessary at an early stage to claim the necessary financial resources for their acquisition (see § 3.2.1).

Based on the descriptive research about the design of specifications for training simulators and other instructional products, presented in Section 3.2, I conclude that the task of designing functional specifications for training simulators has four main aspects (see § 3.2.3):

I) Gathering information
Collecting information is obviously important during the analysis of the design problem. However, not all information will be immediately available and it is important that information is kept up-to-date during the whole design process. All information needs to be organised and stored, and ready for review during the design process and afterwards.

II) Solving the design problem
Solving ill-structured design problems involves problem analysis and problem solving, which may be sequential or overlapping processes (Rowland, 1993). Sub-tasks include, decomposing the problem into sub-problems, designing possible solutions for a sub-problem, selecting the most suitable alternative, integrating sub-problem solutions, conducting formative evaluations and checking a solution against the demands and constraints, etc.

III) Managing the design process
For ill-structured problems, such as design problems, there is no algorithm that defines how to apply instructional design knowledge and guarantees a solution. Every design process is different. It is not possible to prescribe a fixed sequence of actions or tasks. The designers will have to organise and monitor the design process themselves, e.g., plan activities, reflect and evaluate, decide to go more into detail, to go to the next design activity or to go back and review an earlier decision.

IV) Communication
The design process takes place in an organisational setting. The task of designing functional specifications for training simulators is usually not the responsibility of one person, but of a team of designers. Other stakeholders may be less directly involved, but can still influence the design process, especially when they have power over resources. Thus, the design task also involves communication within the design team and with other stakeholders.

Instructional development models support the management of the design process. Ideally, specifications for instructional products should be based on what trainees need to learn, and how they can learn it most effectively and efficiently (cf. Figure 1.1). Then, the instructional designer can make a well-considered choice of what to cover with simulator training, and specify the appropriate level of fidelity (see § 2.4.1). In order to specify a good instructional overlay, i.e. facilities to support training and instruction, insight into the future training method and programme is required. Instructional System Development (ISD) methods (see § 3.3.3) prescribe a sequence of steps or activities to reach this purpose and help to organise and manage the instructional design process. However, the design of functional specifications for training simulators is an inherently iterative process. Instructional designers have to deal with all kinds of events that 'disturb' the ideal design process, and with the uncertainty that is caused by the incompleteness and unreliability of information (see § 3.2.3). Experienced instructional designers have the problem solving skills necessary to organise, monitor and adapt the design process (see § 3.2.2.1 and § 3.2.2.3). The designers of functional specifications for simulators are usually not experts, however, and descriptive research shows that novices are not very good at managing their own design process (see § 3.2.2.3). Thus, current ISD methods permit iteration, but do not sufficiently support this aspect of the design process for novice designers: it is not clear how and when iteration needs to occur and it is up to the designers to organise and manage iteration during the design process. Rapid prototyping development models provide ideas to deal with iteration in a different way (see § 3.3.5). In practice, iteration will also to some extent be enforced by acquisition procedures, which structure the design process in time (see § 3.3.6).

5. Which aspects of the design process need to be supported (process requirements)?

In the answer to the fourth research question (see above) I concluded that the task of designing functional specifications for training simulators (and other instructional products) has four main aspects: gathering information, solving the design problem, managing the design process and communication. Designers need facilities to support all four aspects (see § 3.4). First of all they need support for information gathering:

A) Facilities for handling input information
Facilities to, for example, organise and store information, to provide an overview, and to retrieve information when necessary, in an appropriate format.

B) Facilities for modelling the problem and the solution(s)
Facilities to, for example, represent the results of different kinds of analyses, the decomposition into sub-problems and solutions to (sub-)problems; and facilities to store and compare alternative solutions to design (sub-)problems.

C) Facilities for documentation
Longer term storage of the design results and the design process (see § 3.3.7.3) concerns questions like: Where did information come from? Who took certain decisions? What were their reasons? Which alternatives were considered?

Secondly, instructional designers need support for solving the design problem:

D) Strategic support
Advice on how to execute the different activities in the design process, such as advice about where information can be found, how it can be easily collected and organised, and who could give advice on specific issues.

E) Heuristics and guidelines regarding instructional design
In this context, knowledge about using simulators for training purposes and about simulator training programmes. This knowledge is currently dispersed over peoples' personal experience and research literature in different areas, e.g., hardware and software development, instructional design theory and learning psychology. Guidelines should be linked to specific steps or activities in the design process.

F) Decision making support
At important decision points specific support can help designers to deal with multiple and contradicting requirements and constraints. Decision support can take the form of automated advice or facilities to organise information and arguments and compare alternatives (see § 3.3.7.2).

Thirdly, instructional designers need support for managing the design process:

G) Procedural support
A prescriptive method can help to identify important activities and decisions in the design process and make sure that no important issues are forgotten. The method will probably have to be adapted to fit the organisational context and the problem at hand.

H) Support to deal with iteration
Training simulator specification is clearly not a linear process (see § 3.2.3). For different reasons it will be necessary to go back and execute (parts of) some design activities again, for example to review a decision or to elaborate an analysis in more detail. This is called iteration (see also § 5.4.1). Designers iterate to develop solutions in more and more detail, and revisions or adaptations may be necessary for different reasons. Related to this is the issue of dealing with uncertainty: designers will have to make assumptions and take preliminary decisions based on incomplete information, and they will have to keep on checking whether these

decisions are still valid given the most recent information and insights. For possible ways to support dealing with iteration see Section 5.4.2 and the discussion in Section 10.3.

I) Support for quality control
Quality control is important for the organisations' managers, not only to ascertain that the designers have done their work well, but also to be able to claim resources and to prove that the acquisition of a new simulator is necessary to solve existing training needs.

Finally, instructional designers need support for communication:

J) Support communication within the team
Facilities to, for example, exchange information, and to divide and plan the work to be done.

K) Support communication with stakeholders
Facilities to, for example, retrieve important input information, to involve stakeholders in the design process, and to acquire the approval necessary for the acquisition and successful application of new training simulators.

In the literature different kinds of prescriptive theories and methods are described that all support one or more of aspects of the design task (see § 3.3.9). None of them, however, provides coherent support for all four aspects. Designers are tempted to follow a strictly linear method, or to react intuitively and not use a systematic approach at all. The challenge is to offer a clear structure and a systematic method, while at the same time allowing flexibility and supporting iteration during the design process. Advice and guidance should be available when needed: just in time and just enough.

6. To what extent does the MASTER method fulfil the product requirements defined in Question 3 and the process requirements defined in Question 5?

An ISD-based method for the design of functional specifications for training simulators has been developed in the context of the European defence research project MASTER (EUCLID, RTP 11.1). The MASTER method is described in Section 4.2. All the issues that need to be covered in training simulator specifications, defined under Question 3, are addressed by the MASTER method (see § 4.3.1). However, the MASTER method does not explicitly address the instructors' tasks and needs from the instructors' point of view. With regards to the process requirements defined under Question 5 (see § 4.3.2), the MASTER method structures the development process (Aspect G) and provides a framework for organising and storing all the data that designers collect (Aspects A, B and C). It also makes it possible to provide concrete, context-specific strategic advice (Aspect D) and heuristics guidelines (Aspect E) to make up for possibly lacking instructional design and development knowledge at the moment that it should be applied (although the content of guidelines and libraries has not been developed within the MASTER project). It does not provide specific facilities to support decision making (Aspect F), quality control (Aspect I) or communication (Aspect J and K). See Section 9.4 for a short discussion about facilities that can support communication within a design team.

Although the MASTER method enables an iterative design process, it still gives a rather linear perspective: the method prescribes a sequence of steps to be taken in a pre-specified order. It is possible to go back to previous steps, to break up the design problem into subproblems and/or to partly develop and compare alternative solutions. It is, however, up to the user to decide whether this is necessary and how it should be done. Iteration is facilitated by the clear input/output relationships between steps and by keeping intermediate results and the user's notes available throughout the design process, but it is not actively encouraged or supported. The MASTER method does not, in other words, sufficiently support managing iteration during the design process (Aspect H). Research Questions 7 and 8 focus on this aspect. Another point of criticism was that the MASTER method had not yet been tested with a realistic case. This point is addressed by Question 9.

For my research regarding the remaining research questions I have used a part of the MASTER method: the Training Programme Design phase (TPD). For the purpose of evaluation studies the TPD was further elaborated and implemented in a prototype tool. For this version of the method the term MASTER-TPD method is used in this thesis. For the evaluation studies a limited set of guidelines and some strategic support has been developed for each step, but since this was not the focus of my research no claims are made about their validity or completeness. In the prototype MASTER-TPD tool specific support for managing iteration has been implemented (see under Question 7 below). In the context of specifying training simulators with the MASTER method I define iteration as: going back to a previous step, or, in other words, to a design activity that has already been executed. There are different ways to iterate and different reasons for iteration (see § 5.4.1).

7. How can designers be supported during the iterative design of simulator training programmes?

The optimal way of designing training simulator specifications depends on characteristics of the specific design task and the designers. It will be different for every project: if, when and how often iteration is desirable depends on many factors, including the complexity of the domain, the capacities and the preferences of the designers, the availability of information, changes in the available resources, organisational factors, etc. Training simulator specification is, like all instructional design problems, an ill-structured problem (see § 3.2.2.2). The management of the design process cannot be automated or formulated in an algorithm: it is not possible to prescribe a fixed sequence of design activities or fixed moments for iteration. Therefore, the only way to encourage and support an iterative design process is to help users to manage their own design process better. For this kind of support, I use the term process-oriented support. Process-oriented support should help users to keep track of what they are doing and to decide what to do next. For example, by making the relationships between different steps or activities in the design process explicit, helping users to recognise triggers that might necessitate iteration and making it easy for them to go back to previously executed steps or activities in the design process.

Documenting not only the (intermediate) design products but also the design rationale is important in an iterative design process for several reasons:
- to maintain consistency,
- to (re)use all the information and arguments that are available (see also § 3.3.7.3),
- to explain or defend design decisions, and
- to make it possible that somebody else takes over (when designers are working in a team).

Especially in an iterative process, it is important that designers make notes immediately when they make decisions, instead of at the end of the design process (Boy, 1999; Boy and Barnard, 2003). When designers go back to review earlier decisions, they need to know the reasons for the initial decision, the assumptions that it was based on, the alternatives that have already been considered, etc. Notes are even more important when more than one person is involved. In this case, designers may have to review or elaborate work that was done by another member of the team. There are indications that novices are not proficient in representing design problems to themselves and others, and are not aware of why they made certain decisions (Kerr, 1983; see also § 3.2.3.3). Therefore, process-oriented support should include helping users to make adequate notes with each design activity.

In the prototype MASTER-TPD tool four types of process-oriented support have been implemented (see § 5.4.2 for a more elaborate description and § 5.4.3 for examples):

1. Hints on the Notepads attached to each step to help and encourage designers to make adequate notes.
2. Process-oriented guidelines to help users to manage the design process.
3. Warnings when events that should trigger iteration can be automatically recognised.
4. Definition of two design cycles: explicit descriptions for each design activity to help users to decide how far they should go into detail to get to global specifications, and later on to get to more detailed specifications (cf. using the method at different levels of detail, see § 4.2.2).

Process-oriented support should help the users to take process-oriented decisions, supporting a methodological and systematic, and at the same time iterative, design process. I will discuss support for iteration in more detail in Section 10.3.

8. How is iteration used in the design process?

The research presented in this thesis does not completely answer this question, but it does provide a number of important insights. First of all, both the analysis of descriptive literature and discussions with designers of simulator specifications make clear that iteration is inherent to the design process. When the domain is rather small and all the relevant information is immediately available, the design of training simulator specifications might be an almost linear process. Usually, however, this is not the case (see § 3.2.1). Designers will have to take preliminary decisions based on incomplete and insecure information, and they will have to review and alter their decisions frequently when new information becomes available. Table 3.1 in Section 3.2.3 gives an overview of the different kinds of information that can influence the design process.

There is also a more fundamental reason for iteration: in an ill-structured design problem the sub-problems are not unrelated and can, therefore, never be solved in isolation (see § 3.2.3). Moreover, it is impossible to know beforehand which ideas will lead to a good solution and which will not, because:

- solutions have to be partially developed before they can be checked against the problem definition, and they may have to be adapted or discarded when they do not optimally fulfil the demands,
- good solutions for sub-problems may not be feasible anymore when they are combined with solutions to other sub-problems, and
- practical factors like changing resources or interruptions by other stakeholders may actually change the problem definition during the development process.

This means that the design process can be only partially planned beforehand. Iteration is unavoidable, but it is not necessarily bad. On the contrary, it should improve the final result of the design process (see also § 5.4.1). However, dealing with iteration during the design process is difficult for novices.

The results of both studies confirm that iteration is inherent to the design process: all subjects iterate. In the first study, there were large differences in the amount of time that subjects spent on the design task (see Table 6.5). However, there was no relation between time on task and the number of iterations (see Figure 6.3). In the second study, the time on task was controlled, and it less was than the average time on task in the first study. However, the average number of iterations is more or less the same in both studies (see Table 7.6). It is possible that the interventions that were introduced in this study stimulated the subjects to iterate more than they would have done otherwise. In both studies some subjects made clear that they did not like to iterate, using phrases such as 'having to do the same thing again' or 'now I can throw away some of the work that I have already done' (see § 6.4.3.2 and § 7.4.2.2). Other subjects, however, have no problems with iteration and, in fact, the possibility to iterate is also mentioned as a strong point of the MASTER method, especially in the second study (see § 8.3.1.2).

The evaluation studies do not show a relationship between the quantity of iteration and the quality of the result of the design process. The number of iterations varies considerably between subjects, and there is no relationship with the quality of the resulting training programme designs (see Figures 6.2 and 7.1). Apparently, different design styles -with more or less iteration- can lead to good results. Some subjects work out a good concept and then work through the method step-by-step, going back to previous steps only when it is necessary. Others come to a good design by going back and forth between steps more often. Maybe they leave more decisions open, go to the next step(s) to elaborate the design partly, get new ideas, and then go back to work on the initial concept again, etc. Inspection of the subjects' notes shows that the number of iterations does not correspond to the amount of rethinking or reviewing done during the design period. The results of both studies show that frequent iteration can be part of an ineffective design style as well (see § 6.4.3.3 and § 7.4.2.1), perhaps indicating a lack of overview. Weak designers will probably make more errors and will have less insight into the possible consequences of early decisions. Good designers might have to iterate less to repair errors, possibly because they make a more thorough problem analysis beforehand. On the other hand, good designers may use iteration for another purpose: to check whether they have designed the most optimal solution or to consider possible alternatives. This would be in line with the findings of descriptive research which show that experienced designers do indeed spend more time and effort on a problem analysis, generate several alternative solutions, and do not commit themselves until later on (see § 3.2.2.3).

In the evaluation studies there is no relation between the number of iterations and the time spent on help, not even with the time spent on the process-oriented guidelines that include specific advice regarding iteration (see Figure 8.2). In the second study, subjects were explicitly asked whether they had iterated because the guidelines advised them to (see § 7.4.2.2). Only one of the eight subjects answered 'yes, sometimes'. However, the reasons for iteration that the subjects mention themselves are very similar to the ones that are specified in the guidelines:
- new insights or discovering later on that the initial choices were not the best,
- new information e.g., from answers from experts by e-mail, and
- repairing mistakes caused by misunderstanding the purpose of a step.

Furthermore, most subjects made adequate notes (see § 8.2.2). Tables 8.1-A and 8.1-B show that all subjects have looked at guidelines, and inspection of the subjects' notes shows that they have used them. Thus, although the evaluation studies do not provide data to prove it, the process-oriented support has probably helped subjects to manage iteration during the design process.

Finally, the results of case studies in real projects with target users presented in Section 9.3.2 indicate that working in a team makes the design process more complicated and more iterative. There were almost continuously 'interventions' caused by discussions between team members, or by new information brought in by one of the team members or by others who were not directly involved. In these case studies another approach was used to deal with iteration. The design activities were executed during workshops with the most important stakeholders and with experienced instructional designers as facilitators (often including the author of this thesis). During these workshops the different aspects of the design task were split up between the participants and the facilitators. The participants were responsible for gathering information and for solving the design problem. The facilitators were responsible for the organisation and management of the design process, including the aspect of dealing with iteration, for storing all information and decisions (in the form of a report) and, to a large extent, for managing the communication between the participants.

In conclusion, iteration is inherent to the design process and it can -but does not always- improve the result. Iteration can be used in different ways, depending on environment-related and subject-related factors. There are different reasons to decide to iterate or not. As long as we do not have a good understanding of what those reasons can be, it will remain difficult to support this aspect of the design process. I will come back to this in Section 10.2.

9. Can novice designers design simulator training programmes with the MASTER-TPD method in an adequate way?

The evaluation studies presented in this thesis show that novice designers can design simulator training programmes with the MASTER-TPD method with appropriate support, i.e. a tool that guides them through the method step-by-step, an explanation of the method and tool, and experts that are available for advice. In the first study the quality rankings of two instructional design experts correlate highly with the rankings of two instructors/Subject Matter Experts (SMEs) and the marks given by lecturers (see Tables 6.3 and 6.4). The two instructors/SMEs commented that the quality of the training programme designs varied a lot, but that the best ones would definitely be suitable to be used, as they are, as input for training simulator specifications. In the first study the subjects worked at home, at their own pace. The results show that the time that subjects spent on the design task is related to the quality of their results. In the second evaluation study this effect was prevented by letting subjects work for a fixed amount of time, under supervision. Even though the subjects in the second evaluation study had less time then those in the first study, all of them were able to complete the assignment. According to three experts all designs were acceptable, and of at least of the same level of quality as the ones of the first study (see § 7.4.1.1).

The subjects responses to questionnaires confirm that they liked working with the MASTER-TPD method and did not find it too difficult (see Tables 6.6 and 7.5). Almost all subjects are positive about the structured approach proposed by the MASTER-TPD method, although some of them also see some limitations of such an approach (see Tables 8.6, 8.7-A and 8.7-B). The usability of the MASTER-TPD method is further confirmed by the fact that the planned procedures could be executed without major changes in both studies.

A striking difference between the two evaluation studies was that the subjects in the first study appeared to be very insecure and asked for feedback continuously. In the second study this was not the case. The most probable reason for this difference is that for the subjects in the first study failure meant failing an important course and not receiving the study credit points for that course (see § 6.4.1.4). The subjects in the second study were either paid for their participation irrespective of the results, or participated in work time. In real design projects in real environments, failure will have large consequences. Therefore, similar feelings of uncertainty are expected. In fact, a short evaluation with real target users indicated that they feel a great need for confirmation of the correctness of their decisions (see § 9.2.3).

In many respects the situation during the two evaluation studies described in Chapters 6 to 8 was similar to the situation that the MASTER method was developed for: novice designers specifying a training simulator. The case that was used was realistic in size and complexity (see § 6.2). There are, however, some differences that may have consequences for the validity of the results: the subjects in these studies have less knowledge about the domain and simulator training in general, and more theoretical knowledge about instructional design and development models than the target users are expected to have. Also in real design processes in a real environment designers often work in the context of a design team, and not individually as in these studies (although it is possible that one person is responsible for a part of the design process and works individually part of the time).

There are no indications that a lack of knowledge about the domain prevented subjects from fulfilling their design task (see § 6.4.1.4, § 7.4.1.1 and § 8.2.3). They had some problems with simulator-specific terms and activities such as the design of training scenarios, but these could be solved with some extra examples and explanations. It was not possible to compare the subjects' designs to designs made by domain experts, because these were not available. However, the two instructors/SMEs that judged the subjects' training programme designs in the first study were enthusiastic, especially about the best ones (see above). Of course, designers will always need information about the domain, which can be provided in documentation and interaction with domain experts during meetings and/or by e-mail (as in the studies described in Chapters 6 to 8), or by involving domain experts as members of the design team (as in the case studies described in § 9.3.2).

To check the validity of the two evaluation studies reported in Chapters 6 to 8, two studies with real target users have been executed. These studies were limited in time-scale, and rather informal because they were executed in the context of other projects. As expected, the real target users had fewer problems with simulator-specific terms and activities. The results also indicate that an extensive background in instructional design and development theory is not required for using the MASTER-TPD method, at least not when experts are present to answer questions about the method and the tool, as was the case in this study (see § 9.2.3). Some caution is necessary with regards to this conclusion because, due to the limited time scale, it was not possible to check the quality of the resulting training programme designs. Therefore, it seems advisable to include an instructional design expert in the design team or ensure that the team has access to such people for support.

Finally, in the studies presented in this thesis the differences between working individually and working in a design team have not been systematically investigated. However, the results of case studies with a method similar to the MASTER method indicate that working in a team makes the process more complicated and more iterative (see § 9.3.2.5). Working in a team also introduces a social aspect to the design process. A systematic method can help to structure the design process, but other support is required to manage group dynamics. A combination of a systematic and relational approach seems to be required (see § 3.3.7.1). When the team does not always work together in the same place, additional facilities to support communication and co-ordination are required (see § 9.4).

In summary, instructional development methods and tools are available. An example for the area of designing training simulator specifications is the MASTER method. Users, also novices, who learn how to use this method, and spend and adequate amount of time and effort on the design task, get to acceptable results without major problems (see the discussion in § 7.5.1). No doubt an expert instructional designer specialised in the design of training simulators could do better (see, e.g., the comments of Rater 2 in § 7.4.1.1), but experts are often not available (see § 3.2.1). The evaluation studies also provided more specific information about the difficulty of individual steps of the MASTER-TPD method (see § 8.3.2 for an elaborate discussion).

