Instructional Science (2006) 34:481–517
DOI 10.1007/s11251-005-3346-0

Which events can cause iteration in instructional design? An empirical study of the design process

D.M.L. VERSTEGEN[1,*], Y.F. BARNARD[2] & A. PILOT[3]
[1]*Rotterdam Institute for Social Policy Research (RISBO), Erasmus University Rotterdam, P.O. Box 1738, 3800 DR Rotterdam, The Netherlands;* [2]*EURISCO International, Toulouse, France;* [3]*IVLOS, Utrecht University, Utrecht, The Netherlands (*Author for correspondence, e-mail: verstegen@risbo.eur.nl)*

Received: 16 November 2004; in final form: 4 August 2005; accepted: 21 September 2005

Abstract. Instructional design is not a linear process: designers have to weigh the advantages and disadvantages of alternative solutions, taking into account different kinds of conflicting and changing constraints. To make sure that they eventually choose the most optimal one, they have to keep on collecting information, reconsidering continuously whether their own decisions are still justified in the light of the latest insights. We have studied the role of iteration during instructional design. For our research, we have used an ISD-based method for the specification of training simulators. During our empirical evaluation study, we introduced five events that are likely to cause iteration. The results show that the quality of the designs is not directly related to the amount of iteration. We conclude that there are different kinds of iteration, triggered by different kinds of actions and events. We propose a list of triggers for iteration some of which originate from outside (new information, new opinions/arguments and acquisition procedures); others are caused by, or evolve from interaction with the design process itself (discovery of missing input, need to repair errors, new insights based on work later on in the design process, and new ideas of the designers).

Keywords: causes for iteration, instructional design, iteration, iterative design process, specification of instructional products, support for iteration, training simulators

Introduction

Instructional design is not a linear process: designers have to weigh the advantages and disadvantages of alternative solutions, taking into account different kinds of – often conflicting – constraints. To make sure that they eventually choose the most optimal one, they have to keep on collecting information, reconsidering continuously whether their own decisions are still justified in the light of the latest insights. Frequently, preliminary decisions have to be taken before all the necessary input is available, which means that earlier decisions may have to be revised later on when new information becomes available or when further analysis leads to new insights. Moreover, it is impossible to know beforehand which ideas will lead to a good solution and which will not. Finally, the instructional design process is often complicated because of changes in the constraints, stakeholders with different interests or changes in the design team. Reacting to these 'disturbances' can make the design process chaotic, but not reacting to them will certainly lead to solutions that do not fulfil the users' needs optimally. Managing such an iterative design process adequately is not an easy task, especially not for less experienced instructional designers. Prescriptive instructional development models[1] recognise the iterative nature of design, but do not sufficiently support iteration.

In a general sense, iteration means executing activities again (Van Wagenberg, 1992). In its simplest form, it is a procedural repetition of the same activity, e.g. adapting the thermostat of the heating once or more every day. During design processes, however, the designer monitors the results of one activity and chooses the next activity based on those results. In this context, we elaborate the definition of iteration to: going back to a design activity that the designer has already worked on before. Van Wagenberg differentiates different types of iteration that can also happen during instructional design:

1. Iteration to correct errors: reviewing the concept design and correcting errors.
2. Iteration to improve the design: fundamental revisions to get to a better design.
3. Iteration as repetitive activity: repeating the same design activity.
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 iteration seems desirable. If errors are made, they should be corrected as soon as possible. In this context it is important to realise that in design processes not all errors can be avoided because not all information is available from the start and constraints and demands can change. Fundamental revisions to get to a better design are also desirable (provided that the required time and resources are available). Expert designers always consider several alternative solutions in parallel (see below) and more of this kind of iteration may be required to get to innovative designs. 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). One would expect, therefore, that more iteration would lead to better designs.

In this article, we describe the results of a study on iteration in instructional design. The long-term aim of our research is explorative in nature: we want to clarify the role of iteration in instructional design and develop measures that can be taken to help designers deal with this difficult aspect of the design process. In order to get more grip on the issue of iteration we need more insight in the concrete events that trigger iteration. We will address the following research questions regarding iteration:

- Is the amount of iteration related to the quality of the resulting designs?
- What are the subjects' opinions about the interventions planned to evoke iteration?
- Do subjects iterate when expected, i.e. after the events introduced to evoke iteration? Do subjects iterate at other times?

In the next paragraph, we discuss findings from the literature and argue as to why design has to be an iterative process. We also briefly discuss how different instructional design methods deal with iteration. Subsequently, we introduce the domain that we have used: the design of training simulator specifications. Then we describe the method and results of our study. In the discussion we present a framework that can serve as basis for further research and for the definition of measures that can be taken to help designers – especially novices – to deal with this difficult aspect of the design task.

The nature of the instructional design process

Instructional design in practice

In practice, instructional design is often done by teachers or instructors who are not professional instructional designers. The results of a large field study in the area of training simulators indicate that these nonprofessionals are not inclined to use a systematic approach for the design of instructional products (Verstegen, 2003; Verstegen et al., 2002). This is confirmed by descriptive research in other settings (see, e.g. Gilbert, 1999; Odenthal et al., 2000; Verstegen et al., 2001). Several studies indicate that following a systematic design approach improves the resulting designs (van Berlo, 2002; Hoogveld et al., 2001), preferably combined with a relational or communicative approach that challenges stakeholders to become involved and reveal their perceptions of the ideal curriculum (Kessels, 1993).

Professional instructional designers design specifications for instructional products in a more systematic way, which is not surprising since they are trained to do so. They use prescriptive design and development models (Rowland & Adams, 1999; Saroyan, 1993), although not very accurately: they conduct some of the prescribed activities, but not all of them and not always the same ones (Le Maistre & Weston, 1996; Visscher-Voerman, 1999). 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, for instance depending on the size of the projects, the constraints impeded by clients and preferences of the designers (see also Pieters & Bergman, 1995; Visscher-Voerman, 2000; Wedman & Tessmer, 1993). In a survey Holcomb et al. (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. It seems that professional instructional designers use different strategies to design and develop, and often perform activities in an integrated way (Visscher-Voerman, 1999; Visscher-Vaerman et al., 1999).

The problem solving perspective

Like all design tasks, instructional design can be seen as a problem-solving task. Greeno, et al. (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 also no predefined solution process, no algorithm or established set of procedures. Designers have to employ heuristics, problem solving, creativity and decision making (Nelson et al., 1988). Finally, it is not always clear what exactly is the problem. The designer's task includes analysing 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). The ill-structured nature of design problems will be illustrated by the characteristics of the specific instructional design task used in our study (see below).

Goel & Pirolli (1989, 1992) defined generic features of design tasks such as the need for problem structuring, distinct phases in the problem solving process, the use of problem decomposition and incremental design, the application of a limited commitment mode control strategy and the combination of top-down and bottom-up strategies. Analysing think-aloud protocols, they found that these features indeed distinguish design tasks from nondesign problem solving tasks. Other protocol studies confirm that design processes can be seen as a form of problem solving and report similar characteristics, e.g. Greeno et al. (1990) and Rowland (1991, 1992) for instructional design, Hamel (1990) in the domain of architecture, and Blessing (1994) in the field of mechanical engineering. Braha & Maimon (1997) compare the design process to scientific research.

Design tasks cannot be solved with general knowledge only, but require a substantial amount of professional, domain-specific knowledge (de Jong, 1986; Mettes et al., 1981): 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 likely to lead to a solution. This is certainly true for the design of specifications for complex instructional products: designers need extensive knowledge of the task domain and the technical systems to be used as well as knowledge about learning and instructional design. Not only the presence of different kinds of knowledge is important, but also its organisation. The problem solver should be able to retrieve and apply 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.