10. Do subjects use the different forms of support that are offered?

During both studies subjects used all types of help information offered by the tool, spending least time on information about the MASTER-TPD tool and more time on information about the MASTER-TPD method and on the guidelines (see Table 8.1-A and Table 8.1-B). Closer inspection of the subjects' notes showed that all subjects regularly mention guidelines, which means that they have read and used them. In the second study subjects had less time to complete the same assignment (see § 7.3.2). They did not, however, spend less time on help. This confirms the general impression during both studies that the assignment was so difficult for the subjects that they needed all available information to complete it. There are no indications of problems with accessing help information in the tool and there are no clear relationships between the amount of time spent on help and the quality of the subjects' training programme designs, the number of iterations, and the quantity of notes that subjects make. Therefore, the hypothesis is that differences in time spent on help are either caused by the way time was measured (see § 8.2.1) or, more probably, by differences between subjects in the amount of time they needed for (re)reading help information.

Documenting not only the (intermediate) design products but also the design rationale is important in an iterative design process (see under Question 7). The expectation beforehand was that it would be hard to convince users to make notes. This proved to be not true during both evaluation studies: all subjects made notes, and often -but not always- those notes were sufficient to follow the subjects' reasoning. The quantity of notes varied a lot and, although there is no linear relationship, there are indications that subjects who performed better made more notes (see Figure 8.3), especially in the second study. This is not very surprising since the judgements about the quality of the training programme designs were influenced by how well subjects explained and defended their design decisions (see § 6.4.1.1). The fact that there is no clear relationship between the quantity of notes and the amount of time spent on different help forms can mean a number of things. It is possible that subjects were already motivated to make notes or that the data on the time spent on help were not reliable enough to show any effect (see § 8.2.2). Probably, however, all subjects got enough information about making notes from the general advice to make notes in the descriptions of the MASTER-TPD steps, the more explicit statements in the Hints on the Notepads and from whatever amount of time they spent on reading the process-oriented guidelines. Note that it is possible that the subjects in these evaluation studies made more notes than real target users might because they are students who are used to writing assignments. There are indications that engineers, for example, have much more difficulty to make notes that are understandable for non-experts (Barnard and Boy, 2003). Target users might also be more reluctant to make notes because they can be held personally responsible if wrong decisions can be traced back to them. The study with real target users, reported in Section 9.2, was too short to evaluate this aspect, but the participants did say that making notes is, in their opinion, very important. In a real design process in a real environment, the users' notes will also be used by others, i.e. other members of the design team and/or people responsible for quality assurance. These people will no doubt ask for clarification if the notes are incomplete or unclear; they might even find the hints and guidelines that were implemented for the designers in the MASTER-TPD tool useful to check whether the notes are complete.

In both studies the subjects sent questions to experts by e-mail (see Table 8.3-A and Table 8.3-B). See Table 8.4 and the discussion in that section for the kinds of questions that were asked. In the first study the good performers asked more questions, but in the second study there is no such relationship (see Figure 8.6). In both studies experts were not always able to answer immediately. This frustrated some subjects who claimed that they could not continue with their task without the answers to their questions (see § 8.2.3). This can be a sign that they found it hard to deal with uncertainty and incomplete information (see also the discussion above about insecurity).

The subjects are quite positive about all forms of support (see Table 8.5). In the first study the support facilities were also often mentioned as a strong point of the MASTER-TPD method and tool (see Table 8.7-A and Table 8.7-B).

10.2 Triggers for iteration

In order to develop support for managing iteration during the design process it is necessary to understand in which circumstances iteration needs to occur, and how these circumstances can be predicted or recognised. For this purpose, I propose a list of triggers for iteration in this section.

10.2.1 Background

In Section 5.4.1, I reported how van Wagenberg (1992) differentiates between four types of iteration that can also occur during instructional design processes:

1. Iteration as repetitive activity: repeating the same design activity.
2. Iteration to correct errors: reviewing the concept design and correcting errors.
3. Iteration to improve the design: fundamental revisions to get to a better design.
4. Mutual iteration: this form of iteration happens when designers decide to execute two separable activities in parallel; it can also be seen as a form of decomposition.

Reviewing these different types of iteration, the first conclusion is that some types of iteration are desirable: iteration to improve the design is obviously a good sign (provided that the required time and resources are available). Repetitive iteration can make the design process easier to manage or more efficient: designers can decide to decompose the problem into sub-problems or to work at different levels of detail. Similarly, an organisation can decide to execute different design processes in parallel, e.g., the specification of a simulator and CBT for the same course (mutual iteration). For these kinds of iteration, the difficulty lies in the decision about the amount and timing of iteration: which design activities are important to improve the quality of the design? To what level of detail should activities be executed at different moments in the design process? On the other hand, the need for iteration to correct errors often occurs unexpectedly. In an efficient design process making errors, and having to correct them later, should be avoided as much as possible. For this reason, expert instructional designers execute a thorough problem analysis before they start designing solutions (see § 3.2.2.3). Not all errors can be avoided, though, especially when not all information is available from the start and when constraints and demands can change during the design process (see § 3.2.3). Thus, designers need to be able to recognise and correct errors as early as possible.

In the context of designing simulator training programmes the procedures for the acquisition of (expensive) instructional products (see § 3.3.6), that are obligatory in large organisations, are an obvious trigger for iteration as a repetitive activity. In the evaluation studies this was mimicked by defining global and detailed design phases. Iteration to correct errors will occur because the designers are likely to get new information about the domain that they are working with and because the project's constraints often change during the design period (see § 3.2.1). Iteration to improve the design can occur when new arguments or ideas come up in discussions with others, either stakeholders or peers, but also when the designers get new ideas or insights themselves. Based on the results of the second evaluation study, I concluded that new insights can be caused by different events: designers can discover errors, they can discover that input for later steps or activities is missing or not sufficiently elaborated, or they can get new ideas unexpectedly (i.e. the unexpected 'Aha' or 'Eureka') or based on their later work (see § 7.4.2.3).

10.2.2 Seven triggers for iteration

Some triggers for iteration during the design of training simulator specifications were described in Section 7.2. Based on the results of the second evaluation study (see § 7.4.2.3), this list has been redefined and elaborated to the seven triggers for iteration depicted in Figure 10.1. The first four triggers are caused by, or evolve from interaction with the design process itself. They are: the discovery of missing input, the need to repair errors, new insights based on work later on in the design process, and new ideas of the designer(s). The other triggers originate from outside. New information that is relevant to the design process becomes available, or other people bring in new opinions or arguments that the designer(s) were previously unaware of (see for different kinds of information that can influence the design process Table 3.1). A third trigger from outside can be the acquisition procedures that are enforced in many large organisations. These procedures influence the design process through feedback on obligatory intermediate products, such as milestone documents containing preliminary versions of the training simulator specifications, as shown in Figure 10.1:

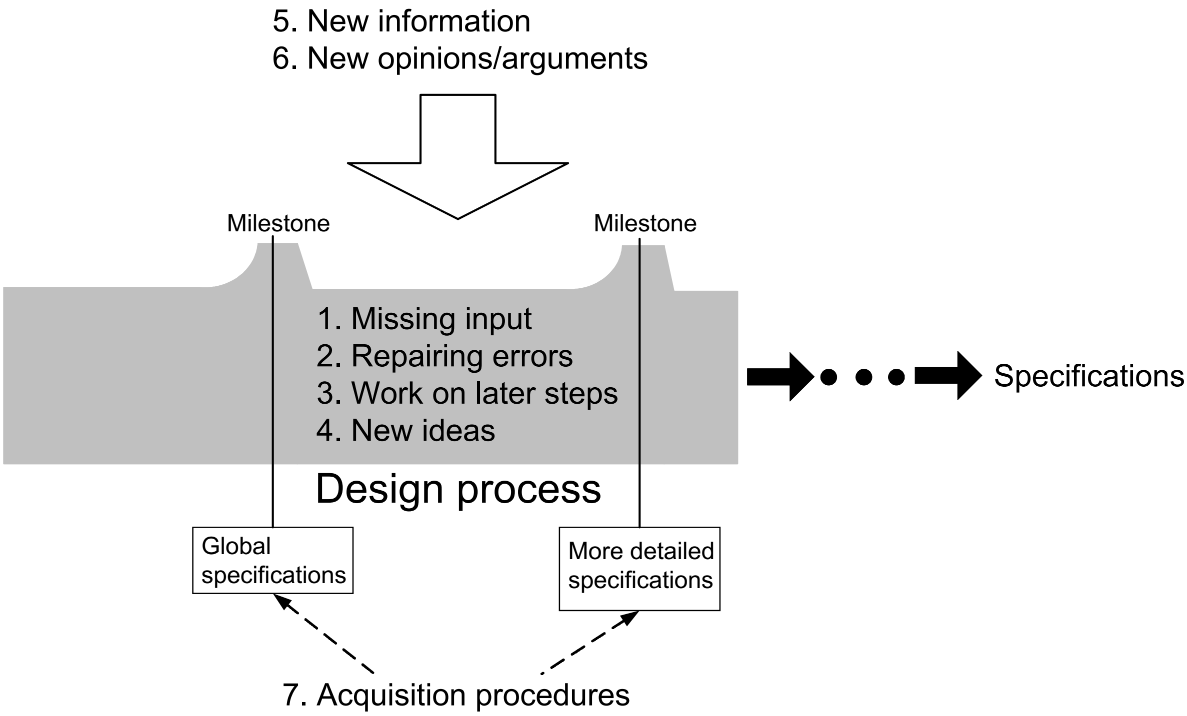


Figure 10.1: Seven triggers for iteration.

The different triggers for iteration are described below, with some examples and a description of characteristics that can be used to develop different kinds of support to handle the different triggers (which will be further discussed in § 10.3).

Trigger 1: Discovery of missing input
One trigger for iteration can be that the designers discover that they are missing necessary input that should have been available from an earlier design activity. For example: it is difficult to sequence training goals (in MASTER-TPD Step 1.2) when these are not elaborated in sufficient detail and/or information about the difficulty and criticality of the tasks is missing (in the MASTER method this information is expected to be available from the Training Needs Analysis). The key phrase here is 'in sufficient detail'. It is not always possible to foresee how much detail will be required. Designers have the choice between the risk of spending time and effort on detailed analyses or elaborations that will never be used, or the risk of having to iterate later on when they discover that a more global level is not sufficient.

This trigger is easy to recognise. It is possible to describe the input required for each design activity in the design process, and to provide guidelines that direct designers back to where they need to further elaborate their work if vital input is missing. When they go back, the designers might discover that they do not have enough information and need to consult others, e.g., domain experts or instructional design experts, or to collect new information from other sources (see Trigger 5).

Trigger 2: The need to repair errors
Another cause for iteration can be that designers discover that they have made an error earlier on. A good example of this showed up in the feedback rounds in the first evaluation study (see § 8.3.2.1): many subjects had formulated training activities in very general terms, such as 'study and practise' or 'perform real task' (in MASTER-TPD Step 2.1). Some subjects did not specify training activities for each training goal, but used one general definition for a whole

group. This lack of detail makes it difficult to decide which training activities should be covered with the future simulator and which not (in MASTER-TPD Step 2.2), and to define exercises or training scenarios (in MASTER-TPD Step 5). This problem was also found in the second study (see § 7.4.2.2).

Often this trigger is related to other triggers: errors can be discovered when designers evaluate the input required for the next design activity (Trigger 1) or they can be identified by others (Trigger 6 or Trigger 7). Dealing with errors can be supported in similar ways, e.g., by describing demands on the output, or by providing guidelines that encourage designers to ask advice from peers or experts. Another possibility is to include explicit decision points with checklists (as in the SLIM method, see § 9.3.1) or formal signing-off procedures at important points.

Trigger 3: New insights based on work later on in the design process
Sometimes further elaboration gives more insight into the consequences of earlier decisions. An example can clarify this: when designers are specifying training activities (in MASTER-TPD Step 2.1), they might realise that the learning principles or training strategies, that they have based their design on, do not really suit the type of training goals that they are working with. The MASTER-TPD method prescribes that, in this case, they should go back to make changes in the relevant step (i.e. in MASTER-TPD Step 1.1) and go through the intermediate steps to see whether this decision has consequences for their design, for example for the sequencing of training goals (in MASTER-TPD Step 1.2), etc.

It is not always possible to foresee if and when this trigger will occur. However, some activities in the design process are logically related, e.g., because one provides input for the other. Experienced designers are another source of information: they will be able to predict when this trigger is more likely to occur. Their input can be used to develop support, such as advice in the descriptions of design activities and in guidelines (see § 5.4.3.1 and § 7.2.2 for some examples). Another example is a warning that pops up, for instance, when it becomes clear that only a small number of training activities has been allocated to simulator training, advising users to check whether a simulator is really necessary (see § 5.4.3.3).

Trigger 4: New ideas of the designer(s)
Designers can get new ideas at any time. This kind of 'Eureka!' inspiration can be triggered by seemingly unrelated events. Unexpected new ideas are difficult to foresee and support. This trigger is related to creativity in the design process. Generating and maintaining several ideas makes the design process more difficult to manage and will probably cause more iteration. Presumably, however, it will lead to a better solution. There seem to be design activities where considering alternatives is especially useful, for example the definition of (innovative) training activities or the selection of (a combination of) instructional products. Design styles play a role as well. The results of the evaluation study showed that a large number of iterations can be part of an effective design style, but also of an ineffective design style. On the other hand, some of the best subjects iterated relatively few times.

There might be a relation with the experience of the designers. Descriptive research studies show that experienced designers reuse (parts of) successful solutions to other design problems, and that they always generate a number of alternative ideas, and do not commit to one particular solution immediately. Novice designers, on the other hand, commit to one solution early on, and do not seem to explore possible alternatives (see § 3.2.2.3). Experienced designers may also have a better overview of the design process, and thus more memory capacity for generating new ideas. They can probably plan their design process better and, in consequence, they might be better able to recognise other triggers for iteration and use those moments to try out new ideas as well, or they might even plan moments to contemplate and generate new ideas.

Trigger 5: New information
New information can trigger iteration, such as new information about the domain, e.g., about the tasks of operators, the missions that the operational system will be used for, the interface of the operational system, or the background of trainees that will be trained. New information can also concern the available resources and other constraints, e.g., changes in the budget that is available for acquisition of training systems, or the number of instructors that will be available for the delivery of training. The organisation's policies may change, dictating, for example, that field training should be minimised because of environmental pollution, or that training systems must be bought 'off-the-shelf' whenever possible. The problem is that designers might not know what information is available and that their colleagues may not always realise that such information is relevant for the instructional design process. Moreover, the risk is that designers are preoccupied with other factors, such as the instructional factors that are prominent in the ISD approach (see § 3.3.3), and never notice that new information about constraints should have influenced their design.

Designers can get new information in two ways:
- they can take the initiative to look for information (e.g., in documentation or by asking questions), or
- other people can provide new information.

It will probably not be difficult for designers to realise that they have received new information, but it might be difficult for them to decide when and how to look for information, to decide whether information is relevant and, if so, what they should do with it. Guidelines can indicate which kind of information can be relevant for specific steps or activities, and encourage designers to look for additional information and to check whether the information they have is still up-to-date (see § 7.2.2 for examples).

Trigger 6: New opinions/arguments from other people
Discussions with other parties involved in the design process can cause iteration as well. Experts or stakeholders not directly involved in the design team can bring forward arguments that the designers were previously unaware of. A domain expert, for example, might argue against simulator training because trainees never experience the stress of a war situation in a simulator. Or an instructor might argue for more controlled training scenarios because then it is easier to give effective feedback. Even when the designers do not agree with these arguments, they will have to spend some time to defend their decisions and explain why they have not changed their design.

This trigger for iteration should be easy to recognise, and it will in most cases be relatively easy for designers to decide whether arguments or opinions are relevant. Guidelines or other help forms can draw the designers' attention to moments when it may be opportune to consult experts, stakeholders or more experienced peers (see for examples § 7.2.2). Another possibility would be to explicitly plan brainstorm sessions or reviews at specific moments in the design process.

Trigger 7: Acquisition procedures
Since training simulators are expensive, their acquisition is usually regulated by procedures that enforce an iterative design process, for example by requesting milestone documents (see § 3.3.6). To obtain resources for the design and acquisition of training simulators, designers usually have to present and defend their choice of this type of instructional product early on. This means that they will have to take preliminary decisions based on quick, global analyses and incomplete information. As discussed in Section 5.4.1, the MASTER method can be used at different levels of detail for different purposes. The SLIM method, derived from the MASTER method, has been specifically developed for the first phase in the acquisition trajectory (see § 9.3).

Because it is defined beforehand, this kind of iteration is easy to support with guidelines, warnings and explicit descriptions of the design cycles. In fact, the end of a design cycle may be a good moment for a review, as proposed above, to evoke comments that can be used during the next design cycle.

10.2.3 Experiences with triggers

In the second evaluation study five interventions were executed to mimic the last three triggers for iteration (Trigger 5-7), i.e. those that can be initiated by the environment. Subjects also iterated in between interventions, which means they also went back to previous steps because of other reasons, e.g., when they missed input information or wanted to review an earlier decision. The experiences with the interventions are briefly summarised in this section. For a more detailed discussion see Section 7.4.3.

The strongest intervention was the explicit scheduling of two design cycles (to mimic acquisition procedures, i.e. Trigger 7). Subjects were asked to go through all the steps quickly on the first day to make a global training programme design and then to start again at step one and go through all the steps in more detail. Inspection of the log files shows that all subjects understood the idea of two design phases and applied it. The global design phase, however, was sometimes shorter or longer than planned (see Table 7.8). This type of iteration was strongly encouraged by the explanations of the global and detailed design phases, and in the Manual and the guidelines. In practice, however, acquisition procedures also oblige designers to make several versions of their specifications and send them to superiors and/or stakeholders for approval (see § 3.3.6).

Trigger 6 (new ideas/arguments from others) was introduced in the form of an opportunity to review each other's work (and get comments on their own work), and a discussion in a chat session. After the review round all subjects iterated (see Table 7.7), and this intervention was highly appreciated (see § 7.4.3.3). Subjects were not very positive about the chat session, and not all subjects iterated afterwards. Trigger 5 (new information) was introduced in the form of a contact moment with a domain expert and an e-mail with new information concerning the available resources. Most subjects also iterated after the contact with the domain expert. Fewer of them did so after the e-mail message (see Table 7.7). Some subjects were frustrated because not all information had been provided at the start of the study.

It is possible that the timing of interventions played a role: the e-mail message and the chat session took place later in the week and subjects may not have had enough time to adapt their design according to the new information they received. Another possibility is, however, that it was less clear to subjects what the consequences should be. The comments provided by other subjects were directly related to the subjects' work and to specific steps. For the other three interventions this was not the case: new information or new ideas were provided without direct instructions as to how and where they could influence the design of simulator training programmes. This problem could be addressed by:

- providing general guidelines stating where different kinds of information are used during the design process, and/or
- coupling scheduled meetings or discussions to the design cycles.

The risk of iterating immediately is that it might be difficult for designers to keep an overview of their design and the design process. It also seems more efficient to organise iteration in design cycles only: all new information, opinions, arguments, etc., can be collected at one moment and used to adapt the design in one go during the next design cycle. However, this might be very inefficient also: if important information about, for example, the resources available for training is not used immediately, designers might spend a lot of time on a design that it not feasible before they iterate at the end of a design cycle.

Even with a list of triggers for iteration it remains impossible to predict when and how iteration will be necessary or desirable. Thus, designers will have to organise and manage iteration themselves. The results of the evaluation studies suggest that there may be more than one good way to deal with this problem: circumstances differ (e.g., the time and manpower available) and designers have different capabilities and preferences. In the next section I will define measures that can be taken to support this aspect of the design task.

10.3 Support for iteration

Given the fact that iteration during the instructional design process is inevitable, the next question is: how can designers be helped to manage an iterative design process? In a way, designers face a double task. On the one hand, they need to work systematically in order to ensure that all important issues are addressed, and on the other hand, they should react adequately to triggers for iteration in order to ensure that their solution is optimal given the latest insights and information. They will have to choose the right level of detail, keep an overview of the evolving design product as well as the design process, decide when iteration is necessary and when it might actually be counterproductive, etc. Based on the literature and empirical research reported in this thesis, I propose a number of measures that can help designers to deal with iteration during the instructional design process (see Table 10.1). Some of them have already been mentioned as support for the design process in general (see § 3.4), while others are more specific.

Table 10.1: Measures to support iteration

Category	Measure
Structure the design process	1. Use a systematic instructional development method
	2. Define design cycles
Plan triggers for iteration	3. Plan decision points regarding the design process
	4. Plan review moments
	5. Generating and compare alternative solutions explicitly
Specific advice regarding iteration	6. Strategic advice
	7. Help to recognise triggers for iteration
Measures to make iteration easier	8. Keep intermediate results easily available
	9. Make consequences of changes visible
	10. Store notes and keep them available
	11. Keep information on constraints separate and available
	12. Make assumptions explicit
	13. Have experts available for advice

Measures to support the management of iteration include explicitly structuring the design process by using systematic methods and defining design cycles:

1) Use a systematic instructional development method
When the design of training simulator specifications is clearly structured, it will be easier for designers to keep an overview of what they have done and what they still need to do, and to make sure that no important aspects are forgotten. The process can be structured by prescribing which design activities need to be executed and which decisions have to be made. Many methods also prescribe the order in which these should be executed, in the form of a sequence of steps. This does not mean that the design process will always be the same: depending on, for example, the complexity of the domain and the amount of information available, design activities will take more or less time, or need more or less emphasis. The relationships between different design activities should be made explicit, so that users can understand the impact their decisions will have on the rest of the design.