Expert vs. novice instructional designers

Experienced instructional designers have the knowledge and skills required for design tasks. They collect knowledge about the domain and about the current design problem and construct an elaborate representation of the design problem; when necessary they come back to adjust or elaborate the problem analysis later on (Rowland, 1991, 1992). Experts start to consider solutions early on, but generate a number of alternative possibilities and do not commit to one solution until much later. Experienced instructional designers also reuse (parts of) solutions of other design problems that they have encountered in the past (Perez et al., 1995). Less experienced instructional designers, on the other hand, are inclined to spend little time in the analysis phase and to commit themselves to one solution early on (Perez et al., 1995; Rowland, 1991, 1992; Kerr, 1983). Moreover, several studies showed that designers find it hard to discard ideas or (parts of) solutions (e.g. Greeno et al., 1990). 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 to follow a more structured approach (Perez et al., 1995). With experience, designers develop cognitive structures or schemata which can guide the design process and can be used to categorise, consider and store information (Nelson et al., 1988). Thus, experts are able to recognise patterns, infer relationships, disregard irrelevant information and recall similar problems from the past.

Instructional development models

Because instructional design problems are ill-structured in nature, the instructional design process is inherently iterative. Especially novice designers need support for such a complex process. There are many prescriptive instructional design and development methods that provide support in different ways. They can provide a framework for gathering, organising and storing information, especially in combination with software tools. Furthermore, instructional design methods prescribe what instruction and instructional products should be like under certain conditions and, thus, give a theoretical foundation for important decisions (see, e.g. overviews by Alessi & Trollip, 2001; Kearsley, 2003; Ryder, 2003, Reigeluth, 1999). Further support can be provided by different kinds of decision-support systems (e.g. van der Hulst et al., 1999; Seel et al., 1995). Some methods offer more 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 (e.g. Halff, 1993; van Merriënboer, 1997; Merrill & the ID2 Research Group, 1996).

Instructional development models support the organisation and management of the design process. Many of these models are based on Instructional System Design and prescribe which instructional design activities should be executed in which order (see e.g. overviews by Andrews & Goodson, 1980; Dijkstra, 2001; Gustafson, 1991). Although many ISD models agree that the design process should be iterative and adaptable, they are often represented as a linear sequence of steps with feedback loops only after formative or summative evaluations. Prototyping approaches follow a more pragmatic approach of repeated try-out and revision of design prototypes (e.g. Tripp & Bichelmeyer, 1990). This is, however, only possible when prototypes can be constructed easily and quickly and can be adapted at minimal costs. A more fundamental problem is that rapid prototyping models also do not specify when and how to iterate. Furthermore, according to Tripp and Bichelmeyer, rapid prototyping models may encourage informal design methods. 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 (protocol studies showed that less experienced designers are definitely inclined to follow such a strategy, see above). Thus, instructional development methods permit iteration, but do not explicitly support this aspect. 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.

Domain of empirical study: Designing a training simulator

Most studies about instructional design are either post-hoc questionnaires or rather short laboratory studies using think-aloud protocols. However, to study the role of iteration in design we needed a realistic design task, complex enough to enforce iteration, and a longer time span that would allow us to study this aspect. We were looking for a realistic setting in which we could introduce events that we expected to evoke iteration in a realistic way. Since the long-term aim was to specify support for an iterative design process, the focus was on less experienced designers who need this support most. We opted for a domain for which we could create an authentic setting and were sure that we could easily find a homogeneous group of subjects with equally little knowledge about the design problem: designing specifications for a training simulator.

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). It is equipped with additional facilities to support training and instruction (Farmer et al., 1999). Training simulators are developed to learn complex tasks using technically advanced systems, 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. Two examples are shown in Figure 1.