2) Define design cycles
The definition of design cycles is a way to make iterations explicit. Because each project is different, it is impossible to prescribe a fixed sequence of steps or a fixed number of iterations. It is possible, though, to differentiate between the way to execute design activities early

in the design process and the way to work later on in more detailed design stages. Designing on a more global level first enables users to get a rough idea of the kind of training and the kind of instructional product they are designing and will help them to get an overview of the design and the design process before elaborating the design in more detail. In a way, design cycles force the designers to come back to their own work and review what they have done. Describing the expected intermediate products and the level of detail explicitly helps users to decide how far they should go into detail.

Other measures concern triggers for iteration that can be explicitly planned:

3) Plan decision points regarding the design process
Structuring the design process can be supported by planning explicit decision points to reflect on, and if necessary, adapt the design process. Decision points can be planned at fixed moments in the design method, prescribed by the organisation or defined for a specific project by the designers themselves. When decision points are linked to the design method, checklists or guidelines can be provided. This could help users, for example, to decide whether or not they have gone into enough detail to go on. In a more extreme case, users may have to decide that they cannot continue because vital information is not available. At decision points users can also be asked to check whether the assumptions that they have made are still valid, and whether their information about the available resources is still up-to-date (see also Measure 11 and 12).

4) Plan review moments
Another measure to support the design process is the definition of review moments where other people are invited to inspect the design and give comments or suggestions. The reviewers can be stakeholders or managers (e.g., in acquisition procedures), but also other designers (as in the peer review in the second evaluation study, see § 7.3.6.3) or experts in a certain area (as in the feedback rounds in the first evaluation study, see § 6.3.6.3). Another example of the latter is the suggestion to involve instructional design experts in the definition of training activities since this proved to be difficult (see § 8.3.2.3). Review moments are certainly necessary when not all of the relevant kinds of expertise are represented in the design team. Involving a wider range of stakeholders in reviews might complicate the design process, but will probably also enhance the acceptance of the chosen solution. Even a summary and discussion of the work done so far by the design team itself ('zoom out') can be seen as a review moment. Review moments are often included in formative evaluations and can be combined with decision points (Measure 3), or design cycles (Measure 2).

5) Generate and compare alternative solutions explicitly
Experts generate alternative solutions to design problems, and do not commit to one solution until later on (see § 3.2.2.3). It seems probable that the product of the design process will improve when designers generate and compare alternative ideas at important moments in the design process (see also § 10.2, under Trigger 4). An example of this measure is the separate step in the first phase of the SLIM method where users are asked to sketch a number of possible solutions in general terms, up front (see § 9.3.1). Comparing the advantages and disadvantages of alternative solutions can be supported with dedicated decision-making support systems (see § 3.3.7.2).

Specific advice can be given about how to recognise and deal with triggers for iteration that cannot be planned:

6) Strategic advice
Strategic advice concerns the way a certain design activity should be executed. It tries to give an answer to questions like: What should I do now? How should I do this work? Which mistakes should I avoid? Which information do I need? Which constraints should I take into account? How do I decide whether I have enough input or information to go on? When should I stop with this and go on to the next design activity? Who can I ask for advice or additional

information? Strategic advice can be provided in the form of, for example, process-oriented guidelines, examples, or checklists. It can be made concrete and directly applicable by linking it to steps or activities in the design method, to decision points or to review moments (Measures 1, 3 and 4).

7) Help to recognise triggers for iteration

This specific form of advice aims at helping users to recognise when triggers for iteration occur. It concerns the relationships between design activities and events that should alert users to go back and review decisions or elaborate their work further. This kind of advice can be related to a specific design activity or step, such as the different forms of process-oriented help in the MASTER-TPD tool (see § 5.4.2 and § 5.4.3). It can be given in the form of general advice, such as guidelines stating where and how different kinds of information should influence the design (as proposed in § 10.2), or it can be attached to regular checks at decision points or review moments (Measure 3 and 4). This advice can also explain when iteration is not desirable or necessary. For example, that less important adaptations can be postponed until the next design cycle. Again different formats are possible: guidelines, warnings, the description of demands on the input and the output, etc. In a software tool warnings can be provided if events that might make iteration necessary can be automatically recognised.

Table 10.2: Relation of Measures 1 to 7 with triggers for iteration

Measures ► / ▼ Triggers	M1: Systematic method	M2: Design cycles	M3: Decision points	M4: Review moments	M5: Explicit alternative	M6: Strategic advice	M7: Recognise triggers
T1: Discovery of missing input	*		*			*	*
T2: Need to repairing errors	*			*			*
T3: New insights based on work later on	*	*				*	*
T4: New ideas of the designer(s)		*			*		
T5: New information			*	*		*	*
T6: New opinions/arguments from other people				*		*	*
T7: Acquisition procedures		*		*			*

The seven measures proposed above try to support dealing with triggers for iteration in different ways. More research is needed to evaluate their usefulness and usability. As a starting point for this kind of research an overview of the most important relationships between triggers and measures is proposed in Table 10.2:

The relationship is, of course, most clear for the support that is intended to help designers to recognise triggers for iteration (M7). This measure is related to all triggers, except for the 'unexpected' new ideas of designers (T4), which are difficult to foresee and support (see § 10.2). Furthermore, identifying and dealing with missing input (T1) is most clearly supported by a using a systematic method (M1), decision points (M3) and strategic advice (M6). Identifying and repairing errors (T2) is most notably supported by clearly linked steps or design activities in a systematic instructional development method (M1) and review moments (M4). Dealing with new insights that result from work later on in the design process (T3) might be supported by a systematic method (M1) as well, by defining design cycles (M2) and strategic advice (M6). Generating new ideas (T4) is explicitly encouraged by M5 and, to some extent supported by planning design cycles (M2). Collecting and dealing with new information (T5) is supported most obviously by planned decision points (M3) and review moments (M4), together with strategic advice on this issue (which can be part of M6). Review moments (M4) and strategic advice (M6) can also encourage designers to explicitly ask for other opinions and new arguments (T6). And finally, acquisition procedures are supported explicitly by defining design cycles (M2) with planned review moments (M4).

The Measures 8 to 13, described below, are forms of knowledge management that make it easier for designers to iterate. They can be seen as prerequisites for iteration.

8) Keep intermediate results available
To make it possible to reuse information that was collected earlier and to elaborate on previous work, all intermediate results should be stored and be made available to users at all times. It should be easy to navigate to and review earlier work. All measures to help users to keep an overview can also stimulate iteration. When designers feel sure that they can explore without losing overview and without losing the work that they have done so far, they might be more inclined to elaborate more alternatives in parallel or to explore less standard solutions.

9) Make consequences of changes visible
Where possible users should be helped to see the consequences of their changes. For example, when the user changes the definition of a training goal, the part of training that addresses that training goal (i.e. the training activities and training scenarios) could be highlighted so that the users' attention is drawn to things that may have to be revised as well.

10) Store notes and keep them available
Notes are important to document the reasons that underlie decisions, the alternatives that were considered and why they were discarded. (see § 5.4.2). Notes can be used to defend decisions to stakeholders or managers, but they can also support iteration. Reviewing notes ensures that valid arguments that have already been discussed are not forgotten when a decision is reviewed. When the first choice turns out to be not feasible, one of the alternatives described in the notes can be taken as a new starting point. It is even possible that a previously discarded idea is better in the light of new insights or changed constraints. Notes are even more important when designers work in a team, or when other people will have to take over at some moment during the design process. Different types of help can be used to advice users on making notes (see, e.g., the guidelines and hints in § 5.4.3.1). Target users even suggested to force users to make notes at important decision points (see § 9.2.2).

11) Keep information on constraints separate and available
The resources that are available for designing specifications, for the acquisition of instructional products, and for the delivery of instruction and training later on, influence the design process, but they are not specifically linked to one step or design activity. Both target users and one subject in the second evaluation study proposed to start with an explicit inventory of resources and constraints. This measure was implemented in the SLIM method and proved to be successful (see § 9.3.1). Dealing explicitly with resources and constraints may also counter the inclination of novice designers to spend too little time on problem analysis (found in descriptive studies, see § 3.2.2.3). Guidelines can advice users to check whether their information about resources and constraints is up-to-date, e.g., after each step, at decision points, at review moments or at the beginning of a new design cycle (Measures 1 to 4). General guidelines can inform users what is likely to be affected when certain types of constraints change (Measure 7).

12) Make assumptions explicit
Often not all the required information is available, information is insecure, or designers cannot foresee whether their ideas will be feasible within the given constraints. Therefore, the designers will have to make assumptions and take preliminary decisions based on whatever information is available. This is not necessarily a problem: if an assumption turns out to be incorrect, earlier decisions can be revised. However, this is only possible when the designers have kept track of the assumptions they have made and regularly check whether these assumptions are still valid, e.g., at decision points, review moments or at the beginning of a new design cycle (Measures 2, 3 and 4).

13) Have experts available for information and advice
During both evaluation studies subjects used the possibility to ask experts for advice (see Tables 8.3 and 8.4). Also during projects with customers it proved to be necessary to collect additional information and ask colleagues for advice (sometimes even during workshops, see § 9.3.2). When experts are not part of the design team they should be available for questions, for example, by phone or by e-mail. It might be helpful to explicitly identify experts in different areas at the beginning of the design process and to ask them when they are available (see also Table 9.1 in § 9.5).

The different measures described above should help the users to take process-oriented decisions and encourage an iterative design process. The challenge is to offer concrete advice at the right moment and to clearly structure the design process, while at the same time allowing and supporting users to organise their own work flow in very different ways, depending on the characteristics of the project and the domain and their own preferences. Some of the measures are quite strong, e.g., prescribed design cycles or automatic warnings. They should not be seen as algorithms or cookbook recipes since it is clear that these are not available for ill-structured instructional design problems. The risk is that designers, especially novice designers, will interpret them as prescriptions instead of the heuristics that they are. Therefore, more research is needed to investigate how designers use this kind of help, and whether it helps them to deal with iteration adequately.

10.4 Generalising results

10.4.1 From evaluation studies to reality

In the evaluation studies described in Chapters 6 to 8 subjects worked on a realistic case and spent a realistic amount of effort on the design task. For the most part, the subjects in the evaluation studies had the same characteristics as target users and they worked in a very similar way. The subjects had more background knowledge about instructional design and development models and less domain knowledge than most target users of the MASTER method would have. However, the results of the evaluation study with target users reported in Section 9.2 do not contradict the results of the evaluation studies in Chapters 6 to 8. They indicate that an extensive theoretical background is not required, at least not when experienced instructional designers are available to answer questions about the method and the tool. In practice, it seems advisable to make sure that the team includes, or has access to, both domain experts and instructional design experts. Both the instructional design literature and the case studies presented in § 9.3.2 indicate that, in practice, it is advisable to combine a systematic design approach with a relational or communicative approach (see § 3.3.7.1).

The evaluation studies were condensed in time. In reality, the design of simulator specifications will take much longer. There will be more factors to take into account (e.g., environmental or political factors not represented in the case) and more 'disturbing' events to trigger iteration (e.g., new information about the operational system that will be bought, comments from managers and stakeholders, or budget changes). In the end, the draft training programmes will have to be elaborated in more detail than in these studies, in order to be able to formulate detailed simulator specifications. Working in a design team will also cause additional iteration, especially when the work of different team members has to be integrated or when personnel changes occur. This means that measures to support iteration will become even more important.

10.4.2 Novice and experienced designers

The design method used in the evaluation prescribes a systematic approach based on Instructional System Development (see § 3.3.3). The ISD approach has been challenged in the last

years. Literature research and the studies in this thesis indicate that it works for novices: it structures the design process, prescribes design activities and decisions, helps the designers to keep an overview, and frees up memory for other tasks. Is it, however, also suitable for experienced instructional designers? Experts may not need a systematic approach to help them organise and manage the design process. Even for them, though, it has advantages:
- it supports iteration when combined with suitable support (see § 10.3),
- it enhances the traceability of the design process and the reuse of information,
- it ensures that all important issues are addressed,
- it enables designers to justify their decisions, and
- it enables designers to delegate and carry over design tasks.

Moreover, in many cases organisations will enforce the use of a specific systematic method.

Whether experts will want to use a systematic, prescriptive method will probably depend on whether they are given the freedom to adapt the method to their preferences and to the specific design problem at hand. In practice, instructional designers leave out design activities that do not apply in a given project or execute them in a different order (see § 3.2.2.1) and they combine working top-down with working bottom-up, using design patterns and solutions from previous projects (see § 3.2.2.3). Descriptive research literature (see § 3.2.2.3) and experiences with the SLIM method (see § 9.3.2.3) show that experienced instructional designers can use a systematic method, and adapt it on the fly. Even when experienced designers choose their own (sequence of) design activities a systematic method could be used as a way to check which activities have already been done and which issues are already addressed and which are not.

10.4.3 Simulators and other instructional products

In Section 1.3.3, I discussed that simulators differ in some aspects from other instructional products. Technical and domain-related factors necessarily have much influence on the design process. A training simulator has to simulate the operational system and/or the environment. Therefore, designers have the double task of ensuring that the required (functional) fidelity is achieved and that the required facilities for training and instruction are included. The two are difficult to separate and designers are inclined to focus on fidelity aspects, probably because these are both complex and concrete. Often, thinking about facilities for training and instruction and the integration of simulators in the course is seen as an extra task rather than a fundamental starting point. This might be different for other instructional products. The designers of a lesson plan for teachers or for CBT to be used in a school, for example, are more likely to have a background in instructional design and development theory. There is also more support available in the form of templates or hybrid design and development models (see § 3.3.4) and authoring systems and other tools (see, e.g., § 3.3.8).

Simulators are likely to be technically more complex and more expensive than many other instructional products. This might make the design process longer and more iterative. However, the design of all instructional products is an inherently iterative process and there are no reasons to believe that the triggers for iteration defined in Section 10.2 will not occur in other instructional design processes. It is possible, though, that there could be other ways to deal with iteration: when the end product can be more easily tried out and adapted, a prototyping approach can be used as an additional measure to support iteration. Trying out prototypes, for example, can be a powerful way to trace design errors or sub-optimal decisions, and to compare alternative solutions (see § 3.3.5, also for risks and disadvantages). It is a more concrete review of the design than is possible with only specifications and it makes it easier to involve target trainees and evaluate the effects of the instructional product on the learning process.

10.5 Further research

Of course, my research has led to new questions, both regarding the usability of the MASTER method and regarding the role of iteration in instructional design. I will briefly discuss some interesting directions below.

Further research regarding the MASTER method
For the case that was used, the MASTER method proved to be suitable. Although there are no reasons to expect very different results with other cases, further application of the MASTER method is required in order to confirm its usability for other domains and settings.

A more fundamental question is how the instructional development methods can best be applied in practice. In general, there are two approaches. One is to try to find ways to turn novice instructional designers into experts as quickly as possible. The other one is to accept that non-experts will design specifications for instructional products and to try to develop tools (in a broad sense) that incorporate instructional design and development models and, thus, can make up for the novices' lack of background in this field. The MASTER method and the research in this thesis have taken the latter approach.

Two options to support novice designers that have been tried out in this thesis are:
1. A systematic method with an accompanying software tool, combined with face-to-face meetings with experts and the option to ask questions by e-mail.
2. A systematic method applied collaboratively during workshops under guidance from an instructional design expert.

A third option would be to develop domain-specific templates or partly elaborated solutions for certain types of design problems that novice designers can use by filling in information (see also § 3.3.4.2).

More research is necessary to investigate to what extent these options are feasible and to investigate which of them is the most effective and efficient choice in which circumstances. We may have to accept that novice designers come up with adequate, but not always optimal or innovative, designs and that they need more time and effort than experts would need.

Factors that determine the desired amount of iteration
As discussed in Section 7.5, future research should investigate in more detail when iteration is desirable and when not. For this purpose, the instructional design process needs to be closely followed. One option is to participate and observe design teams in real projects (ethnographic studies), or to introduce support for iteration and investigate the effects in the tradition of formative development research (van den Akker, 1999; Reigeluth and Frick, 1999). First attempts in that direction are the case studies presented in Section 9.3.2. Another option is to combine empirical studies similar to the ones reported in this thesis with shorter periods of registering 'think-aloud' protocols, preferably at significant moments in the design process. The list of triggers for iteration, proposed in Section 10.2, can serve as a basis for more research in this area.

The desirable amount of iteration is influenced by characteristics of the design problem, such as the amount of time and resources available for the design of training simulator specifications. In general, the more people involved and the longer a project lasts, the more the design process needs to be structured and documented. Also the main goals and demands of the customers play a role. Van Wagenberg (1992) suggests, for example, that a lot of iteration is good for the goal of improving the quality of the design and for enhancing innovation, but not for the goal of minimising the design time and costs (see also § 5.4.1). Finally, there may also be a link between iteration and the capabilities of the designers. Even when the amount of iteration is similar, good designers might iterate for different reasons than weak designers (see the discussion in § 7.5.2). Good designers might be more capable of planning successful iterations, i.e. iterations that improve the result of the design process. They might also be better at recognising when the optimum has been reached and further iteration will no longer lead to changes that improve the learning process of future trainees.

Research regarding support for iteration
The research presented in this thesis has provided more insight into the kinds of events that can trigger iteration and different forms of support that can help designers to deal with iteration. However, more research is required to investigate how designers use these forms of support, and whether it helps them to deal with iteration adequately. The measures proposed in Section 10.3 could be a starting point for this research.

With more insight into the factors that determine the desired amount and kind of iteration (see above), support for iteration could be further adapted to characteristics of the design problem and/or the designer. Another approach could be to investigate 'best practices' and to try to get to more differentiated instructional development models, adapted to the characteristics of different kinds of design problems.

Communication and the design process
Within the design team, the designers have to co-operate. To be able to share the workload they need to share information and results, and integrate the work done by different team members. Designers also have to gather the information that they need for (their part of) the design task. It is difficult, especially for novice designers, to know what they are looking for and to ask the right questions, especially when time and resources are limited. Furthermore, people will not always be willing or able to easily share their knowledge. Information can be contradictory and, sometimes, knowledge does not reside in one head and can only be obtained by bringing people together in order to construct the full picture or to generate new ideas.

Thus, communication plays a large role in design processes. To optimise this aspect of the design process more research into communication and organisation theories is required. It is not immediately clear which communication model fits the instructional design process. A receptive model is clearly not suitable: designers cannot wait until others take the initiative to communicate or send important information to them. An interaction model might fit, provided that it suits both co-operation between members of the design team and negotiations with other stakeholders. Acquisition procedures, on the other hand, suggest a submit and review model. A knowledge engineering model might be more suitable for the situations where designers have to 'extract' knowledge from experts. As starting point for this kind of research, Table 9.1 gives an overview of the roles that people involved in a design process can take. Zagers (2001) proposes different ways to support communication and co-ordination in a web-based project environment (see § 9.4).

Design for innovation
Applying a systematic instructional development method, such as the MASTER method, ensures that important issues are addressed, but it does not always lead to innovative solutions. This was clearly visible in the case studies represented in Section 9.3.2. In the two evaluation studies reported in Chapters 6 to 8 the subjects also came up with rather standard solutions. It is not clear whether this was caused by the relatively limited time scale, the subjects' inexperience, the MASTER method or the case itself. By prescribing a standard sequence of design activities, an instructional development method supports the planning and managing of the design process. It also offers a framework for the selection and application of instructional design models and principles. When a part of the load is taken of the designers' shoulders, they should, in principle, have more time and memory capacity left for exploring less standard solutions. However, some subjects expressed the feeling that the structured approach of the method limited their creativity (see § 8.3.1.2). There are indications that a good deal of iteration is necessary for innovation (Blessing, 1992; Braha and Maimon, 1994, see also § 3.2.2.2; and van Wagenberg, 1992, see also § 5.4.1). Some authors claim that a

prototyping approach would be more suitable if the product should be innovative (e.g., Visscher-Voerman, 1999).

The question is how iteration and creativity can be used in an effective and efficient way. In the design of simulator training the definition of training activities is the key to innovation: as long as designers define traditional training activities, the result will be a traditional training programme of classroom-based theory lessons followed by supervised practise (see § 8.3.2.3). Examples of measures that can be taken to encourage creativity specifically in this design activity are: explicitly asking users to generate a number of alternative (combinations of) training activities, advising users to organise a brainstorm sessions oriented towards generating 'unexpected' solutions, and providing facilities to compare advantages and disadvantages of different types of training activities. Iteration and reconsideration of training activities can be stimulated by, for example, focusing reviews specifically on training activities (e.g., at decision points or in acquisition procedures) or encouraging the designers to go through a number of design cycles quickly in order to get an idea of the consequences of choosing certain (combinations of) training activities on the design of the learning environment. Another idea is to encourage creativity at the beginning of the design process, for example by asking designers to sketch a number of possible solutions in global terms very early on (as for example in the SLIM method described in § 9.3.1). This makes the designers conscious of the fact that other solutions exist and provides a point to iterate back to if there are reasons for investigating alternatives. In the field of mechanical engineering Blessing (1994) advocates the use of problem-oriented models, which prescribe that the design process should start with an abstraction and reconsideration of the problem, rather than with the immediate analysis of the initial product idea (see also § 3.2.2.2). Descriptive research in the area of instructional design confirms that experienced designers spend more time and effort on the analysis of the design problem and generate a number of alternative solutions.

The idea behind such measures to encourage creativity at specific moments in the design process is to support users with systematic design methods for those parts and aspects of the design task that can be standardised, so that they can spend more time and effort on those design activities that require human skills and creativity. More research is required to investigate whether such measures can indeed stimulate the right kind of iteration and creativity and whether they result in designs which are better than the more traditional ones.

10.6 Final remarks

The general problem addressed this in this thesis is (see § 1.1):

How can instructional designers be supported during the iterative design process in order to design effective instructional products?