Van Rooij (1997) describes that the design and 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. In our study we focussed on the design of functional specifications which describe – on a behavioural level – what the future simulator should be able to do in order to fulfil the needs of the future users. The development of functional simulator specifications is mainly an instructional design task. Later on, the functional specifications will have to be translated into technical specifications, but this requires technical expertise and is usually not the task of an instructional designer. For the design of functional specifications a rapid prototyping approach is usually not feasible for designing training simulators because developing fully functional prototypes is expensive, and because

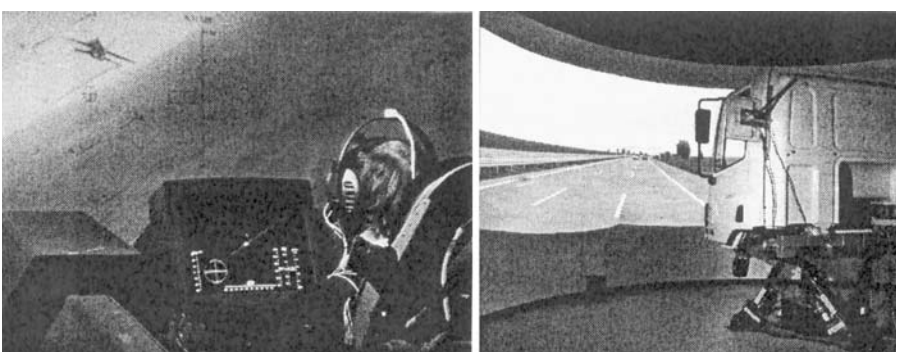

Figure 1. On the left: a flight simulator with 3D outside view. On the right: a truck simulator on a moving base. (Pictures W. van Dijk, Copyright 2003, TNO Human Factors, reprinted with permission).

design and implementation are done by different parties. This puts more pressure on the design phase.

The people responsible for the design of functional specifications for training simulators often 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. However, they usually have knowledge and experience in related fields. They are domain experts, experienced instructors, designers of class-room teaching materials, etc. They might also have experience with simulator training in the role of trainee and/or instructor.

Characteristics of the design of simulator specifications

The design of training simulator specifications has eight main characteristics (Verstegen, 2003; Verstegen & van der Hulst, 2000):

(1) Complex domains: simulators are often used for training complex tasks. The task analysis is difficult and a large amount of data will have to be collected.
(2) Different kinds of expertise are required for the design of specifications for training simulators, such as domain, technical and educational knowledge.
(3) Multiple constraints: such as budget, time, and resources.
(4) Contradicting constraints and interests: with a cheaper simulator, for example, more instructors may be needed; or instructors may be aiming for an optimal system and management for the cheapest solution.
(5) Diversity in design problems: projects differ in purpose, complexity, the number of people involved, the type of constraints and aspects that play an important role, etc.
(6) Design processes are long and time-consuming: in the Royal Netherlands Army, for example, 4–5 years is considered normal (van Rooij, 2002), and Kincaid (1997) gives an example of a major equipment acquisition project that lasted for 15 years.
(7) Information is incomplete and insecure: demands and constraints may change during the design process; designers will have to make preliminary decisions when it is not yet clear what is feasible both technically and financially.
(8) Timing problems: postponing the design of specifications until all the necessary information is available is impossible for two reasons: specifications are necessary at an early stage to claim

financial resources, and training should be implemented and available before the operational systems are delivered.

These characteristics reflect the ill-structured nature of design problems and the iterative nature of the instructional design process. 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 5-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..."

The case used for the study

The design problem used for the study described in this article concerned the training of analysts of thermal images that are collected with Unmanned Aerial Vehicles (UAV). The image analysts also have to operate the controls to turn around the camera to find interesting spots and hold them in sight, they have to tune the camera, they have to communicate with the navigator that controls the UAV and they have to report important information to their superiors, following correct procedures. This design problem, a real case taken from a project for a client (see van den Bosch et al., 1999), is selected because it is realistic in size and complexity, which will force subjects to look for additional information and to iterate during the design process. It concerns a domain that none of the subjects had much knowledge about (to avoid differences between subjects), but is not too difficult and can be solved within a realistic time span for an evaluation study. See Verstegen (2003) for a more elaborate description of the design problem.

The subjects used the MASTER method, a specific ISD model for the specification of training simulators (Farmer et al., 1999). More specifically, we have used the middle phase of that method: Training Program Design (MASTER-TPD method), which is similar to the design phase of other ISD-based methods. Using only a part of the MASTER method made it possible to let subjects work with a case of realistic size and complexity within a reasonable time span. At the same time, executing the TPD takes long enough to study iteration.

The MASTER-TPD method has been implemented in a prototype tool, a customised version of Designer's Edge (DE 2.0, 1997). The prototype tool guides users through the TPD in six main steps, each divided in sub-steps. It also provides Notepads with each step and logs the users' actions. The MASTER-TPD tool supports iteration because intermediate results (including the subjects' notes) remain available. Part of the guidelines provided for each design activity concern this aspect of design and provide, for example, information about the relations between different design activities, strategic advice about how certain activities can best be executed, and indications for taking notes. A small number of warnings appears automatically when an event that is a likely cause for iteration can be measured. For example when only a small part of a training is allocated to the simulator, the designer is asked whether the (probably expensive) simulator is really necessary for such a limited part of the training.

Figure 2 shows a screen print of the MASTER-TPD prototype tool (see for a more elaborate description Verstegen, 2003; Verstegen et al., 2000).

In summary, the subjects' task was to design a training program with the MASTER-TPD method and tool for the purpose of designing functional specifications for a training simulator to be used for training the analysts of thermal images from UAVs. The input was a

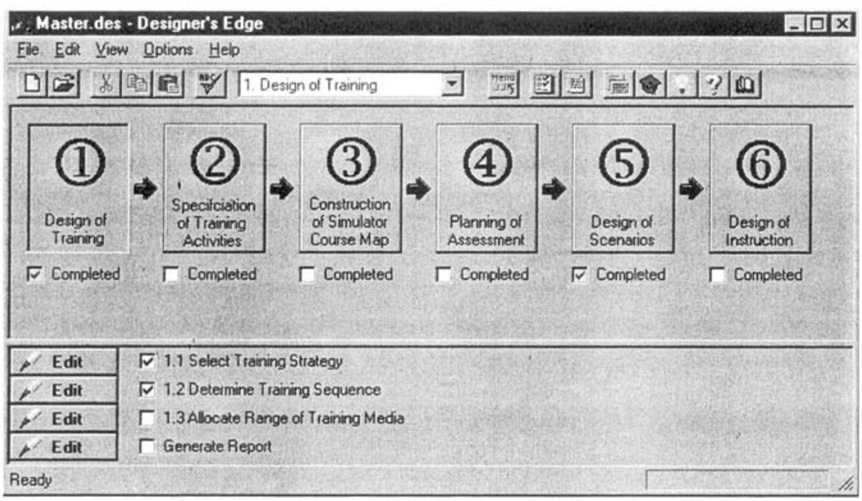

Figure 2. The prototype MASTER-TPD tool, a customised version of DE 2.0 (1997).

set of training goals and a description of the target group of trainees, derived from the needs analysis conducted in the real project (van den Bosch et al., 1999). We already had gained experience with both the tool and the task in an earlier study, in which we found that the task was not too difficult and that subjects could produce a design product that was judged as reasonable or good by both instructional design and domain experts (Verstegen 2003; Verstegen et al., 2003).

Method

Subjects

Real designers of training simulator specifications are professionals who cannot participate in a long study. Because these professionals usually do not have much experience in designing training simulators we took other novice instructional designers as subjects: five students of Educational Science and Technology (third year or higher), and one student of Educational Psychology (final year), one subject who had just graduated in Applied Communication Science (having done courses in Educational Science and Technology). The last subject had a few years experience in instructional design and research, but also no experience with the design of training simulators. Subjects were either paid for their participation in this study (i.e., the students) or completing the assignment in work time.