This question is clearly important from a theoretical viewpoint. However, the need for support for an iterative design process is clearly felt in practice as well. The process of designing specifications for instructional products can be chaotic, especially when different groups of stakeholders are involved. Although instructional design and development models are very useful, they do not help instructional designers to customise and continuously adapt the design process itself. They also do not help them to deal with the large variety of pragmatic factors (ranging from financial constraints to conflicts in the design team). Nearly all the instructional development models described in the literature recognise the iterative nature of the design process. They do not, however, explain when and how iteration should take place.

In the instructional research literature I found hardly any studies regarding this aspect of the design process. At the time I was surprised, since iteration is obviously an important and difficult aspect of the design task. Now, at the end of my own Ph.D. research, I understand why I did not find research studies regarding iteration in instructional design: iteration is difficult to observe and measure. Studying iteration during the design process requires a longer time-span, a case that is realistic in size and complexity, and the occurrence of the kind of events and interferences that would trigger iteration in practice. Thus, controlled laboratory studies are not suitable, which is unfortunate because think-aloud protocols would give the most information about why and how designers iterate. Case studies, on the other hand, would be very long and labour-intensive, and they give the researcher no control over what happens. Furthermore, the differences between cases are likely to be so large that it will be difficult to compare them in order to come to a more general theory regarding iteration. In my own evaluation studies the MASTER method and prototype tool and the realistic case offered an environment in which it was possible to define and study iteration. The empirical setting allowed some control over what happened and made it possible to record a considerable amount of information about the subjects' design process.

The research presented here does not give a definite answer to the question posed above, but it contributes to our understanding of the role of iteration in instructional design. Both descriptive studies in the literature and my own studies showed that iteration during the instructional design process is unavoidable and not necessarily bad. Iteration can improve the result and there are indications that truly innovative design requires a good deal of iteration. So, what designers need is support to deal with iteration. Developing such support requires more insight into the different reasons for iteration, and more insight into when iteration is desirable and when it is not. The list of triggers for iteration, proposed in Section 10.2, and the measures to support iteration, proposed in Section 10.3, provide a common language to discuss this aspect of instructional design and they can serve as a basis for more research. The description of triggers for iteration is also interesting for practitioners because it can give them more insight into, and grip on, this difficult aspect of their design task.

References

Aducchio, M. T. (1997). Lessons learned in the development of high fidelity maintenance trainers. In *Proceedings of the 19th Interservice/Industry Training, Simulation and Education Conference* (pp. 369-376). Arlington, VA: National Training Systems Association (NTSA).

Advanced Distributed Learning Initiative. (2001). *Sharable Content Object Reference Model (SCORM): Version 1.2. The SCORM Overview* (d.d. October 1, 2001). Retrieved September 20, 2003, from http://www.adlnet.org.

Ainsworth, S. E., Underwood, J. D., & Grimshaw, S. K. (1999). Formatively evaluating REDEEM: An authoring environment for intelligent tutoring systems. In S. Lajoie & M. Vivet (Eds.), *Artificial intelligence in education open learning environments: New computational technologies to support learning, exploration and collaboration* (pp. 93-100). Amsterdam: IOS Press.

Akker, J. van den. (1999). Principles and methods of development research. In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 1-14). Dordrecht, the Netherlands: Kluwer Academic.

Akker, J. van den, Branch, R., Gustafson, K. L., Nieveen N., & Plomp, T. (Eds.). (1999). *Design approaches and tools in education and training.* Dordrecht, the Netherlands: Kluwer Academic.

Alessi, S. M. (1988). Fidelity in the design of instructional simulations. *Journal of Computer-Based Instruction, 15*(2), 40-47.

Alessi, S. M., & Trollip, S. R. (2001). *Multimedia for learning: Methods and development.* Needham Heights, MA: Allyn & Bacon.

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review, 94*(2), 192-210.

Anderson, J. R., Boyle, C. F., Corbett, A. T., & Lewis, M. W. (1990). Cognitive modelling and intelligent tutoring. *Artificial Intelligence, 42,* 7-49.

Andrews, D. H., & Goodson, L. A. (1980). A comparative analysis of models of instructional design. *Journal of Instructional Development*, *3,* 2-16.

Barnard, Y. F., & Boy, G. (2003, September). *Knowledge management in the design of safety-critical systems.* Paper presented at EuroCogSci03 (Cognitive Science Europe), Osnabruck, Germany.

Barnard, Y. F., Boy, G., Tremaud, M., Payeur, F., & Fauré, X. (2002). Articulation of Operational and Training Materials. In S. Chatty, J. Hansman, & G. Boy (Eds.), *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics* (pp. 30-35). Menlo Park, CA: AAAI Press.

Barnard, Y. F., & Sandberg, J. A. C. (1994). *The learner in the centre: Towards a methodology for open learner environments.* Doctoral dissertation, University of Amsterdam.

Beck, K. (1999). *Extreme programming explained: Embrace change*. Reading, MA: Addison-Wesley.

Berg, E. van den. (1999). Walker's deliberative approach in a small-scale project: The SPIN-case. In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 71-80). Dordrecht, the Netherlands: Kluwer Academic.

Berlo, M. van. (2002). *Experimental validation of instructional design guidelines supporting the analysis of team tasks* (Report No. TM-02-B004). Soesterberg, the Netherlands: TNO Human Factors.

Blessing, L. T. M. (1994). *A process-based approach to computer-supported engineering design.* Doctoral dissertation, University of Twente, Enschede, the Netherlands.

Block, J. H. (1971). *Mastery learning*. New York: Holt, Rineheart & Winston.

Boer, J. P. A. (1991). *Het gebruik van simulatoren voor opleiding en training I: Bepalende factoren voor de waarde van een simulator als leermiddel* [The use of simulators for education and training I: Factors that determine the value of a simulator as a learning tool] (Report No. IZF-1991-A-48). Soesterberg, the Netherlands: TNO Institute for Perception (in Dutch).

Bone, R., & Lintern, G. (1999). Rehearsal versus map study as preparation for a flight navigation exercise. *Human Factors, 41*(3), 467-473.

Boot, E. W., & Bots, M. (2002). Learning object creation, management and re-use by non-experienced content developers. In *Proceedings of the 22th Interservice/Industry Training, Simulation and Education Conference* (pp. 446-453). Arlington, VA: National Training Systems Association.

Boot, E. W., & Rooij, J. C. G. M. van. (2000). *Gestructureerde ontwikkeling van computerondersteund onderwijs met behulp van templates: Specificatie* [Structured development of CBT-using templates: Specification]. (Report No. TM-00-A004). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Boot, E. W., Verstegen, D. M. L, & Veerman, A. L. (2002). *Behoeftestelling voor GOLMen ten behoeve van SQUIRE* [Needs assessment for advanced training means for the SQUIRE radar system] (Report No. TM-02-A029). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Bosch, K. van den, Barnard, Y. F., & Helsdingen, A. S. (1999). *Taak- en trainingsanalyse beeldanalist SPERWER* [Task and training analysis image interpreter SPERWER], (Report No. TM-99-A018). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Boy, G. A. (1999). Organizational memory systems. In *The encyclopaedia of Computer Science and Technology: Vol. 40* (pp. 1-45). New York: Encyclopaedia of Computer Science and Technology.

Braha, D., & Maimon, O. (1997). The design process: Properties, paradigms, and structure. *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans, 27*(2), 146-166.

Branson, R., Rayner, G., Lamar Cox, J., Furman, J., King, F., & Hannum, W. (1975). *Interservice procedures for instructional systems development: Part 5. Executive summary and model.* (Report No. AD-AO19 486). Florida State University.

Breda, L. van, & Boer, J. P. A. (1988). *Validatiestudie rijsimulator Leopard II.* [Validation study driving simulator Leopard II] (Report No. IZF 1988-M7). Soesterberg, the Netherlands: TNO Institute for Perception (in Dutch).

Cannon-Bowers, J. A., Tannenbaum, S. I., Salas, E., & Converse, S. A. (1991). Toward an integration of training theory and technique. *Human Factors, 33*(3), 281-292.

Carver, E. M., McGuinness, B., & Bosch, K. van den. (1996). *Skill retention: Field orientation*. (EUCLID RTP 11.1 MASTER Deliverable B3.2). Soesterberg, the Netherlands: TNO Human Factors.

Cline, R. W., & Merrill, M. D. (1995). Automated instructional design via instructional transactions. In R. D. Tennyson & A. E. Barron (Eds.), *Automating instructional design: Computer-based development and delivery tools* (pp. 317-353). Berlin, Germany: Springer-Verlag.

Commando Opleidingen KL. (1996). *Beleid Opleiden COKL* (d.d. 7-6-1996, Aanvulling Hoofdstuk 3, d.d. 12-12-1997) [Policy for Education and Training by the Staff Education and Training from the Royal Netherlands Army, d.d. 7-6-1996; Update Chapter 3, d.d. 12-12-1997]. Utrecht, the Netherlands: Royal Netherlands Army, COKL, Dept. PL/BO (in Dutch).

Croock, M. B. M. de, Paas, F., Schlanbusch, H., & Merriënboer, J. J. G. (2002). ADAPT^{IT}: Tools for training design and evaluation. *Educational Technology Research and Development, 50*(4), 47-58.

DE 2.0. (1997). *Designer's Edge, Version 2.0* [Computer software]. Salt Lake City, Utah: Allen Communication.

Dehncke, R. W., & Brooks, D. J. (1998). Program management of a complex simulation program: Lessons learned from STOW. In *Proceedings of the 20th Interservice/Industry Training, Simulation and Education Conference* (pp. 33-42). Arlington, VA: National Training Systems Association (NTSA).

Dick, W. (1995). Instructional design and creativity: A response to the critics. *Educational Technology, 35*(4), 5-11.

Dick, W. (1997). A model for the systematic design of instruction. In S. Dijkstra, N. M. Seel, F. Schott, & R. D. Tennyson (Eds.). *Instructional design: international perspectives: Vol. 2* (pp. 83-113). Hillsdale, NJ: Erlbaum.

Dick, W., & Carey, L. (1996). *The systematic design of instruction* (4th ed.). New York: Harper Collins College.

Dijkstra, S. (1997). The integration of instructional systems design models and constructivistic design principles. *Instructional Science*, *25,* 1-13.

Dijkstra, S. (Ed.). (2001a). Epistemology, psychology of learning and instructional design. [Special issue]. *Instructional Science, 29*(4-5).

Dijkstra, S. (2001b). The design space for solving instructional-design problems. *Instructional Science*, *29,* 275-290.

Dijkstra, S., Seel, N. M., Schott, F., & Tennyson, R. D. (Eds.). (1997). *Instructional design: International perspectives: Vol. 2. Solving of instructional design problems.* Hillsdale, NJ: Erlbaum.

Directie Materieel. (1993). *Handleiding Materieelprojecten* (KL, d.d. 1-3-1993) [Manual Acquisition Projects, d.d. 1-3-1993]. The Hague, the Netherlands: the Royal Netherlands Army, Directie Materieel (in Dutch).

Standard Learning Technology Systems Architecture (2001) *LTSA Draft 9* (d.d. 30-11-2001). Retrieved September 20, 2003, from http://edutool.com/ltsa.

Farmer, E. W., Jorna, P. G. A. M., Riemersma, J. B. J., Rooij, J. C. G. M. van, & Moraal, J. (1999). *Handbook of simulator-based training.* London: Ashgate.

Flexman, R. E., & Stark, E. A. (1987). Training simulators. In G. Salvendy (Ed.), *Handbook of Human Factors* (pp. 1012-1038). New York: Wiley & Sons.

Fowler, M. (2003). *The new methodology* (d.d. April 2003). Retrieved September 20, 2003, from http://www.martinfowler.com/articles/newMethodology.html.

Gagné, R. M. (1962). Simulators. In R. Glaser (Ed.), *Training Research and Education* (pp. 222-246). New York: Wiley.

Gagné, R. M. (1985). *The conditions of learning and theory of instruction* (4th ed.). New York: Holt, Rineheart & Winston.

Gagné, R. M., Briggs, L. J., & Wager, W. W. (1992). *Principles of Instructional Design* (4th ed.). New York: Holt, Rineheart & Winston.

Gagné, R. M., Tennyson, R. D., & Gettman, D. J. (1991). *Designing an advanced instructional design advisor: Conceptual frameworks* (Report No. AL-TP-1991-0017 VOL-5). San Antonio, TX: Brooks Air Force Base.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Reading, MA: Addison-Wesley.

Gettman, D., McNelly, T., & Muraida, D. (1999). The guided approach to instructional design advising (GAIDA): A case-based approach to developing instructional design expertise.

In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 175-182). Dordrecht, the Netherlands: Kluwer Academic.

Gilbert, L. (1999). Some valuable lessons from the teaching and learning technology programme in the UK. *Journal of Interactive Learning Research, 10*(1), 67-85.

Goel, V., & Pirolli, P. (1989). Motivating the notion of generic design within information-processing theory: The design problem space. *AI Magazine, 10*(1), 19-36.

Goel, V., & Pirolli, P. (1992). The structure of design problem spaces. *Cognitive Science, 16*(3): 395-429.

Goodyear, P. (1994). Foundations for courseware engineering. In R. D. Tennyson (Ed.), *Automating instructional design, development and delivery* (pp. 7-28). Berlin, Germany: Springer-Verlag.

Goodyear, P. (1997). Instructional design environments: Methods and tools for the design of complex instructional systems. In S. Dijkstra, N. M. Seel, F. Schott, & R. D. Tennyson, (Eds.), *Instructional design: International perspectives: Vol. 2* (pp. 83-113). Hillsdale, NJ: Erlbaum.

Greeno, J. G., Korpi, M. K., Jackson III, D. N., & Michalchik, V. S. (1990). *Processes and knowledge in designing instruction* (Report No. N00014-88-K-0152). Stanford, CA: Stanford University.

Gustafson, K. L. (1991). *Survey of instructional development models* (2nd ed.). Syracuse, NY: Syracuse University (ERIC Documentation Reproduction Service No. ED 335 027).

Halff, H. M. (1993). Prospects for automating instructional design. In J. M. Spector, M. C. Polson, & D. J. Muraida (Eds.) *Automating Instructional Design* (pp. 67-132). Englewood Cliffs, NJ: Educational Technology Publications.

Hamel, R. (1990). *Over het denken van de architect: Een cognitief psychologische beschrijving van het ontwerpproces bij architecten* [On the thought processes of architects]. Amsterdam: AHA books (in Dutch).

Hamel, R. (1999). Probleemoplossen en begrijpen [Problem solving and understanding]. In R. Hamel, M. Elshout-Mohr, & M. Milikowski (Eds.), *Meesterschap: Zestien stukken over intelligentie, leren, denken en probleemoplossen* (pp. 61-76). Amsterdam: Vossiuspers AUP (in Dutch).

Hannum, W. (2001). *Welcome to.... Instructional systems development* (d.d. 2001). Retrieved September 20, 2003, from http://www.soe.unc.edu/ISD/index.html.

Harvey, E. P. (1998). Simulation system verification and validation: The program management perspective. In *Proceedings of the 20th Interservice/Industry Training, Simulation and Education Conference* (pp. 66-75). Arlington, VA: National Training Systems Association (NTSA).

Holcomb, C., Wedman, J. F., & Tessmer, M. (1996). ID activities and project success: Perceptions of practitioners. *Performance Improvement Quarterly, 9*(1), 49-61.

Holleman, J. W., & Pilot, A. (2003). Onderwijsleerfuncties: Verslag van een denkproces [Instructional functions: Report of a reflection]. *Tijdschrift voor Hoger Onderwijs, 21*(2), 70-84.

Hoog, R. de, Jong, T. de, & Vries, F. de. (1994). Constraint-driven software design: An escape from the waterfall model. *Performance Improvement Quarterly*, *7*(3), 48-63.

Hoog, R. de, Kabel, S., Barnard, Y., Boy, G., DeLuca, P., Desmoulins, C., Riemersma, J., & Verstegen, D. (2002). Re-using technical manuals for instruction: Creating instructional material with the tools of the IMAT project. In Y. Barnard, M. Grandbastien, R. de Hoog, & C. Desmoulins (Eds.), *Integrating Technical and Training Documentation, Proceedings of the ITS2002 workshop* (pp. 27-38). San Sebastian, Spain/Biarritz, France: 6th International conference on Intelligent Tutoring Systems.

Hoogveld, A. W. M. (2003). *The teacher as designer of competency-based education.* Doctoral dissertation, Open Universiteit Nederland, Heerlen, the Netherlands.

Hoogveld, A. W. M., Paas, F., Jochems, W. M. G., & Merriënboer, J. J. G. van. (2001). The effects of a web-based training in an instructional systems design approach on teachers' instructional design behavior. *Computers in Human Behavior, 17,* 363-371.

Hoogveld, A. W. M., Paas, F., Jochems, W. M. G., & Merriënboer, J. J. G. van. (2002). Exploring teachers' instructional design practices from a systems perspective. *Instructional Science, 30,* 291-305.

Hsieh, P. Y., Halff, H. M., & Redfield, C. L. (1999). Four easy pieces: Development systems for knowledge-based generative instruction. *Journal of Artificial Intelligence in Education, 10,* 1-45.

Hulst, A. H. van der, Hoog, R. de, & Wielemaker, J. (1999). BOOT: Decision support for the selection of facilities for education and training (Report No. FEL-99-A188). The Hague, the Netherlands: TNO Physics and Electronic Laboratory.

Hulst, A. H., van der, & Verstegen, D. M. L. (2000). *GOLM ontwikkeling KL: Vormen van ondersteuning voor behoeftestelling* [Specification of advanced training means within the Dutch Army: forms of support for the user requirements specification] (Report No. FEL-00-A191). The Hague, the Netherlands: TNO Physics and Electronic Laboratory (in Dutch).

ID2 (2003). *Web site with information on ID2*. Retrieved September 20, 2003, from http://www.id2.usu.edu.

Janssen, N. H. E., Boot, E. W., & Verstegen, D. M. L. (2003). *The first outlines of blended learning solution for mechanical technical training of F-16 specialists: Outcome of the SLIM instructional design process* (Report No. TM-03-A034). Soesterberg, the Netherlands: TNO Human Factors.

Jeantheau, G. G. (1969). The use of multi-man system trainers. *Ergonomics, 12*(4), 533-542.

Jong, A. J. M. de. (1986). *Kennis en het oplossen van vakinhoudelijke problemen: Een voorbeeld uit een natuurkundig domein* [Knowledge based problem solving: an example from a physics domain]. Doctoral dissertation, Technical University of Eindhoven, the Netherlands (in Dutch).

Jong, T. de, Limbach, R., Gellevij, M., Kuyper, M., Pieters, J., & Joolingen, W. van. (1999). Cognitive tools to support the instructional design of simulation-based discovery learning environments: The SIMQUEST authoring system. In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 215-224). Dordrecht, the Netherlands: Kluwer Academic.

Kearsley, G. (2003). *Explorations in learning and instruction: The Theory Into Practice (TIP) database*. Retrieved September 20, 2003, from http://tip.psychology.org.

Kennedy, H. (2000). NATO steps up modelling and simulation activities. In *Training and simulation: trends and technology review* (pp.46-49). Arlington, VA: National Defense Magazine.

Kennedy, H. (2001a, July). Environmental regulations limit training of U.S. troops. *National Defense*, 2001, July, 32-34.

Kennedy, H. (2001b, November). Simulations help train for extreme risks. *National Defense*, 2001, November, 34-46.

Kerr, S. T. (1983). Inside the black box: Making design decisions for instruction. *British Journal of Educational Technology, 14*(1), 45-58.

Kessels, J. W. M. (1993). *Towards design standards for curriculum consistency in corporate education*. Doctoral dissertation, University of Twente, Enschede, the Netherlands.

Kessels, J. W. M. (1999). A relational approach to curriculum design. In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 59-70). Dordrecht, the Netherlands: Kluwer Academic.

Kessels, J. W. M., & Smit, C. A.(1989). *Opleidingskunde. Een bedrijfsgerichte benadering van leerprocessen* [Educational science: A commercial company oriented approach of learning processes]. Deventer, the Netherlands: Kluwer Bedrijfsinformatie (in Dutch).

Kincaid, B. (1997). *A dinosaur in Whitehall: The true cost of defence procurement bureaucracy*. London: Brassey's.

Kleiss, J. A. (1995). Visual scene properties relevant for simulating low-altitude flight: A multidimensional scaling approach. *Human Factors, 37*(4), 711-734.

Korteling, J. E., Bosch, K. van den, & Emmerik, M. L. van. (1997). *Low-cost simulators 1a: Literature review, analysis of military training and selection of task domains* (Report No. TM-97-A035). Soesterberg, the Netherlands: TNO Human Factors.

Korteling, J. E., Helsdingen, A.S, & Baeyer, A. von (Eds.). (2000). *Handbook of low-cost simulation for military training* (EUCLID 11.8 ELSTAR Deliverable ELS-DEL/5-HB). Grenoble, France: Corys TESS.

Korteling, J. E., Padmos, P., Helsdingen, A. S., & Sluimer, R. R. (2001). *Certificering van trainingssimulatoren 1: kennisinventarisatie* [Certification of training simulators 1: knowledge inventory] (Report No. TM-01-D003). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Kuiper, H. (1995). *An instructional support system for training simulators.* Doctoral dissertation, University of Twente, Enschede, the Netherlands (in Dutch).

Kuyper, M. P. (1998). *Knowledge engineering for usability: Model-mediated interaction design of authoring instructional simulations*. Doctoral dissertation, University of Amsterdam.

Kuyper, M., Hoog, R., de, & Jong, T. de. (2001). Modelling and supporting the authoring process of multimedia simulation based educational software: A knowledge engineering approach. *Instructional Science*, *29*(4/5), 337-359.

Le Maistre, C. (1998). What is an expert instructional designer? Evidence of expert performance during formative evaluation. *Educational Technology Research and Development, 46*(3), 21-36.

Le Maistre, K., & Weston, C. (1996). The priorities established among data sources when instructional designers revise written materials. *Educational Technology Research and Development, 44*(1), 61-70.

Lintern, G., & Garrison, W. V. (1992). Transfer effects of scene content and crosswind in landing instruction. *The International Journal of Aviation Psychology, 2*(3), 225-244.

Loube, J. (2001). In defence of ISD. *Modern Simulation & Training, 2001*(2), 44-46.

Major, N., Ainsworth, S., & Wood, D. (1997). REDEEM: Exploiting symbiosis between psychology and authoring environments. *International Journal of Artificial Intelligence, 8,* 31-340.

Marcke, K. van. (1998). GTE: An epistemological approach to instructional modelling. *Instructional Science*, *26,* 147-191.

McClure, J. T. (1998). Working smarter: Training and simulation developed through collaboration. In *Proceedings of the 20th Interservice/Industry Training, Simulation and Education Conference* (pp. 83-89). Arlington, VA: National Training Systems Association (NTSA).

McGuinness, B., Bosch, K. van den, & Verstegen, D. M. L. (1996). *Literature review on skill retention*. (EUCLID RTP 11.1 MASTER Deliverable B3.1). Soesterberg, the Netherlands: TNO Human Factors.

McKenney, S., Nieveen, N., & Akker, J. van den. (2002). Computer support for curriculum developers: CASCADE. *Educational Technology Research and Development, 50*(4), 25-35.

Melis, P., & Berlo, M. P. W. van. (2003). *Richtlijnen voor het opstellen van functionele specificaties voor computer-based maintenance training* [Guidelines for the formulation of functional specifications for computer-based maintenance training]. (Report No. TM-03). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Merriënboer, J. J. G. van. (1997). *Training complex cognitive skills.* Englewood Cliffs, NJ: Educational Technology Publications.

Merriënboer, J. J. G. van, Clark, R. E., & Croock, M. B. M. de. (2002). Blueprints for complex learning: The 4C/ID-model. *Educational Technology, Research and Development, 50*(2), 39-64.

Merriënboer, J. J. G. van, & Croock, M. B. M. de. (2002). Performance-based ISD: 10 steps to complex learning. *Performance Improvement, 41*(7), 33-38.

Merrill, M. D. (1983). Component display theory. In C. M. Reigeluth (Ed.), *Instructional design theories and models: An overview of their current status* (pp. 279-333). Hillsdale, NJ: Erlbaum.

Merrill, M. D. (1987). A lesson based on the component display theory. In C.M. Reigeluth (Ed.), *Instructional theories in action: Lessons illustrating selected theories and models* (pp. 201-245). Hillsdale, NJ: Erlbaum.

Merrill, M. D. (2001). Components of instruction towards a theoretical tool for instructional design. *Instructional Science*, *29,* 291-310.

Merrill, M. D., & the ID2 Research Group. (1996). Instructional transaction theory: Instructional design based on knowledge objects. *Educational Technology, 36*(3), 30-37.

Merrill, M. D., Li, Z., & Jones, M. K. (1992). Instructional transaction shells: Responsibilities, methods and parameters. *Educational Technology, 32*(2), 5-27.

Merrill, M. D., & Thompson, B. M. (1999). The IDXelerator™: Learning-centered instructional design. In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 265-279). Dordrecht, the Netherlands: Kluwer Academic.

Mettes, C. T. C. W., & Pilot, A. (1980). *Over het leren oplossen van natuurwetenschappelijke problemen: Een methode voor ontwikkeling en evaluatie van onderwijs*, *toegepast op een kursus Thermodynamika* [On teaching and learning problem solving in science: A method for the development and evaluation of instruction, applied to a course in thermo-dynamics]. Doctoral dissertation, University of Twente, Enschede, the Netherlands (in Dutch).

Mettes, C. T. C. W., Pilot, A., & Roossink, H. J. (1981). Linking factual and procedural knowledge in solving science problems: A case study in a thermodynamics course. *Instructional Science, 10,* 333-361.

Microsoft NetMeeting, version 2.1. [Computer software] (1997). Microsoft Corporation.

Mindware Creative. (2003). *IDXelerator* [Computer software]. Information about IDXElerator. Retrieved July 18, 2003, from www.mindwaresystems.com.

Ministry of Defence. (1996). *Defensiematerieelkeuzeproces (DMP)* (d.d. 23-9-1996) [Military Acquisition Procedure, d.d. 23-9-1996]. The Hague, the Netherlands: Ministry of Defence (in Dutch).

Ministry of Defence. (1997). *CDS Aanwijzing nr. 101 inzake behoeftestellingen* (d.d. 22-5-1997) [CDS guideline no. 101 regarding needs statements, d.d. 22-5-1997]. The Hague, the Netherlands: Ministry of Defence (in Dutch).

Moonen, J. (1999). The design and prototyping of digital learning material: Some new perspectives. In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 95-112). Dordrecht, the Netherlands: Kluwer Academic.

Moraal, J. (1985). *The validity of a tank driver training simulator.* Paper presented at the symposium on the military value and cost-effectiveness of training, Brussels, Belgium. (Report No. Proceedings 1985 1153). Soesterberg, the Netherlands: TNO Institute for Perception.

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education, 10,* 98-129.

Nash, T. (2002). Putting the horse before the cart: The TNA process explored. *Military Training & Simulation News, 4*(1), 12-14.

Nelson, W. A., Magliaro, S., & Sherman, T. M. (1988). The intellectual content of instructional design. *Journal of Instructional Development*, *11*(1), 29-35.

Nieveen, N. (1997). *Computer support for curriculum developers*. Doctoral dissertation, University of Twente, Enschede, the Netherlands.

Nieveen, N. (1999). Prototyping to reach product quality. In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 125-136). Dordrecht, the Netherlands: Kluwer Academic.

Nieveen, N., & Gustafson, K. (1999). Characteristics of computer-based tools for education and training development: An introduction. In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 155-174). Dordrecht, the Netherlands: Kluwer Academic.

Odenthal, L., Kuiper, W., Voogt, J., & Terwindt, S. (2000). Balanceren tussen ruimte en structuur: Lerarenopleiders van de Educatieve Faculteit Amsterdam als curriculumontwikkelaars [Balancing between space and structure: Teacher-trainers of the Educational Faculty of Amsterdam as curriculum developers]. *HRD Thema*, *1,* 22-32 (in Dutch).

Oser, R. L., Cannon-Bowers, J. A., Salas, E., & Dwyer, D. J. (1999). Enhancing human performance in technology-rich environments: Guidelines for scenario-based training. *Human/Technology Interaction in Complex Systems, 9,* 175-202.

Paladeau, R. E. (1998). Focusing system development efforts: The requirements definition and management process. *Proceedings of the 20th Interservice/Industry Training, Simulation and Education Conference* (pp. 89-102). Arlington, VA: National Training Systems Association (NTSA).

Patrick, J. (1992). *Training: Research and practice*. San Diego, CA: Academic Press.

Pena-Shaff, J., Martin, W., & Gay, G. (2001). An epistemological framework for analyzing student interactions in computer-mediated communication environments. *Journal of Interactive Learning Research, 12,* 41-68.

Perez, R. S., Fleming-Johnson, J., & Emery, C. D. (1995). Instructional design expertise: A cognitive model of design. *Instructional Science*, *23,* 321-349.

Perez, R. S., & Neiderman, E. C. (1993). Modelling the expert training developer. In R. J. Seidel & P. R. Chatelier (Eds.), *Advanced technologies applied to training design* (pp.261-280). New York: Plenum Press.

Pieters, J. M., & Bergman, R. (1995). The empirical basis of designing instruction. *Performance Improvement Quarterly, 8*(3), 118-129.

Pilot, A. (1997). De ontwikkeling van het begrip 'onderwijsfunctie' in ontwerpprocessen, stand van de discussie over de optimalisering van een ontwerpwijze [The development of the concept 'instructional function' in design processes, report on the discussion about optimising a way of designing]. Paper for the study days of ICO division II. Later published in W. Holleman, P. van Eijl, A. Pilot, & S. Ramaekers (Eds.) (2001), *Curriculumfuncties in discussie* (IVLOS-Mededeling nr. 64). Utrecht, the Netherlands: University of Utrecht, IVLOS (in Dutch).

Pirolli, P. L., & Greeno, J. G. (1988). The problem space of instructional design. In J. Psotka, L. Massey, & S. Mutter (Eds.), *Intelligent tutoring systems: Lessons learned* (pp. 181-201). Hillsdale, NJ: Erlbaum.

Plomp, Tj. (1992). Onderwijskundig ontwerpen: Een inleiding [Instructional design: an introduction]. In Tj. Plomp, A. Feteris, J. M. Pieters, & W. Tomic (Eds.), *Ontwerpen van onderwijs en trainingen* (pp. 19-38). Utrecht, the Netherlands: Lemma (in Dutch).

Plomp, Tj., & Feteris, A. (1991). Onderwijskundig ontwerpen als wetenschappelijke activiteit [Instructional design as scientific activity]. In S. Dijkstra, H. P. M. Krammer, & J. P. Pieters (Eds.), *De Onderwijskundige Ontwerper* (pp. 5-18). Lisse, the Netherlands: Swets & Zeitlinger (in Dutch).

Polzella, D. J., Hubbard, D. C., Brown, J. E., & McLean, H. C. (1987). *Aircrew training devices: Utility and utilization of advanced instructional facilities: Phase IV: Summary report* (Tech. Rep. No. AFHRL-87-21). San Antonio, TE: Brooks Air Force Base, Air Force Human Resources Laboratory.

Reigeluth, C. M. (1983). Instructional design: What is it and why is it? In C. M. Reigeluth (Ed.), *Instructional design theories and models: An overview of their current status* (pp. 3-36). Hillsdale, NJ: Erlbaum.

Reigeluth, C. M. (1987). Lesson blueprints based on the elaboration theory of instruction. In C. M. Reigeluth (Ed.), *Instructional theories in action: Lessons illustrating selected theories and models* (pp. 245-289). Hillsdale, NJ: Erlbaum.

Reigeluth, C. M. (Ed.). (1999). *Instructional design theories and models: Vol. 2. A new paradigm of instructional theory.* Hillsdale, NJ: Erlbaum.

Reigeluth, C. M., & Frick, T. W. (1999). In C. M. Reigeluth (Ed.), *Instructional design theories and models: Vol. 2. A new paradigm of instructional theory* (pp. 633-651). Hillsdale, NJ: Erlbaum.

Reigeluth, C. M., & Schwartz, E. (1989). An instructional theory for the design of computer-based simulations. *Journal of Computer Based Instruction, 16*(1), 1-10.

Reigeluth, C. M., & Stein F. S. (1983). The Elaboration Theory of Instruction. In C. M. Reigeluth (Ed.), *Instructional design theories and models: An overview of their current status* (pp. 335 -381). Hillsdale, NJ: Erlbaum.

Riemersma, J. B. J., Blessing, W., Bermejo Muñoz, J., Riffel, M., Fitzgerald, J., Grau, J. Y., & Krawies, J. (1998). *Elaboration/integration: Training needs analysis protocol* (EUCLID 11.1 MASTER Deliverable A1.3). Soesterberg, the Netherlands: TNO Human Factors.

Riemersma, J. B. J., Rooij J. C. G. M. van, Bosch, K. van den, Verstegen, D. M. L., Jorna, P., Pal, J. van der, Roessingh, J. J., Farmer, E., Newman, P., Hardinge, N., Evans, L., Fontenilles, H. de, Moscato, M., Blessing, W., Heinrich, J., Jakob, T., Trumm, E., Seuferling, P., Reimer, J., Wegertseder, P., González Vega, N., & Bermejo Muñoz, J. (1996). *Harmonization of experimental work* (EUCLID RTP 11.1 MASTER Deliverable D1.4). Soesterberg, the Netherlands: TNO Human Factors.

Riemersma, J. B. J., Rooij, J. C. G. M. van, Just, J., Farmer, E., Paris, Ph., Fuchs, M., Reinschlüssel, R., Jorna, P. G. A. M., & Bermejo Muñoz, J. (1994). *Reference framework* (EUCLID RTP 11.1 MASTER Deliverable D1.1). Soesterberg, the Netherlands: TNO Human Factors.

Riemersma, J. B. J., Rooij, J. C. G. M. van, Just, J., Farmer, E., Paris, Ph., Fuchs, M., Reinschlüssel, R., Jorna, P. G. A. M., & Bermejo Muñoz, J. (1995). *Common plan for field orientation* (EUCLID RTP 11.1 MASTER Deliverable D1.2). Soesterberg, the Netherlands: TNO Human Factors.

Romiszowski, A. (1984). *Instructional development: Part 1. Producing instructional systems*. London: Kogan Page.

Romiszowski, A. (1986). *Developing auto-instructional materials: From programmed texts to CAL and interactive video*. London: Kogan Page.

Rooij, J. C. G. M. van. (1997). *Onderwijskundige richtlijnen ten behoeve van de specificatie van simulatoreisen* [Didactic guidelines for the specification of simulator requirements]. (Report No. TM-97-A077). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Rooij, J. C. G. M. van. (2002, May). Digitalisatie van het trainingsveld [Digitalisation in the field of training]. *Opleiding & Training,* 17, 4-8 (in Dutch).

Rooij, J. C. G. M. van, Barnard, Y. F., Verstegen, D. M. L., Bermejo Muñoz, J., & Retamero Merino, S. (1998a). *Functional specification* (EUCLID RTP 11.1 MASTER Deliverable A2.5). Soesterberg, the Netherlands: TNO Human Factors.

Rooij, J. C. G. M. van, Barnard, Y. F., Verstegen, D. M. L., Bermejo Muñoz, J., Retamero Merino, S., Krawies, J., Hardinge, N., & Molloy, J. (1998b). *Prototype-validation studies* (EUCLID RTP 11.1 MASTER Deliverable A2.4). Soesterberg, the Netherlands: TNO Human Factors.

Rooij, J. C. G. M. van, Verstegen, D. M. L., Barnard, Y. F., Bermejo Muñoz, J., Krawies, J., Hardinge, N., & Molloy, J. (1997). *Training programme design* (EUCLID RTP 11.1 MASTER Deliverable A2.3). Soesterberg, the Netherlands: TNO Human Factors.

Rooij, J. C. G. M. van, Verstegen, D. M. L., Barnard, Y. F., González Vega, N., Bermejo Muñoz, J., Krawies, J., Hardinge, N., & Molloy, J. (1996). *Field orientation* (EUCLID RTP 11.1 MASTER Deliverable A2.2). Soesterberg, the Netherlands: TNO Human Factors.

Rowland, G. (1991). *Problem solving in instructional design.* Doctoral dissertation, Indiana University.

Rowland, G. (1992). What do instructional designers actually do? An initial investigation of expert practice. *Performance Improvement Quarterly*, *5*(2), 65-86.

Rowland, G. (1993). Designing and instructional design. *Educational Technology Research and Development, 41*(1), 79-91.

Rowland, G., & Adams, A. (1999). Systems thinking in instructional design. In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 29-44). Dordrecht, the Netherlands: Kluwer Academic.

Russel, D. (1988). IDE: The interpreter. In J. Psotka, L. Massey, & S. Mutter (Eds.), *Intelligent tutoring systems: Lessons learned* (pp. 323-350). Hillsdale, NJ: Erlbaum.

Russell, D., Moran, T., & Jordan, D. (1988). The Instructional Design Environment. In J. Psotka, L. Massey, & S. Mutter (Eds.), *Intelligent tutoring systems: Lessons learned* (pp. 203-229). Hillsdale, NJ: Erlbaum.

Ryder, M. (2003). *Instructional design models* (d.d. 16-9-2003). Retrieved September 20, 2003, from http://carbon.cudenver.edu/~mryder/itc/idmodels.html.

Salas, E., Bowers, C. A., & Cannon-Bowers, J. A. (1995). Military team research: Ten years of progress. *Military Psychology, 7,* 55-76.

Salas, E., Bowers, C. A., & Rhodenizer, L. (1998). It is not how much you have but how much you use it: Toward a rational use of simulation to support aviation training. *The International Journal of Aviation Psychology, 8*(3), 197-208.

Saroyan, A. (1993). Differences in expert practice: A case from formative evaluation. *Instructional Science, 21,* 451-472.

Schaafstal, A. M., & Bots, M. J. (1998). *De integratie van training needs analysis en logistic support analysis: een toepassing op het gebied van onderhoudsopleidingen* [The integration of training needs analysis and logistic support analysis] (Report No. TM-98-A015). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999): Learning by doing. In C. M. Reigeluth (Ed.), *Instructional design theories and models: Vol. 2. A new paradigm of instructional theory* (pp. 161-183). Hillsdale, NJ: Erlbaum.

Schank, R. C., Fano, A., Bell, B., & Jona, M. (1994). The design of goal-based scenarios. *The Journal of the Learning Sciences, 3*(4)*,* 304-345.

Schraagen, J. M. C., Breda, L. van, Schaafstal, A. M. (1994). *Simulator time and its sea time equivalence* (Report No. TM-94-C43). Soesterberg, the Netherlands: TNO Institute for Perception.

Seel, N. M., Eichenwald, L. D., & Penterman, N. F. N. (1995). Automating decision support in instructional system development: The case of delivery systems. In R. D. Tennyson & A. E. Barron (Eds.), *Automating instructional design: Computer-based development and delivery tools* (pp. 177-216). Berlin, Germany: Springer-Verlag.

Simon, H. (1969). *The sciences of the artificial* (2nd ed.). London: MIT Press.

Smuling, E. B., Brants, J., & Pilot, A. (1990). *Oriëntatie op leren en onderwijs* (2nd ed.) [Orientation on learning and education]. Groningen, the Netherlands: Wolters-Noordhoff (in Dutch).

Snelbecker, G. E. (1983). Is instructional theory alive and well? In C. M. Reigeluth (Ed.), *Instructional-design theories and models: An overview of their current status* (pp. 437-472). Hillsdale, NJ: Erlbaum.

So, R. H. Y., Chung, G. K. M., & Goonetilleke, R. S. (1999). Target-directed head movements in a head-coupled virtual environment: Predicting the effects of lags using Fitt's law. *Human Factors, 41*(3), 474-486.

Sommerlad, E., Danau, D., & Hendrikse, A. (1995). *User involvement in the development and application of learning technology* (Tech. Rep.). London: The Tavistock Institute.

Sparks, R., Dooley, S., Meiskey, L., & Blumenthal, R. (1999). The LEAP authoring tool: Supporting complex courseware authoring through reuse, rapid prototyping, and interactive visualizations. *International Journal of Artificial Intelligence in Education, 10,* 75-97.

Spector, J. M., & Muraida, D. J. (1991). Evaluating transaction theory. *Educational Technology*, *31*(10), 29-35.

Spector, J. M., & Muraida, D. J. (1997). Automating instructional design. In S. Dijkstra, N. M. Seel, F. Schott, & R. D. Tennyson (Eds.), *Instructional design: International perspectives: Vol. 2. Solving instructional design problems* (pp. 59-81). Hillsdale, NJ: Erlbaum.

Spector, J. M., Polson, M. C., & Muraida, D. J. (Eds.). (1993). *Automating instructional design.* Englewood Cliffs, NJ: Educational Technology Publications.

Spector, J. M., & Song, D. (1995). Automated instruction design advising. In R. D. Tennyson & A. E. Barron (Eds.), *Automating instructional design: Computer-based development and delivery tools* (pp. 377-402). Berlin, Germany: Springer-Verlag.

Stammers, R. B. (1986). Psychological aspects of simulator design and use. In J. Lewins & M. Becker (Eds.), *Advances in nuclear science and technology: Vol. 17. Simulators for nuclear power* (pp.117-132). New York: Polonium.

Stanton, N. (1996). Simulators: A review of research and practice. In N. Stanton (Ed.), *Human Factors in Nuclear Safety* (pp. 117-141). Southampton: Taylor & Francis.

Strachan, I. W. (2000). Technology leaps all around: Propel advances in simulators. In *Training and simulation: trends and technology review* (pp.74-79). Arlington, VA: National Defense Magazine.

Steutel, S. (1999). *Gebruikersondersteuning* [User Support]. Unpublished master's thesis, University of Amsterdam (in Dutch).

Swezey, R. W., Owens, J. M., Bergondy, M. L., & Salas, E. (1998). Task and training requirements analysis methodology (TTRAM): An analytic methodology for identifying potential training uses of simulator networks in teamwork-intensive task environments. *Ergonomics, 41*(11), 1678-1697.

Swift, D., Martindill, C., & Allender, C. (1998). Optimising specialist military training Advice within the procurement cycle. In *Proceedings of the 20th Interservice/Industry Training, Simulation and Education Conference* (pp. 102-114). Arlington, VA: National Training Systems Association (NTSA).

Tennyson, R. D. (1993). A framework for automating instructional design. In J. M. Spector, M. C. Polson, & D. J. Muraida (Eds.), *Automating Instructional Design* (pp. 191-212). Englewood Cliffs, NJ: Educational Technology Publications.

Tennyson, R. D. (Ed.). (1994). *Automating instructional design, development, and delivery.* Berlin, Germany: Springer-Verlag.

Tennyson, R. D. (1995). Instructional system development: The fourth generation. In R. D. Tennyson & A. E. Barron (Eds.), *Automating instructional design: Computer-based development and delivery tools* (pp. 33-78). Berlin, Germany: Springer-Verlag.