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 and some of them also had some experience in related fields. Table 1 gives an overview of the background of the eight subjects:

Setting

Subjects were asked to imagine that they were temporary employees 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 and different kinds of experts could be approached for advice through e-mail at any time. This is comparable to a situation that often occurs

Table 1. Background of the subjects

Subjects had	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 (during courses)	5	3
Experience in designing instruction	3	5
Experience in teaching	1	7
Experience as researcher/research assistant	3	5
Experience with training simulators	0	8

in practice where, for example, an instructor or Subject Matter Expert could be responsible for this part of the design process with the input of a task analysis performed by himself or by somebody else. He or she is likely to be in contact with (domain and other) experts and peers who can answer questions or give comments.

The subjects worked under supervision for one full week, i.e. five successive working days. The results of a previous study (Verstegen, 2003; Verstegen et al., 2003) showed that this amount of time was enough for subjects to design a concept simulator training program (obviously, more time would be required to elaborate the training program to the level of detail required for simulator specification or the implementation of training scenarios). In the previous study, we found a strong relation between time on task and the quality of the designs (Verstegen et al., 2003). To rule out this factor, time on task was fixed in this study.

Materials

The subjects used:

– the MASTER-TPD method and tool (see above),
– normal e-mail software and their own e-mail address (to receive e-mail), and
– Microsoft NetMeeting, version 2.1 (1997) for the chat sessions.

Additionally the following materials were provided to the subjects:

– 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,

– 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).
– an explicit description of the assignment by e-mail,
– a description of the domain, and
– assignments regarding the two design phases and the other interventions (by e-mail and on hand-outs).

The full text of the assignment, the domain description, the glossary and all e-mails can be found in the appendices of Verstegen et al., (2004).

Data gathering methods

As stated above, the research questions for this study were:

– Is the amount of iteration related to the quality of the resulting designs?
– What are the subjects' opinions about the interventions planned to evoke iteration?
– Do subjects iterate when expected, i.e. after the events introduced to evoke iteration? Do subjects iterate at other times?

The MASTER-TPD tool stored the subjects' training program designs (final and intermediate versions), and the subjects' notes on the Notepads. The quality of the final designs was rated by three experts.[2] The number of iterations was deducted from log files collected by the MASTER-TPD tool. The log files contained actions with time stamps and also allowed us to analyse whether subjects iterated after events introduced to evoke iteration. The subjects' opinions were gathered using three questionnaires containing both open questions and questions on a 5-point scale (which can be found in Verstegen et al., 2004):

– questionnaire 1 concerned the subjects' background,
– questionnaire 2 concerned 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 concerned the subjects' opinions about the MASTER-TPD method, its usability and usefulness.

Other data collected during the study were:

– the questions sent to experts by e-mail,
– the files with the subjects' comments on each others' designs (result of intervention 2),

- audio-tapes of the discussions with the domain expert (results of intervention 3),
- log files of the chat sessions (results of intervention 5),
- video-tapes of the subjects' presentations on the last day (see below),
- the slides that subjects used during their presentations, and
- ratings by two experts present during the presentations (not involved in other analyses) on the quality of the presentation, their impression of the quality of the training program design, and the quality of the explanation and 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.

The results of this study will be presented in tables and figures. We have computed correlations only as a way to find possibly interesting results, since the statistical power of this study (i.e. limited number of subjects) does not allow the valid use of correlation coefficients. The results of these computations are reported in Notes.

Procedure

The subjects worked at two locations. The introduction was given by the first author of this article, on Friday afternoon at one location and on Monday morning at the other location. All subjects started on the design task after the coffee break on the first day, after receiving the background materials (described above under materials). During the last afternoon, all subjects presented their design at a group meeting in the presence of a panel of two simulator training experts (on one location). In the instructions, sent by e-mail beforehand, subjects were asked to explain briefly but clearly their training program design. They were told that the experts would be mainly interested in the explanation and defence of important design decisions, such as the choice of what is covered with simulator training and what is 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.