Tennyson, R. D. (2001). Defining core competencies of an instructional technologist. *Computers in Human Behavior, 17,* 355-362.

Tennyson, R. D., & Barron, A. E. (Eds.). (1995). *Automating instructional design: Computer-based development and delivery tools*. Berlin, Germany: Springer-Verlag.

Tennyson, R. D., & Breuer, K. (1994). ISD expert: An automated approach to instructional design. In R. D. Tennyson (Ed.), *Automating instructional design*, *development and delivery* (pp. 139-161). Berlin, Germany: Springer-Verlag.

Tennyson, R. D., & Elmore, R. L. (1995). Integrated courseware engineering system. In R. D. Tennyson & A. E. Barron (Eds.), *Automating instructional design: Computer-based development and delivery tools* (pp. 303-316). Berlin, Germany: Springer-Verlag.

Tennyson, R. D., Schott, F., Seel, N. M., & Dijkstra, S. (Eds.). (1997). *Instructional design: International perspectives: Vol. 1. Theories and models of instructional design.* Hillsdale, NJ: Erlbaum.

Tessmer, M., & Wedman, J. F. (1990). A layers-of-necessity instructional development model. *Educational Technology research and development, 38*(2), 77-85.

Tessmer, M., & Wedman, J. F. (1995). Context-sensitive instructional design models: a response to design research and criticism. *Performance Improvement Quarterly, 8*(3), 37-53.

Toomer, C. W., Selvy, D. S., Howard, R., & Davies, D. (1998). Alpha contracting: Streamlining procurement through partnership. In *Proceedings of the 20th Interservice/Industry Training, Simulation and Education Conference* (pp. 66-75). Arlington, VA: National Training Systems Association (NTSA).

Tripp, S. D., & Bichelmeyer, B. (1990). Rapid prototyping: An instructional design strategy. *Educational Technology Research & Development*, *38*(1), 31-44.

Veldhuis, G. J., & Veerman, A. L. (2002). *Evaluatie van opleidingen* [Evaluating training] (Report No. TM-02-B003). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Verstegen, D. M. L. (1997). *Trainen met gesimuleerde teamleden* [Training with simulated team members] (Report No. TM-97-B023). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Verstegen, D. M. L., & Barnard Y. F. (1998). Supporting the design of simulators from a training point of view. In *Proceedings of the 20th Interservice/Industry Training, Simulation and Education Conference* (pp. 431-441). Arlington, VA: National Training Systems Association.

Verstegen, D. M. L., & Barnard, Y. F. (2000). *Beproeving MASTER-methode in het kader van de klankbordgroep GOLM-ontwikkeling* [Evaluation of the MASTER method for the GOLM project] (Report No. TM-00-M024). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Verstegen, D. M. L., Barnard, Y. F., Hulst A. H., van der, & Sabel, A. A. (2000). *Stappenplan voor de behoeftestelling voor GOLMen* [Development of needs statements for advanced training systems] (Report No. TM-00-A020). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Verstegen, D. M. L., Barnard, Y. F., Kabel, S., Riemersma, J. B. J., Desmoulins, C., & Grandbastien, M. (2001). *Scenarios and guidelines, Final version* (ESPRIT 29175 IMAT Deliverable R.IV.3). Soesterberg, the Netherlands: TNO Human Factors.

Verstegen, D. M. L., Barnard, Y. F., & Pilot, A. (2002). *Do training simulators support instruction and practice? Results of field visits to 44 simulators* (Report No. TM-02-B005). Soesterberg, the Netherlands: TNO Human Factors.

Verstegen, D. M. L., Barnard, Y. F., & Pilot, A. (2003). *Iterative design of training programs: Results of an evaluation study* (Report No. TM-03-B005). Soesterberg, the Netherlands: TNO Human Factors.

Verstegen, D. M. L., Barnard, Y. F., & Pilot, A. (in press). *Iterative Design of Training Programs: Results of the second evaluation study*, (Report TNO-TM). Soesterberg, the Netherlands: TNO Human Factors.

Verstegen, D. M. L., Barnard, Y. F., & Rooij, J. C. G. M. van. (1999a). *De specificatie van geavanceerde onderwijsleermiddelen* [The specification of advanced training means] (Report No. TM-99-A044). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Verstegen, D. M. L., Barnard, Y. F., & Rooij, J. C. G. M. van. (1999b). *The use of simulators for training and instruction* (Report No. TM-99-B003). Soesterberg, the Netherlands: TNO Human Factors.

Verstegen, D. M. L., & Hulst, A. H. van der. (2000). Standardized development of a needs statement for advanced training means. *Proceedings of the 22th Interservice/Industry Training, Simulation and Education Conference* (pp. 1136-1144). Arlington, VA: National Training Systems Association, (NTSA).

Verstegen, D. M. L., Steutel, S., & Barnard, Y. F. (2000). Support for iteration in training program design, (Report No. TM-00-B009). Soesterberg, the Netherlands: TNO Human Factors.

Verstegen, D. M. L., Veerman, A. L., & Arend, J. G. M. van der. (2001). *Methode voor de behoeftestelling voor geavanceerde onderwijsleermiddelen: Evaluatie en verbetering* [Method for the development of needs statements for advanced training systems: Evaluation and improvement] (Report No. TM-01-A073). Soesterberg, the Netherlands: TNO Human Factors (in Dutch).

Verstegen, D. M. L., Veldhuis, G. J., Staalstra, J., & Hendriks, M. (2001). *Report on the use of training material in scenarios and organisational learning, final feedback RNLAF.* IMAT Deliverable R.I.3. Soesterberg, the Netherlands: TNO Human Factors.

Verwijs, C. (1998). *A mix of core and complementary media: New perspectives in media-decision making.* Doctoral dissertation, University of Twente, Enschede, the Netherlands.

Visscher-Voerman, I. (1999). *Design approaches in training and education: A reconstructive study*. Doctoral dissertation, University of Twente, Enschede, the Netherlands.

Visscher-Voerman, I. (2000). Ontwerpbenaderingen in opleidingspraktijken [Design approaches in training]. *HRD Thema*, *1,* 5-15 (in Dutch).

Visscher-Voerman, I., Gustafson, K.L., & Plomp, T. (1999). Educational design and development: An overview of paradigms. In J. van den Akker, R. Branch, K. L. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 15-28). Dordrecht, the Netherlands: Kluwer Academic.

Wagenberg, M. J. G. M. van. (1992). *Gericht CAD-ondersteund ontwerpen en organiseren* [Goal-oriented CAD-supported designing and organising]. Doctoral dissertation, Delft University of Technology, the Netherlands (in Dutch).

Wager, W. W., Polkinghorne, S., & Powley, R. (1992). Simulations: Selection and development. *Performance Improvement Quarterly, 5*(2), 47-64.

Wallace, Ph., & Northham, G. (1998). A training task analysis. *Modern Simulation &Training, 1998*(4), 10-19.

Walker, D. (1990). *Fundamentals of curriculum*. San Diego, CA: Harcourt Brace Jovanovich.

Wedman, J. F., & Tessmer, M. (1993). Instructional designers' decisions and priorities: A survey of design practice. *Performance Improvement Quarterly, 6*(2), 43-57.

Weert, T. J. van, & Pilot, A. (2003). Task-based team learning with ICT, design and development of new learning. *Education and Information Technologies, 8*(2), 195-214.

Weert, T. J. van, Pilot, A., Aa, P. van der, Eijl, P. J. van, Hezemans, M., Ritzen, M., & Quaak, M. J. (2002). *Taakgericht teamleren met ICT: Succesfactoren bij ontwerp en ontwikkeling* [Task-oriented team learning with ICT: Success factors for design and development]. Utrecht, the Netherlands: University of Utrecht, IVLOS, The Faculty of

Medicine, Cetis, The Faculty of Economics and Management, University Medical Center Utrecht (in Dutch).

Wells, D. (2003). *Extreme programming: A gentle introduction* (d.d. 26-1-2003). Retrieved September 20, 2003, from http://www.extremeprogramming.org.

Wetzel, D. (1993). Generative aspects of the Computer Based Educational Software System (CBESS). *Instructional Science, 21*(4), 269-293.

Winnips, J. C. (2001). *Scaffolding-by-design: A model for WWW-based learner support*. Doctoral dissertation, University of Twente, Enschede, the Netherlands.

Zagers, S. C. A. A. (2001). *Performance support for design teams: A project environment for the designers of training simulator specifications.* Unpublished master's thesis, University of Twente, Faculty of Educational Science and Technology: Enschede, the Netherlands: (in Dutch).

Appendix A: Implementation MASTER-TPD tool

The prototype MASTER-TPD tool is a customised version of DE 2.0 (1997). A more elaborate description of this prototype tool can be found in Verstegen, Steutel and Barnard (2000) and Steutel (1999). As far as possible the existing DE functionality has been used. Where this was not possible, special applications have been written, and at a few points minor changes to the TPD method have been made (see below). In all the main steps of the MASTER-TPD method the report-facility has been added as a non-obligatory sub-step. This sub-step provides users with an overview of their work at all times (in the form of a simple text document). Figure 5.2 (in § 5.2.1) gives an overview of the main steps in the MASTER-TPD tool. See Section 5.3 for a more elaborate description of the steps and sub-steps.

Designer's Edge (DE 2.0, 1997) can be customised in several ways:

- Steps can be reordered and new steps can be constructed with the Customizer. New steps are defined by a name and one or more sub-steps. For the execution of sub-steps there are three possibilities: use one of the existing sub-steps (i.e. calling a DLL from the DE library), use an existing external application (e.g., a flow-charter or an authoring tool), or use a custom-made application (i.e. calling an executable or DLL from the user's own library implemented for this purpose).
- The content of libraries can be changed with the Customizer.
- The content of on-line help files and Training Cards can be replaced.
- Other applications can read data from the DE database and add data to it. It is also possible to add tables and columns to the database.
- New wizards can be implemented with the Wizard Editor.
- The Wizard Editor can also be used to specify which data or attributes should be included in a report.

It was easy to customise the sequence of steps and sub-steps and to adapt the names of the steps. However, during the implementation of the MASTER-TPD prototype tool two major problems were encountered. Firstly, the names and attributes of objects that are used in DE cannot be changed. For the custom-made sub-steps this was not a problem: the database was expanded and new objects were defined when no matching object was available. When the MASTER-TPD tool uses one of the existing DE sub-steps, however, the names of the objects used in this sub-step cannot be changed to those originally used in the MASTER method. Sometimes these differences were not important, e.g., for the 'assessment points' the objects 'pre-test' and 'post-test' were available in DE 2.0, and for 'scenario’ the DE object 'lesson' was used. It was more difficult to find a solution for the term 'training activity' that is used in the MASTER method to describe what trainees will do to attain training goals (see § 5.3.2). The only feasible solution seemed to be to use DE's object 'training objective', explaining explicitly in on-line help and paper documentation that the intended name was 'training activity'. During the evaluation studies there were no signs that this term caused confusion, although the definition of training activities proved to be a difficult step (see § 8.3.2.3).

It is also not possible to change the attributes of existing objects. This also meant that existing DE wizards, which were not always applicable in the context of the MASTER-TPD method, could not be made unavailable in DE sub-steps. Furthermore, it was impossible to turn off some general facilities available in these sub-steps, such as the use of forms and spelling checking. To avoid confusion, explicit warnings have been included in the description of how to use to tool in the help available for each sub-step.

The second major problem was the fact that not all objects and functionalities are available in all sub-steps. DE 2.0 (1997) makes a distinction between the analysis phase and the design of lessons. In the steps of the analysis phase a list of objects is used, and in some steps a hierarchy is allowed. In these sub-steps objects can be reordered, but libraries are not available. During the design of lessons DE 2.0 uses a Course Map: a hierarchy of lessons, tests, etc. In these sub-steps libraries are available, but the objects cannot be reordered. In the MASTER-TPD prototype tool this restriction has been used to reflect the implicit boundary between de first and the second half of the method. At first, the users still work on the complete course or curriculum. This part ends with a complete set of training activities. In the second half, the focus shift to the part of training that will be covered in the future training simulator (see § 5.3).

DE's analysis phase is used for the first half of the MASTER-TPD method. To sequence training goals, for example, an existing DE sub-step is used (where, unfortunately, training goals can be presented in a flat list only). The DE sub-step that is used for the definition of training activities allows representation in a hierarchy (see Figure 5.3 in § 5.2.2). Instead of the planned library a list of example training activities has been included in the guidelines, which means that the users could not add their own training activities to the library (Note that the concept of libraries was introduced in the MASTER project to guarantee ease of use and the reuse of data over longer periods of time. This prototype is meant for use during research studies only, where this knowledge management aspect does not play a role). Custom-made applications have been implemented for three sub-steps: Select Training Strategy (see Figure 5.5), Allocate Training Media and Determine Suitability for Simulator. Thus, it was possible to make new objects for these sub-steps, and to implement an expandable list of examples that could function as a library (which was not available in this part of DE 2.0).

The DE Course Map is used for the second half of the MASTER-TPD method (see Figure 5.6 in § 5.3.4). For the definition of instructional interventions an existing sub-step and existing libraries are used. The libraries have been filled with examples of briefing elements, interventions during scenario execution and debriefing elements respectively (Note that the names of the libraries could not be changed). The Course Map is also used for the planning of assessment. This seemed the easiest solution, because the objects pre-test and post-test and the library facilities are only available here, and moving this step does not really violate the MASTER method. Thus, this step that was originally a part of the first step of the TPD method, has been implemented as an extra step: Step 4 (see Figures 5.2 and 5.4). For the description of the content of scenarios in terms of process, environment and system characteristics, a special application has been implemented since none of the DE objects had this kind of attributes (see Figure 5.7 in § 5.3.5).

Finally, an extra step had to be introduced because DE requires an explicit move from the analysis phase to the Course Map. In this MASTER-TPD step 3 users can choose between 'minimal set of requirements' or 'maximal simulator coverage', thus making the decision between a need-driven and a constraint-driven approach (see § 4.2.2) more explicit. The user is asked to decide whether to start with a core set of training activities or to focus on all training activities that are suitable for simulator training simultaneously.

A number of tables and columns have been added to the DE database and, to realise some facilities that DE does not offer, an external application has been written that runs at the same time. This external application also provides a Notepad for each sub-step (see Figure 5.3) and it registers the user's actions in a log file (see § 5.2.1).

Appendix B: Glossary of MASTER-TPD terms

Assessment:	Evaluation of the trainees' performance.
Assessment plan:	Description of when, why and how the performance of trainees will be evaluated.
Assessment point:	Point in the training sequence where assessment takes place.
Briefing:	All instructional interventions before the execution of a training scenario. For example, instruction, summary of previously learned knowledge and skills, demonstrations.
Computer Based Training:	Training programme that runs on a desktop computer.
Debriefing:	All instructional interventions after the execution of a training scenario. For example, knowledge of result, feedback, group discussion.
Course Map:	Outline of a training programme, including a description of the training strategy, a list of training activities (in training sequence), associated exercises and instructional interventions, and an assessment plan.
Detailed design:	Further developing the first global design of the future training programme into detail.
End criterion:	Performance criterion that trainees should attain at the end of (simulator) training.
Environment slot:	Description of all environment characteristics that are relevant for a training scenario, and the (ranges of) values they can take.
Global design:	Developing a first, general version of the future training programme by going through the steps of the MASTER-TPD method quickly, taking preliminary decisions and not elaborating into full detail.
Goal of assessment:	The purpose for which the performance of trainees is evaluated.
Instructional interventions:	All actions taken for the purpose of training, e.g., instruction and feedback. Instructional interventions can be executed by the instructor, by the training system (automatically), by fellow trainees, etc.
Instructional interventions: during scenario execution:	All interventions for the purpose of training during a scenario, e.g., feedback or performance registration. During an exercise instructional interventions are often triggered by the behaviour of the trainee, e.g., when the trainee makes an error (event-based) or when he takes to much time to complete a task (time-based).

Intermediate training goals:	Training goals that are inserted for the purpose of training only, e.g., when the step between two consecutive training goals is too big, or when a training goal is only partly covered at this point.
Kind of assessment:	The way that performance of trainees is measured and evaluated.
Training activity:	Activity (e.g., study or practice) performed by trainees to achieve a training goal. In the MASTER-TPD tool training activities are called: objectives of training.
Training goal:	Performance standard that trainees have to achieve at the end of training
Library:	A list of options that is provided by the MASTER-TPD tool. Users can add new items to the list and save them for future use.
MASTER method:	Method for the design of specifications for training simulators.
Notepad:	Window that can be used to take notes regarding your work in a certain sub-step.
Process slot:	Sequence of events and actions that describe what is going to happen during a training scenario, i.e. which events will occur and how the trainee is supposed to react to them.
Range of training media:	The collection of training media that is suitable for a specific training goal.
Sequencing principles:	Principles that can be used to order training goals into a well-structured and consistent training sequence.
Simulator:	An instructional product that simulates aspects of the real system and/or the task environment.
Simulator Course Map:	Course Map for the part of training that will be executed in a simulator. A simulator course map that contains only the training activities marked as 'certainly simulator training' is used to work towards the minimal set of specifications. A simulator course map that includes also the training goals marked as 'maybe simulator training' is used to work towards the specification of an ideal simulator that covers as much as possible.
Simulator training:	The part of a training programme that is executed in a training simulator.
System slot:	Description of all the features and functions of the real system that should be available to trainees during a training scenario.
Target group:	The type(s) of trainees that have to be trained, described in terms of entry level, age, work experience, educational level, etc.
Tell-me-how:	Description of how to operate the MASTER-TPD tool in a specific sub-step.

Training:	All activities that are performed to educate trainees for their future tasks, i.e. to ensure that trainees attain the training goals.
Training Card:	Short description of a step or sub-step.
Training media:	Instructional products that can be used for training.
Training Media Specification:	Third phase of the MASTER method. The purpose of Training Media Specification is to derive specifications for the training simulator from the training needs and the training programme that was designed in the previous phase.
Training Needs Analysis:	First phase of the MASTER method; the purpose of Training Needs Analysis, is to analyse what is going to be taught with the training simulator. The output is a set of training goals and a description of the target group of trainees.
Training objective:	The term that is used in the MASTER-TPD tool for training activity.
Training programme:	Description of the training to teach trainees a certain (set of) tasks, i.e. description of all activities and exercises that are executed to ensure that trainees attain the training goals.
Training Programme Design:	Second phase of the MASTER method. The goal of Training Programme Design is to specify how training is going to be provided. The result is an outline of the training programme and -for those parts that will be executed in the future simulator- prototypical training scenarios with instructional interventions.
Training resources:	The amount of money, time, etc. that is available for the design and/or the delivery of training.
Training scenario:	Exercise that is executed in a training simulator. A training scenario is described in terms of process, environment characteristics and system features and functions.
Training sequence:	The chronological order in which the training goals will be addressed during training.
Training simulator:	An instructional product that simulates aspects of the real system and/or the task environment for the purpose of training.
Training strategy:	The general approach to training, used to guide the design of the training programme.

Appendix C: Training goals of sample domain

For the evaluation studies a sample domain has been used (see § 6.2). The target group was described as (translated from Dutch):

> "Trainees are personnel from military forces, police or civil surveillance organisations; they have a few years of experience in operational jobs and are now trained for the job of thermal image analyst. Trainees are 25 to 40 years old, and have a technical background at medium vocational level."

Training goals have been derived from a preliminary task analysis executed in another project (van den Bosch, Barnard, and Helsdingen, 1999). The training goals have been formulated in more general terms, i.e. on a more abstract level and not related to any specific UAV, to avoid mentioning military or UAV-specific details and to make the domain understandable for the subjects in the evaluation study who had no previous knowledge about the domain. Training goals were given in Dutch, translations are given below:

1. Has knowledge about the tasks of the navigator.
2. Can communicate with the navigator and give indications about the desired flight path.
3. Has knowledge about the theory of thermal imagery.
4. Knows the position of the different controls.
5. Knows all tuning facilities of the camera.
6. Knows the background of the different tuning facilities.
7. Knows how the camera must be operated and tuned.
8. Can steer the camera with a joystick.
9. Can focus the camera.
10. Can decide what the right tuning of the camera is (polarity, auto-search, auto-track, field of view).
11. Can execute self-tests of the camera and do simple repairs.
12. Can save images and recall them from the buffer.
13. Knows what the differences are in striking attributes of moving and still objects.
14. Can detect still and moving objects during the flight (off line).
15. Can detect still and moving objects after the flight (off line).
16. Can identify still and moving objects during the flight (real time).
17. Can identify still and moving objects after the flight (real time).
18. Can identify 90% of elements from the following categories on still daylight images: buildings, vehicles, roads, terrains, living creatures, and industrial installations.
19. Can identify 80% of elements from the following categories on still thermal images: buildings, vehicles, roads, terrains, living creatures, and industrial installations.
20. Can identify 80% of elements from the following categories on moving daylight images: buildings, vehicles, roads, terrains, living creatures, and industrial installations.
21. Can identify 70% of elements from the following categories on moving thermal images: buildings, vehicles, roads, terrains, living creatures, and industrial installations.
22. Knows what often occurring terrain elements look like on thermal images in different situations.
23. Knows what potentially interesting objects look like on thermal images in different situations.
24. Knows the effect of different field of views on thermal images.
25. Knows the effect of differences in polarity on thermal images.
26. Knows the effect of differences in flying altitude on thermal images.

27. Knows the effect of scatter on thermal images.
28. Knows the effect of different times of day on thermal images.
29. Knows the effect of different atmospherical circumstances, including sunshine, on thermal images.
30. Knows the effect of different kinds of landscapes on thermal images.
31. Can interpret ordnance maps.
32. Can recognise correspondences between elements on ordnance maps, daylight images taken from above, and thermal images taken from above.
33. Can determine where the aeroplane is and which route is being flown based on the incoming images.
34. Knows what suitable orientation points are.
35. Knows which objects cool down slowly or quickly in which circumstances.
36. Knows the principles of interaction between objects and environment in thermal images.
37. Can apply knowledge about thermal image principles concerning the interaction between objects and environments in a large collection of objects and environments.
38. Knows that the relevance of objects cannot be deduced from how visually striking they are in thermal images.
39. Can distinguish clutter and scatter.
40. Knows that thermal image analysis consists of formulating and evaluation hypotheses.
41. Can reason critically about his/her own formulation and evaluation of hypotheses.
42. Can analyse images under time pressure.
43. Can decide on the route to be flown based on the image analysis.
44. Can decide about the tuning of the camera based on the image analysis.
45. Can translate the questions concerning a mission to the kind of information (images) that should be sought.
46. Can identify which information from the incoming images is required to answer the questions concerning the mission.
47. Can formulate a short and concise interpretation of the situation, also under time pressure.
48. Is critical about his/her own interpretation of an observed situation.
49. Can give alternative interpretations of the situation.
50. Can draw conclusions concerning the flight based on the interpretation of the situation.
51. Can draw conclusions concerning the position and tuning of the camera based on the interpretation of the situation.
52. Can draw conclusions concerning the necessity to warn headquarters based the interpretation of the situation.
53. Can write a report about the observations.
54. Can hand over the work to another thermal image analyst.

Appendix D: Task description second study

Assignment
You are participating in a TNO research study about the design of training simulators. TNO has developed a special method to design specifications for simulators: based on training goals and a description of the target group a first draft of the training programme is designed. Then you decide which part will take place in the (future) simulator and how that will happen.

A special software-tool has been developed to support the method. The tool is an adapted version of Designer's Edge, a commercial package that is used by many training companies. With this package a training programme can be designed step-by-step. Based on this design you can determine what the simulator should look like.

This week you will design a training programme with this method and tool. The results will only be used for research but you will work in a very realistic setting.

The intent is that you work individually, except on those moments that you are explicitly asked to discuss with each other. Since the differences between the designs are also interesting, we ask you not to talk about the work with each other, also not during breaks or in the evenings. If you have questions, you can ask them by e-mail.

The setting
The company Janssen wants to start producing Unmanned Aerial Vehicles. Unmanned aerial vehicles can be used for surveillance flights, for example over hostile terrains or disaster areas (oil spills at see, malfunctions in nuclear power stations, etc.). The unmanned aerial vehicle is controlled from the ground at distance, just as the camera that hangs under it. The task of the image analyst is to control the camera, to analyse the incoming pictures and to communicate with the person that navigates the unmanned aerial vehicle.

The company expects that a large part of their target customers -especially the smaller organisations, such as police corps and fire brigades- will not be able to train image analysts themselves. Therefore, the company wants to strengthen its position in the market by offering training and training means as well. The company would have to do large investments for this purpose. Therefore, TNO has been asked to execute an exploratory study.

Together with the company Janssen TNO has done a task, training goal and target group analysis. The design of training programmes and training means, however, is a creative process and many alternative solutions are possible. Therefore, a number of staff members is asked to make a first draft first. Later in the project the three best designs will be further elaborated.

Your task
You will design a training programme based on the set of training goals and the description of the target group. The task will be clearly structured, so that your design will grow step-by-step. Of course, you will get all the possible support: by e-mail you can ask questions to a panel of TNO experts.

You can mail directly by clicking on the button 'Experts' on the Notepad and choosing the right expert[42]:
- Simulator-training-specialist: for questions about the design method, its different steps and sub-steps and other questions about the design of simulators.
- Tool-specialist: for questions about the tool.
- Domain-specialist: For questions about the domain, i.e. interpretation of thermal images from unmanned aerial vehicles.
- Instructional design-specialist: for questions in the area of Instructional Science.
- Project-manager: for all other questions.

In this project there is no Director, and no Assistant-Manager.

The intent is that you ask questions by e-mail as much as possible. On the spot there is only somebody to help you with urgent technical problems. You will receive the answers by e-mail. Take into account that the experts will not always be able to answer immediately, e.g., when they are in a meeting. Send an e-mail to the Project-manager in case of problems.

Programme

We will keep to working days: from 9-5 with half an hour lunch break.

3 July:	Introduction and explanation of the method (morning).
	First global design (afternoon).
4 July:	Discussion.
	Elaborate design in detail.
5 July:	Presentation by domain expert.
	Elaborate design in detail.
6 July:	Discussion.
	Work further on final design.
7 July:	Prepare presentation (morning).
	Presentation of designs to each other and to TNO-experts (afternoon).
	Drinks.

[42] The four subjects working in Soesterberg could not e-mail directly from the TPD tool because of technical problems. They were given an alternative version with the e-mail addresses of the experts (tmuu-simulator-training-specialist@tm.tno.nl etc.).

Glossary

Note that terms specific for the TPD phase of the MASTER method are not included in this glossary. They can be found in Appendix B.

Algorithm:	Procedure that defines completely and unambiguously how to apply knowledge and guarantees a solution (Mettes and Pilot, 1980).
Briefing:	A training simulator session usually starts with a briefing in order to activate prior knowledge and prepare for training activities (Farmer, Jorna, Riemersma, van Rooij, and Moraal, 1999), in most cases the briefing is provided by an instructor. During the briefing the instructor explains the goal of the training scenarios and -if necessary- provides additional theory (see § 2.3.2.2).
Debriefing:	A training session with a simulator usually ends with a debriefing to consolidate learning and to evaluate the session in terms of training goals (Farmer et al., 1999); usually provided by an instructor or commander (see § 2.3.2.2).
Design cycles:	The MASTER method can be used at different levels of detail for different purposes. Going through the steps on a rather global way might be sufficient to write a needs statement early on. However, the designers will have to go through the steps again in more detail in order to design detailed specifications for a training simulator. In the MASTER-TPD tool two design cycles are defined explicitly. This is a form of process-oriented help, see for an example Section 5.4.3.4.
Designer's Edge (DE 2.0):	Commercial tool that supports the instructional development process; supports the analysis of the objectives of learning and the design of courses and lesson scripts, mainly focused on the design of CBT and the development of scripts for CBT lessons (DE 2.0, 1997, see also Figure 5.1).
Design process:	See instructional design process.
Field training:	Practice with the operational system (for those aspects that cannot be covered with the simulator).
Functional specifications:	Functional specifications for a training simulator describe what the (future) simulator should be able to do. In other words: they specify a simulator that can fulfil the user requirements of the different types of users. In the development process functional specifications are one step before the technical specifications: they specify on a behavioural level what the simulator should be capable of, not how this can or should be technically realised.

Heuristics:	Rules or procedures that are not always complete or unambiguous and do not guarantee a solution, but improve the chance that a solution will be found (Mettes and Pilot, 1980).
Hints:	Hints on the Notepad attached to each step are a way to encourage designers to make adequate notes. Hints are derived from process-oriented guidelines and given in the form of direct questions. In the MASTER-TPD tool Hints are a form of process-oriented help, see for examples Section 5.4.3.2.
Instruction	In the context of simulator training the term instruction is used in broad sense for all activities executed to enhance training efficiency, including explanations, demonstrations, feedback, etc. Instruction can be provided by an instructor, fellow trainees or automatically by the training system.
Instructional design:	The design of courses and/or specifications for the required instructional products is often referred to as instructional design. In this thesis the term instructional design is used for the product of the design phase (cf. Figure 1.1). To refer to the process the term instructional design process is used (see § 3.1).
Instructional designer:	Someone who specialises in instructional design.
Instructional design models:	Prescriptive models that indicate *what* instruction and instructional products should be like (adapted from Reigeluth, 1983), i.e. which instructional method will lead to the described outcomes under which conditions (see § 3.3.2).
Instructional development:	In this thesis the term instructional development used for the whole process of Analysis, Design and Production of instructional products (cf. Figure 1.1), see Section 3.1.
Instructional development models:	Prescriptive models that indicate *how* to develop instruction and instructional products (adapted from Reigeluth, 1983). These models prescribe a process of systematically designing, sequencing, implementing, evaluating and constantly monitoring instruction with the intent of improving its quality and effectiveness, and thereby enhancing learning (Gustafson, 1991). See Section 3.3.3.
Instructional design process:	In this thesis the term instructional design process is used for the activities executed by the instructional designer during the instructional design phase (see § 3.1). It refers to the process of deciding what methods of instruction are best for bringing about desired changes in the learner's knowledge and skills for a specific course content and a specific learner population (Reigeluth, 1983).
Instructional product:	A product that is made to help a learner acquire some knowledge or skills (Merrill, 2001). Examples are text-books, assignments, lesson plans to used by a teacher in a

	classroom, Computer Based Training (CBT), E-learning packages, simulations or simulators. In learning processes the learner can interact with a number of instructional products and other human beings (teachers, coaches, other learners, etc.).
Instructor:	In the context of simulator training the term instructor is used (instead of teacher).
Iteration:	In a general sense iteration means: executing activities again (van Wagenberg, 1992). The simplest form is the repetition of the same activity, e.g., regulating the thermostat of the heating: users can adapt the thermostat according to their wishes once or more every day. Iteration can, however, also mean monitoring the results of one activity and -based on those results- choosing the next activity out of a range of activities. In the context of specifying training simulators with the MASTER method I define iteration as going back to a design activity that has already been executed. There are different ways to iterate and different reasons for iteration (see Figure 5.8 in Section 5.4.1.
MASTER method:	ISD-based method tailored to the design of specifications for training simulators.
MASTER project:	Military Applications of Simulator and Training concepts based on Empirical Research, EUCLID project (no. 11.1) to investigate the areas for improvement and to work on a new method for design and development for simulators; executed by a consortium of 23 partners from five European countries between 1993 and 1998 (see § 2.2.1 and § 4.2.1).
Problem solving:	The movement from one knowledge state to another, by the application of operators, until the desired end state is reached (Greeno, Korpi, Jackson and Michalchik, 1990).
Procedural simulations:	Simulations used to teach a sequence of actions to accomplish some goal (Alessi and Trollip, 2001; Schwartz and Reigeluth, 1989), see Section 1.2.1.
Process-oriented guidelines:	Guidelines to help users to manage their own design process and to take process-oriented decisions. In the MASTER-TPD tool the process-oriented guidelines are divided into guidelines for the global design phase and guidelines for the detailed design phase. Process-oriented guidelines are a form of process-oriented support, see for examples Section 5.4.3.1.
Process-oriented support:	Support to help users to manage their own design process and to take process-oriented decisions. Process-oriented support should help users to keep track of what they are doing and to decide what to do next, e.g., by making the relationships between different steps in the design process explicit, helping users to recognise triggers that might

	necessitate iteration and making it easy for them to go back to previous steps in the design process.
Simulator training:	The part of the course or curriculum that is covered with a training simulator.
Simulator:	A more or less realistic replication of an operational system and its environment, including the displays and controls that are available to the operator(s) (Farmer et al., 1999). More specifically, Gagné (1962) states that a simulator: attempt to represent a real situation, provides certain controls over the situation, and deliberately omits aspects of the real operational situation.
Technical specifications:	Technical specifications for a training simulator describe how the functional specifications can or should be technically realised.
Trainee:	In the context of simulator training the term trainees is used instead of learner or student.
Training:	In the context of simulator training the term training is used for practice, i.e. repeated attempts to perform a task, or more exactly, trainee activities executed in interaction with a training scenario intended to promote learning (Farmer et al., 1999).
Training simulator:	A device that simulates certain aspects of the real system and/or its environment for the purpose of training (Riemersma et al., 1994). A training simulator is equipped with specific facilities to support training and instruction (Farmer et al., 1999).
Training scenario:	Term used for exercise in a training simulator. One training session contains one or more training scenarios, which can vary in length from a couple of minutes for simple procedures to several days (see § 2.3.2.2).
User requirements:	User requirements for a training simulator describe what users need. For a training simulator there are several types of users, e.g.: trainees, instructors, and training designers. They have different needs: trainees need facilities to learn and practice, instructors need facilities to support the learning process and training designers need facilities to develop training scenarios and training programmes. See also functional specifications for a training simulator.
Warnings:	Warnings pop up automatically when events that should trigger iteration can be automatically recognised or measured. In the MASTER-TPD tool Warnings are a form of process-oriented help, see for examples Section 5.4.3.3.

List of abbreviations

ADL:	Advanced Distributed Learning Initiative
CBT:	Computer Based Training
DE:	Designer's Edge
EUCLID:	European Co-operation for the Long term in Defence
FAQ:	Frequently Asked Questions
ID:	Instructional Design
IOS:	Instructor Operator Station
IPISD:	The Interservice Procedures for Instructional Systems Development
ISD:	Instructional System Development; or Instructional System Design
ITS:	Intelligent Tutoring Systems
LTSA:	Learning Technology Systems Architecture
MASTER:	Military Applications of Simulator and Training concepts based on Empirical Research
PPMCC:	Pearson Product-Moment Correlation Coefficient
RNLA:	Royal Netherlands Army
S:	Seconds (in Figures)
SAT:	System Approach to Training
SCORM:	Shareable Content Object Reference Model
SD:	Standard Deviation
SME:	Subject Matter Expert
SRCC:	Spearman Rho Correlation Coefficient
TNA:	Training Needs Analysis
TMS	Training Media Specification
TPD:	Training Programme Design
UAV:	Unmanned Aerial Vehicle

Curriculum Vitae

Daniëlle Verstegen was born at the 27th of May 1968 in Nuth. She followed secondary education at the Bernardinuscollege in Heerlen, where she obtained a gymnasium diploma in 1986. She then moved to Nijmegen to study Psychology. After the first two years she specialised in Cognitive Science. For her master's thesis she went to Milano (Italy) for seven months, where she worked at Dida*El in the area of Intelligent Tutoring Systems. She obtained a master's degree in Cognitive Science at the Radboud University of Nijmegen in 1992 (formerly Catholic University of Nijmegen). After her graduation she stayed in the Philippines for one year.

Daniëlle Verstegen started working at TNO Human Factors in March 1994, in the group and later the department of Training and Instruction. She conducted and participated in research studies regarding, amongst others, training with simulators and the retention of procedural skills. Some of these studies were executed within the European defence research project MASTER. In the European ESPRIT project IMAT she investigated the reuse of material from electronic manuals for learning purposes. She has led of a series of projects regarding the specification of advanced instructional products, such as simulators, CBT and E-learning. In this context she developed the SLIM method for the development of needs statements of advanced learning means. As project leader she is responsible for research and consultancy projects for various customers, for organising and maintaining collaboration with other partners and for coaching junior researchers. Alongside her other project work, she worked on this thesis since 1998. Her current research interests include iteration in (instructional) design processes, supporting novices in the design of instructional products, and the design of innovative and effective learning activities.

Summary

The operation and maintenance of technically advanced systems, such as cars, aeroplanes, power plants, computers, and radar systems, are critical and complex tasks that are not easy to learn. A large amount of training is necessary to reach adequate performance. Sometimes operation and maintenance tasks are taught with the operational system under supervision, e.g., learner drivers practise in a real car in the normal traffic environment with a driving instructor. In many cases, however, other ways of learning these tasks are more effective and efficient, using advanced instructional products such as training simulators.

Ideally, specifications for all instructional products should be based on what learners need to learn, and how they can learn it most effectively and efficiently. During the specification process analysis, design and production can be performed in a sequential, cyclic or overlapping way. In practice, this process is disturbed by many 'pragmatic' factors, such as conflicting constraints, interference from management, personnel changes in design teams and technological progress leading to new possibilities. Reacting to these disturbances can make the process chaotic, but not reacting to them will certainly lead to solutions that do not fulfil the requirements or will not be accepted in the organisation. Therefore, instructional design is an iterative process.

Within the limited budgets available, the optimal choice and efficient use of training simulators and other instructional products is important. This puts more pressure on the design of such products. There is a tendency to give much attention to technical aspects. This is understandable since training simulators contain advanced technology that has to be carefully designed and constructed. The key question, however, is not what is technologically possible or most advanced, but what can optimise the learning process. Thus, the challenge for the designers of training simulator specifications is to design the best solution from an educational point of view, while at the same time taking other factors into account and reacting promptly to changing conditions. This has led to the formulation of the general problem addressed in this thesis as:

How can instructional designers be supported during the iterative design process in order to design effective instructional products?

To answer this question I have done research in the area of training simulator specification. For my research I have used the MASTER method for the design of functional specifications for training simulators. The MASTER method consists of three main phases: Training Needs Analysis, Training Programme Design and Training Media Specification. For evaluation studies I have used the second phase, because in this phase the designers can exploit the potential advantages of simulators for instruction and training.

This thesis has two parts. The first part (Chapters 1 to 4) concerns the design of training simulators. More specifically, the research questions in this part are:

1. Which factors determine the (kind and level of) fidelity of training simulator?
2. Do training simulators have the necessary facilities to support training and instruction?
3. Which issues should be addressed during the design of functional specifications for training simulators from an educational point of view (product requirements)?
4. How are functional specifications for training simulators designed?
5. Which aspects of the design process need to be supported (process requirements)?
6. To what extent does the MASTER method fulfil the product requirements defined in Question 3 and the process requirements defined in Question 5?

The second part (Chapters 5 to 10) focuses on the iterative aspect of the design process and on the usability of the design phase of the MASTER method: Training Programme Design (TPD), with the following research questions:

7. How can designers be supported during the iterative design of simulator training programmes?
8. How is iteration used in the design process?
9. Can novice designers design simulator training programmes with the MASTER-TPD method in an adequate way?
10. Do subjects use the different forms of support that are offered?

In Chapter 2, I look at how simulators are currently used for training by analysing data of 44 field visits to training simulators. From an educational point of view the minimum level of fidelity should be determined by the training goals that are covered with the training simulator. For those tasks or task aspects that are trained with the simulator, at least functional fidelity should be guaranteed. Only in three of the 44 simulators in the field study is there evidence that the specifications for the simulator have been deduced from training goals. There are strong indications that specifications for training simulators usually focus on technical factors and physical fidelity, or that simulators are acquired (or delivered with the operational system in a package deal) without further specification of training. Many interviewees were not satisfied with the fidelity of their simulator, although it was not always clear to what extent this limited fidelity affected learning and transfer. The extra advantages that simulator training can offer were not fully exploited, and the facilities to support instructors were often limited. To avoid such shortcomings in future, I define issues that should be addressed during the design of functional specifications for training simulators from an educational point of view (i.e. the product requirements).

In Chapter 3, I take a closer look at the process of designing functional specifications for training simulators. I compare the characteristics of this process with descriptive research about the process of instructional design in general and describe various types of methods and models that are available in the literature to support the instructional design process. Designing functional specifications for training simulators is a complex and inherently iterative process. Designers will have to take preliminary decisions based on incomplete and insecure information, and they will have to review and alter their decisions frequently when new information becomes available. There is also a more fundamental reason for iteration: in an ill-structured design problem the sub-problems are not unrelated and can, therefore, never be solved in isolation. Moreover, it is impossible to know beforehand which ideas will lead to a good solution and which will not. Experienced instructional designers have the problem solving skills necessary to organise, monitor and adapt the design process. However, the designers of functional specifications for simulators are usually not experts, and descriptive research shows that novices are not very good at organising their own design process. Current prescriptive models do not sufficiently support this aspect of the design process.

In Chapter 4, I describe the MASTER method for the design of functional specifications for training simulators. The MASTER method fulfils many of the requirements defined in Chapter 2 and 3, but it does not actively support iteration. The MASTER method had also not yet been applied in practice. Subsequently, in Chapter 5, I describe how the design phase of the MASTER method was implemented in an existing commercial package. In this prototype tool support for iteration has been implemented in the form of guidelines, hints, warnings and predefined design cycles. The prototype tool has been used for two studies in order to evaluate the usability of the MASTER-TPD method and the role of iteration in the design process. In both studies subjects used the MASTER-TPD method and prototype tool to design a draft simulator training programme with a realistic case.

In Chapter 6, I describe the design and most important results of the first evaluation study, an empirical study where subjects worked on their training programme designs at home at their own pace over a three month period. The time that subjects had dedicated to the design task appeared to have an important influence on the quality of the results. Therefore, the second study was executed in a controlled setting where subjects worked on the design task for a fixed amount of time. To study iteration in more detail, events that could trigger iteration were introduced during the design process. In this second study, described in Chapter 7, the time on task was controlled. Actually, it was limited to less than the average time on task in the first study. However, the average number of iterations is more or less the same in both studies. The results of both studies confirm that iteration is inherent to the design process: all subjects iterate. The number of iterations varies considerably between subjects, but there is no relation with the quality of the resulting training programme designs. Apparently, different design styles -with more or less iteration- can lead to good results. Some subjects work out a good concept and then work through the method step-by-step, going back to previous steps only when it is necessary. Others come to a good design by going back and forth between steps more often. Maybe they leave more decisions open, go to the next steps to elaborate the design partly, get new ideas, and then go back to work on the initial concept again, etc. The results of both studies show that frequent iteration can be part of an ineffective design style as well, perhaps indicating a lack of overview. Weak designers will probably make more errors and will have less insight into the possible consequences of early decisions. Good designers will have to iterate less to repair errors, but they might use iteration for another purpose: to check whether they have designed the most optimal solution or to consider possible alternatives.