The study took place from July 3 to 7, 2000 (Monday to Friday). Subjects worked full time (i.e., 8 h a day minus half an hour lunch break and two coffee breaks of 15 min), for a total of 35 h, including a 1.5 h introduction on the first day and 4 h for the presentation of the designs on the last day. 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. The e-mail addresses of different kinds of experts (on simulator training, on the domain/case, on instructional design, on the MASTER-TPD tool, or for other questions) were provided. The complete text of all e-mails and handouts can be found in the appendices of Verstegen et al. (2004). Table 2 gives an overview of the procedure.

Interventions to evoke iteration

During the introduction, we explained to the subjects that an instructional design process is usually not a linear process and that they could always go back to previous steps or design activities to make changes. Five events that are likely causes for iteration were introduced during the study.

**Intervention 1: Scheduled global and detailed design phases*
This intervention was meant to mimic the kind of iteration that is often enforced by organisational procedures for the procurement of expensive (learning) systems. The subjects were asked to go through the MASTER-TPD method quickly first and to send in a first draft at the end of the first day. Then they were asked to execute the design activities in more detail during the rest of the week. Subjects were not obliged to follow this advice, but they were told that making a global design first would give them an overview of the decisions that they would have to make during the design of a training program and to get a rough idea of what their own design was going to look like. The division between global and detailed design is supported by explicit descriptions in the Manual, on-line help and guidelines. During the global design phase subjects received reminders by e-mail roughly every hour, indicating which step they should be working on in order to finish their global design at the end of the day. In the detailed design phase subjects worked at their own pace.

** Intervention 2: Peer review*
This intervention was meant to mimic the way that designers get comments from colleagues. On the second day the subjects were asked to

Table 2. Procedure

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

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. The instructions 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? Comments were typed into a text file that was saved on the desktop. The maximum time for this intervention was one hour.

* *Intervention 3: Contact with domain expert*
This intervention was meant to mimic the fact that some relevant information about the design problem is lacking at the start of the design process, and only becomes available later on. On the third day subjects listened to a short presentation from a domain expert (the second author of this article who had been involved in the task analysis that served as a basis for the case description) and then asked questions. Contact was made by speaker phone in two groups for the two locations.

* *Intervention 4: E-mail message from customer about resources*
This intervention was meant to mimic an occasion were the conditions or requirements change during the design process. 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 customer had already had contacts with several companies interested in the training for UAV thermal image analysts that the company was going to offer. The customer also explained that it would probably be very difficult to find good instructors. Therefore, it was important that it would be possible to provide training with a minimum number of instructors, even if that meant that the simulator would be more expensive.

* *Intervention 5: Discussion in chat-session*
This intervention was meant to mimic contacts with other people that do not directly concern the design problem but are related to designing simulator specifications. On the fourth day (in the afternoon) subjects participated in a chat-session using their own computer and Microsoft NetMeeting, version 2.1 (1997). To make sure that the subjects would see a reasonable amount of new arguments, subjects were divided in two groups to discuss about different topics. One group 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 other group 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?

Afterwards subjects could read a text file of the other group's discussion that they received by e-mail. The maximum time for this intervention was 1 h.

Results

In this article, we will focus on the discussion of results relevant for the research questions. Other results, e.g. concerning the usability of the MASTER-TPD method and the abilities of novice designers are discussed in Verstegen (2003) and Verstegen et al. (2004).