In Chapter 8, I look into the results of both evaluation studies in more detail and discuss the usability of the MASTER method and the different forms of support that were available to subjects. The subjects are quite positive about all forms of support. During both studies subjects used all types of help information offered by the tool. There are no indications of problems with accessing help information in the tool and there are no clear relationships between the amount of time spent on help and the quality of the subjects' training programme designs, the number of iterations, and the quantity of notes subjects make. Therefore, the hypothesis is that differences in time spent on help are either caused by the way time was measured or, more probably, by differences between subjects in the amount of time they needed to read the information. The expectation beforehand was that it would be hard to convince users to make notes. This proved not to be true during both evaluation studies: all subjects made notes, and often -but not always- those notes were sufficient to follow the subjects' reasoning. The quantity of notes varied a lot and, although there is no linear relationship, there are indications that subjects who performed better made more notes, especially in the second study. Finally, in both studies the subjects asked experts questions by e-mail. The fact that experts were not always able to answer immediately frustrated some subjects who claimed that they could not continue with their task without the answers to their questions. This can be a sign that they found it hard to deal with uncertainty and incomplete information.

Both evaluation studies show that the subjects can design simulator training programmes with the MASTER-TPD method in an adequate way. In many respects the situation during the two evaluation studies was similar to the situation that the MASTER method was developed for: novice designers specifying a training simulator. The case that was used was realistic in size and complexity. There are, however, some differences that may have consequences for the validity of the results: the subjects in the two evaluation studies have less knowledge about the domain than the target users are expected to have, and more background knowledge about instructional design and development theory. Furthermore, instructional designers often work in the context of a design team in a real design process, and this was not the case during the studies. There are no indications that a lack of domain knowledge prevented subjects from fulfilling their design task. They had some problems with simulator-specific terms and activities such as the design of training scenarios, but these could be solved with some extra examples and explanations. The results of two smaller studies with target users, reported in Chapter 9, indicate that an extensive background in instructional design and development theory is not required for using the MASTER-TPD method, at least not when experts are present to answer questions about the method and the tool. The studies presented in this thesis do not systematically investigate the differences between working individually and working in a design team. However, the results of the studies with target users, discussed in Chapter 9, indicate that working in a team makes the process more complicated and more iterative. Working in a team also introduces a social aspect to the design process. When the team does not always work together in the same place additional facilities to support communication and co-ordination are required.

In Chapter 10, I summarise the answers to the research questions. I propose a list of events that can trigger iteration. The first four triggers are caused by, or evolve in interaction with, the design process itself: the discovery of missing input, the need to repair errors, new insights based on work later on in the design process, and new ideas of the designer(s). The other triggers originate from outside. New information that is relevant to the design process becomes available, or other people bring in new opinions or arguments that the designer(s) were previously unaware of. A third trigger from outside is related to the procedures for the acquisition of (expensive) instructional products that are enforced in many large organisations. Based on the literature and empirical research reported in this thesis, I propose a number of measures that can help designers to deal with iteration during the design process. These measures include explicitly structuring the design process by using systematic methods and defining design cycles, explicitly planning triggers for iteration such as decision points and review moments, specific advice regarding dealing with iteration, and forms of knowledge management that make it easier for designers to iterate. Finally, I discuss to what extent the results of my research can be generalised to other instructional design situations, and give some suggestions for further research.

Both descriptive studies in the literature and my own studies showed that iteration during the instructional design process is unavoidable. Having to iterate is not necessarily bad. On the contrary, iteration can improve the result and there are even indications that truly innovative design requires a good deal of iteration. So what designers need is support to deal with iteration. Developing such support requires more insight into the different reasons for iteration, and more insight into when iteration is desirable and when it is not. The list of triggers for iteration, and the measures to support iteration, that are proposed in Chapter 10, provide a common language to discuss iteration in instructional design and can serve as a basis for more research. For practitioners the descriptions of triggers for iteration might be interesting because it can give them more insight into, and grip on, this difficult aspect of their design task.

Samenvatting

De bediening en het onderhoud van technisch geavanceerde systemen, zoals auto's, vliegtuigen, energiecentrales, computers en radarsystemen, zijn belangrijke en complexe taken. Om deze taken adequaat te leren uitvoeren is veel training nodig. Soms worden zij onderwezen met behulp van het echte systeem, zoals bijvoorbeeld het leren autorijden in een echte auto in de echte verkeersomgeving onder toezicht van een rij-instructeur. In veel gevallen zijn andere manieren van leren, met behulp van geavanceerde leermiddelen zoals trainingssimulatoren, echter effectiever en efficiënter.

Idealiter zouden specificaties voor leermiddelen gebaseerd moeten zijn op wat de leerlingen moeten leren en hoe zij dat het meest effectief en efficiënt kunnen leren. De analyse-, ontwerp- en productiefases kunnen sequentieel, cyclisch of overlappend worden uitgevoerd. In de praktijk wordt dit ontwerpproces echter verstoord door allerlei 'praktische' factoren, zoals tegenstrijdige en veranderende eisen, interrupties door het management, personeelswisselingen in het ontwerpteam en technologische vooruitgang die leidt tot meer mogelijkheden. Door te reageren op deze verstoringen kan het ontwerpproces chaotisch worden, maar als de ontwerpers er niet op reageren zal hun ontwerp zeer waarschijnlijk niet voldoen aan de eisen of niet geaccepteerd worden binnen de organisatie. Daarom is onderwijsontwerp een iteratief proces.

Binnen de doorgaans beperkte budgetten is een optimale keuze en efficiënt gebruik van de doorgaans dure trainingssimulatoren en andere technisch geavanceerde leermiddelen belangrijk. Daardoor staat het ontwerpproces extra onder druk. Ontwerpers van specificaties voor trainingssimulatoren zijn geneigd om veel aandacht te schenken aan technische aspecten. Dat is begrijpelijk, want trainingssimulatoren bevatten geavanceerde technologie die zorgvuldig moet worden ontworpen en gemaakt. Het gaat er bij het ontwerpen van specificaties echter niet om wat er technisch gezien mogelijk is, maar om wat optimaal is voor het leerproces. De uitdaging voor ontwerpers is om de beste oplossing vanuit onderwijskundig gezichtspunt te ontwerpen. Tegelijkertijd moeten zij echter rekening houden met alle andere eisen en adequaat reageren op veranderende omstandigheden. Deze overweging heeft geleid tot de centrale probleemstelling in dit proefschrift:

Hoe kunnen onderwijsontwerpers ondersteund worden tijdens het iteratieve ontwerpproces opdat zij de meest effectieve leermiddelen ontwerpen?

Om deze vraag te beantwoorden is onderzoek gedaan op het gebied van het ontwerpen van specificaties voor trainingssimulatoren. Daarbij is gebruik gemaakt van de MASTER-methode die bestaat uit drie hoofdfases: de analyse van de trainingsbehoefte (Training Needs Analysis), het ontwerpen van trainingsprogramma's (Training Programme Design) en het opstellen van specificaties voor een trainingssimulator (Training Media Specification). De tweede fase is gebruikt voor evaluatiestudies, omdat ontwerpers met name in deze fase de potentiële voordelen van simulatortraining kunnen uitbuiten.

Dit proefschrift bestaat uit twee delen. Het eerste deel (hoofdstuk 1 tot en met 4) betreft het ontwerpen van trainingssimulatoren. De onderzoeksvragen in dit deel zijn:

1. Welke factoren bepalen (het niveau en de aard van) de natuurgetrouwheid van de trainingssimulator?
2. Hebben trainingssimulatoren de benodigde faciliteiten om instructie en training te ondersteunen?

3. Welke onderwerpen moeten vanuit onderwijskundig oogpunt aan bod komen tijdens het ontwerpen van specificaties voor trainingssimulatoren (product requirements)?
4. Hoe worden functionele specificaties voor trainingssimulatoren ontworpen?
5. Welke aspecten van het ontwerpproces moeten worden ondersteund (process requirements)?
6. In hoeverre voldoet de MASTER-methode aan de eisen die gesteld zijn in vraag 3 en 5?

Het tweede deel (hoofdstuk 5 tot en met 10) is gericht op het iteratieve aspect van het ontwerpproces en op de empirische evaluatie van de tweede fase van de MASTER-methode: Training Programme Design (TPD). De onderzoeksvragen in dit deel zijn:
7. Hoe kunnen ontwerpers ondersteund worden in het omgaan met iteratie tijdens het ontwerpen van trainingsprogramma's?
8. Hoe wordt iteratie gebruikt in het ontwerpproces?
9. Kunnen onervaren ontwerpers op adequate wijze trainingsprogramma's ontwerpen met de MASTER-TPD-methode?
10. Gebruiken de proefpersonen de verschillende soorten ondersteuning die worden aangeboden?

In hoofdstuk 2 worden de resultaten van 44 veldbezoeken geanalyseerd om te kijken hoe simulatoren op dit moment gebruikt worden. Vanuit onderwijskundig oogpunt zou het minimaal vereiste niveau van natuurgetrouwheid bepaald moeten worden door de leerdoelen die met simulatortraining moeten worden gehaald. Slechts in 3 van de 44 gevallen zijn er aanwijzingen dat de specificaties voor de simulator inderdaad zijn afgeleid van leerdoelen. Er zijn sterke aanwijzingen dat de specificaties meestal met name gericht zijn op technische factoren en (fysieke) natuurgetrouwheid, en dat simulatoren in veel gevallen worden aangeschaft (of samen met de operationele systemen worden afgeleverd in een 'package deal') zonder dat er verder wordt nagedacht over trainingsprogramma's. Veel geïnterviewden zijn niet tevreden met de natuurgetrouwheid van hun simulator, alhoewel het niet altijd duidelijk is of de beperkingen in de natuurgetrouwheid daadwerkelijk negatieve gevolgen hebben voor het leren en de overdracht van het geleerde naar de werksituatie. De potentiële voordelen van simulatortraining worden lang niet altijd benut en de faciliteiten om instructeurs te ondersteunen zijn vaak erg mager. Om dit soort tekortkomingen in de toekomst te voorkomen zijn een aantal aspecten gedefinieerd die -vanuit onderwijskundig oogpunt- in elk geval aan bod zouden moeten komen tijdens het ontwerp van specificaties voor trainingssimulatoren.

In hoofdstuk 3 staat het ontwerpproces centraal. De karakteristieken van het ontwerpen van specificaties voor trainingssimulatoren worden vergeleken met de resultaten van beschrijvend onderzoek naar het ontwerpen van specificaties voor leermiddelen in het algemeen. Vervolgens worden op basis van literatuuronderzoek verschillende soorten van modellen en methodes voor onderwijsontwerp beschreven. Het ontwerpen van specificaties voor trainingssimulatoren is een complex en iteratief proces. Ontwerpers zullen voorlopige beslissingen moeten nemen op grond van informatie die niet compleet en niet altijd zeker is en zij zullen deze beslissingen steeds moeten heroverwegen wanneer er nieuwe informatie beschikbaar komt. Er is ook een meer fundamentele reden waarom het ontwerpproces iteratief moet zijn. Ontwerpproblemen worden gekarakteriseerd als 'slecht-gestructureerd'. De deelproblemen hangen nauw met elkaar samen en kunnen daarom niet los van elkaar worden opgelost. Bovendien is het onmogelijk om van tevoren te voorzien welke ideeën tot een goede oplossing kunnen leiden en welke niet. Ervaren onderwijsontwerpers beschikken over de benodigde vaardigheden om het ontwerpproces te organiseren, te beheersen en waar nodig aan te passen. De ontwerpers van specificaties voor trainingssimulatoren zijn echter doorgaans geen ervaren onderwijsontwerpers en de resultaten van beschrijvend onderzoek in de literatuur tonen aan dat beginnende ontwerpers niet erg goed zijn in het organiseren van hun eigen ontwerpproces. De huidige modellen en methodes voor onderwijsontwerp ondersteunen dit aspect onvoldoende.

In hoofdstuk 4 wordt de MASTER-methode voor het ontwerp van functionele specificaties voor trainingssimulatoren beschreven. De MASTER-methode voldoet aan veel van de eisen die gesteld worden in hoofdstuk 2 en hoofdstuk 3, maar zij ondersteunt het omgaan met iteratie onvoldoende. Bovendien was de methode ten tijde van het onderzoek nog nooit in de praktijk gebruikt. In hoofdstuk 5 wordt beschreven hoe de tweede fase van de MASTER-methode (TPD) is geïmplementeerd in een commercieel software pakket. In deze prototype MASTER-TPD-tool is ondersteuning voor iteratie geïmplementeerd in de vorm van richtlijnen, hints, waarschuwingen en twee expliciet gedefinieerde ontwerpcycli. Het prototype is gebruikt in twee empirische studies om de bruikbaarheid van de MASTER-TPD-methode te evalueren en om iteratie in het ontwerpproces nader te bestuderen. Tijdens beide studies werd een realistische case gebruikt. De taak van de proefpersonen was om met behulp van de methode en de tool een eerste versie van een trainingsprogramma voor de toekomstige simulator te ontwerpen.

Hoofdstuk 6 bevat de belangrijkste resultaten van de eerste evaluatiestudie. Dit was een empirische studie waarin de proefpersonen gedurende drie maanden thuis in eigen tempo werkten aan een ontwerp voor een trainingsprogramma. Er bleek een samenhang te zijn tussen de tijd die proefpersonen besteed hadden aan deze taak en de kwaliteit van hun resultaten. Daarom is ervoor gekozen om de proefpersonen in een tweede evaluatiestudie allemaal even lang en onder toezicht te laten werken. Bovendien werden in deze tweede studie een aantal interventies uitgevoerd om iteratie uit te lokken. De belangrijkste resultaten van deze studie worden beschreven in hoofdstuk 7. Hoewel de proefpersonen in de tweede studie minder tijd aan hun ontwerp konden besteden dan gemiddeld in de eerste studie is het aantal iteraties in beide studies ongeveer gelijk. De resultaten van beide studies bevestigen dat iteratie inherent is aan het ontwerpproces: alle proefpersonen itereren. Het aantal iteraties varieert aanzienlijk en hangt niet samen met de kwaliteit van de resultaten. Blijkbaar kunnen verschillende ontwerpstijlen -met meer of minder iteratie- leiden tot goede resultaten. Sommige proefpersonen gaan stap voor stap door de MASTER-TPD-methode om tot een goed concept te komen en beginnen dan weer van voren af aan om hun concept verder uit te werken. Tussendoor gaan zij alleen terug naar eerdere stappen als dat echt nodig is. Anderen komen tot een goed resultaat door juist veel meer heen en weer te springen tussen de stappen. Misschien laten zij meer beslissingen open, kijken eerst naar de volgende stappen om ideeën te ontwikkelen en gaan dan weer terug om verder te werken aan hun ontwerp. De resultaten van beide studies maken bovendien duidelijk dat veel itereren ook een kenmerk kan zijn van een niet-effectieve ontwerpstijl. In dat geval wijst veel iteratie wellicht op een gebrek aan overzicht. Bovendien maken minder goede ontwerpers waarschijnlijk meer fouten, en zij hebben minder inzicht in de gevolgen van hun beslissingen. Goede ontwerpers hoeven minder te itereren om fouten te herstellen, maar zij gebruiken iteratie wellicht op een andere manier: om te controleren of zij echt de meest optimale oplossing hebben ontworpen en om eventuele alternatieve oplossingen te overwegen.

In hoofdstuk 8 worden de resultaten van beide evaluatiestudies in detail geanalyseerd om de bruikbaarheid van de MASTER-TPD-methode en de verschillende soorten ondersteuning te evalueren. De proefpersonen waren redelijk enthousiast over alle soorten ondersteuning die hen tijdens de studies ter beschikking stonden. Zij gebruikten alle soorten informatie uit de hulpfuncties in de tool. Er zijn geen aanwijzingen dat proefpersonen problemen hadden om deze hulpfuncties te vinden, en er zijn geen duidelijke relaties tussen de hoeveelheid tijd die proefpersonen besteedden aan de hulp en de kwaliteit van hun resultaten, de hoeveelheid iteraties of de hoeveelheid aantekeningen die zij maakten. Daarom is de hypothese dat verschillen in tijd besteed aan hulp werden veroorzaakt door de meetmethode of, waarschijnlijker, door verschillen in de tijd die proefpersonen nodig hadden om de informatie te lezen en te verwerken.

Vooraf was de verwachting dat het moeilijk zou zijn om gebruikers te overtuigen van het belang van het maken van aantekeningen. Dat bleek niet zo te zijn. Tijdens beide studies maakten de proefpersonen aantekeningen en deze aantekeningen waren vaak -maar niet altijd- voldoende om hun redeneringen te volgen. De hoeveelheid aantekeningen verschilde aanzienlijk en de proefpersonen die beter presteerden leken meer aantekeningen te maken dan de proefpersonen die minder goed presteerden. Tenslotte stelden de proefpersonen in beide studies vragen aan experts per e-mail. Het feit dat de experts die vragen niet altijd meteen beantwoordden, frustreerde sommigen van hen. Zij klaagden dat zij niet door konden werken zonder het antwoord op hun vragen. Dit zou kunnen betekenen dat zij het moeilijk vonden om om te gaan met de onzekerheid en de onvolledige informatie die karakteristiek zijn voor een dergelijk ontwerpproces.

Uit beide evaluaties blijkt dat de proefpersonen in staat zijn om op adequate wijze trainingsprogramma's voor een toekomstige simulator te ontwerpen. In veel opzichten leek de situatie tijdens de studies op de situatie waarvoor de MASTER-methode is ontwikkeld: onervaren ontwerpers die werken aan specificaties voor een trainingssimulator. De case die werd gebruikt was realistisch qua grootte en complexiteit. Er zijn echter een paar verschillen die de validiteit van de conclusies kunnen aantasten. De proefpersonen wisten minder over het domein dan echte gebruikers waarschijnlijk zouden weten, en zij hadden meer achtergrondkennis op het gebied van onderwijsontwerp. Bovendien werken onderwijsontwerpers in werkelijkheid vaak in de context van een ontwerpteam en dat was niet het geval tijdens de evaluatiestudies. Er zijn geen aanwijzingen dat het gebrek aan domeinkennis de proefpersonen heeft gehinderd bij het uitvoeren van hun taak. Zij hadden wat problemen met specifieke termen en ontwerpactiviteiten, zoals het ontwerpen van trainingsscenario's, maar deze konden worden opgelost met extra uitleg en enkele voorbeelden. De resultaten van twee andere, kleinere studies met echte gebruikers geven aan dat een theoretische achtergrond op het gebied van onderwijsontwerp waarschijnlijk geen vereiste is voor het gebruik van de MASTER-TPD-methode, in elk geval niet wanneer experts aanwezig zijn om vragen over de methode en de tool te beantwoorden (zie hoofdstuk 9). In de studies in dit proefschrift wordt niet systematisch onderzocht wat de verschillen zijn tussen individueel ontwerpen en het werken in een ontwerpteam. In de studies met echte gebruikers zijn er echter aanwijzingen dat het ontwerpproces nog complexer en iteratiever wordt wanneer ontwerpers in een team werken. Bovendien krijgt het ontwerpproces dan ook een sociaal aspect. Wanneer het team niet altijd samen op dezelfde plaats werkt, zijn extra faciliteiten nodig om de communicatie en coördinatie te ondersteunen.

In hoofdstuk 10 worden de antwoorden op de onderzoeksvragen samengevat. Er wordt een lijst opgesteld van gebeurtenissen, zgn. triggers, die iteratie kunnen veroorzaken. De eerste vier triggers worden veroorzaakt door, of ontstaan uit de interactie met het ontwerpproces zelf: het ontdekken van ontbrekende input, de noodzaak om fouten te herstellen, nieuwe inzichten gebaseerd op werk dat later in het ontwerpproces wordt verricht en nieuwe ideeën van de ontwerpers. De andere triggers komen van buitenaf. Er kan nieuwe informatie die relevant is voor het ontwerpproces beschikbaar komen en andere personen kunnen nieuwe meningen of argumenten aandragen waaraan de ontwerpers zelf nog niet hadden gedacht. De derde trigger van buitenaf heeft te maken met de procedures die in veel organisaties verplicht zijn voor de aanschaf van (dure) leermiddelen.

Op basis van het literatuuronderzoek en de resultaten van de evaluatiestudies die in dit proefschrift worden gerapporteerd, wordt een aantal maatregelen voorgesteld die ontwerpers kunnen helpen bij het omgaan met iteratie tijdens het ontwerpproces. Deze maatregelen betreffen het expliciet structureren van het ontwerpproces door het gebruik van systematische ontwerpmethodes en het definiëren van ontwerpcycli, het zelf plannen van triggers voor iteratie zoals beslispunten en reviews, specifieke adviezen over het omgaan met andere triggers die iteratie kunnen veroorzaken, en vormen van kennismanagement die het gemakkelijker maken om te itereren. Tenslotte wordt besproken in hoeverre de resultaten van het onderzoek in dit proefschrift gegeneraliseerd kunnen worden en er worden suggesties voor verder onderzoek gedaan.

Zowel uit mijn eigen studies als uit beschrijvende studies in de literatuur blijkt dat iteratie tijdens het ontwerpproces onvermijdelijk is. Het is ook niet slecht, integendeel: door iteratie kan het ontwerp steeds verder verbeterd worden. Er zijn zelfs aanwijzingen dat een behoorlijke hoeveelheid iteratie noodzakelijk is om tot een innovatief ontwerp te komen. Wat ontwerpers dus nodig hebben is meer inzicht in wanneer iteratie wenselijk is en wanneer niet. De in hoofdstuk 10 beschreven lijst met triggers en de voorgestelde maatregelen om iteratie te ondersteunen, vormen een begrippenkader om over dit aspect van het ontwerpproces te discussiëren en zij kunnen dienen als basis voor verder onderzoek. Voor onderwijsontwerpers uit de praktijk kunnen de beschrijvingen van triggers voor iteratie interessant zijn omdat zij inzicht geven in, en meer grip geven op dit moeilijke aspect van het ontwerpproces.

Index