Quality of the designs

Only subjective measures of the quality of the subjects' results are possible since the subjects' products concern only a part of simulator development: the concept training program design is still far away from detailed (technical) specifications or actual training. It is not possible to try out the specified simulator with trainees to evaluate whether it fulfils the training needs. For the purpose of this research, three experts with knowledge about and experience in specifying simulator training (the first two authors of this article and a colleague) judged the quality of the subjects' training program designs. They spent 2–3 days on this task and used (anonymous) prints of the subjects' training program designs, including an overview of the results of all design activities and all the notes that subjects made on the Notepads. The raters were also not aware of the number of times that subjects had iterated. The raters judged the overall quality of the subjects' training program 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 a 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? Therefore, the three raters ranked the subjects' training program designs from the best (8) to the least good (1). The concordance between the raters is reasonably high (Kendall coefficient of concordance: $W = 0.67$, Average ranking correlation = 0.51; statistically significant[3]). The average scores were ranked again to come to an 'Average quality ranking' to use for comparison with other results below. The results are given in Table 3:

Even though the subjects' time was limited, all subjects designed a first concept of a simulator, training program and presented their design at the end. Three 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. During the presentations three other subjects indicated that they had not enough time. In the raters' opinion all subjects had taken their task seriously. They judged the

Table 3. Quality of training program designs[a]

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

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

quality of all eight training program designs as acceptable to good. Two of the raters (Rater 1 and 2) had also been involved in an earlier study where subjects were allowed to work at their own pace and spent, on average, more time on the same design task (see Verstegen, 2003; Verstegen et al., 2003). However, according to these two raters, the average quality of the designs of this study was at least as high. Some of the raters' general impressions are worth mentioning:

- 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. Many of them used a mastery learning concept, and did not show a good understanding of the learning process: this is a task that cannot be learned by simply practising until perfection (Rater 2).

The average quality ranking of the final designs differs considerably from the ratings of the presentations by the expert panel. This can mean two things:

- the differences between subjects in presentation skills overshadowed the differences in the quality of the designs, or
- the experts in the panel have different opinions about the quality of training programs than the raters in Table 3.

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.[4] Table 4 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.

Figure 3 shows that the number of iterations is not related to the quality of the results.[5]

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 they would have

Table 4. Number of iterations

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

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

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 (seven 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 (three subjects), and

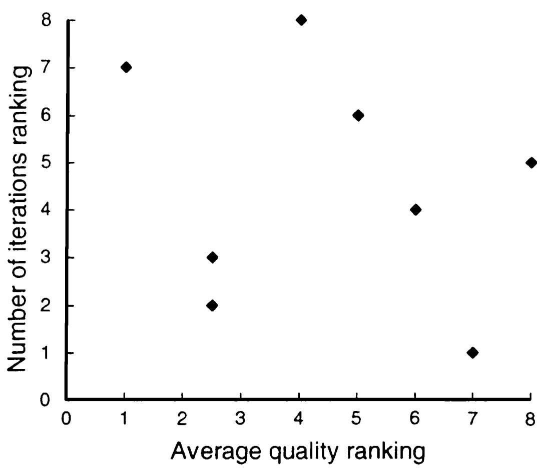

Figure 3. The ranking based on the number of iterations (from Table 4) vs. the average quality ranking (from Table 3).

– made mistakes initially because misunderstood the purpose of one step of the MASTER-TPD method (one subject).

Most (seven subjects) said that the MASTER-TPD method facilitates iteration. Some subjects are not positive about the tool in this respect: in later steps the tool does not show all the training activities, but only those that will take place with the simulator; and sometimes – when the tool is not used as intended – sequencing information (e.g. the sequence of training scenarios) gets lost when subjects iterate.

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 5 gives an overview of the results.

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 (two subjects) or longer than one day (two subjects). For the peer review two experts (the first author of this article and a colleague) analysed the content of the reviews and categorised the comments. 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 subjects gave between 4 and 19 comments (average: 12.0; standard deviation 5.3). 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 went back to a relevant design activity and two subjects

Table 5. Iteration after interventions

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

* Immediate iteration to a step linked to the intervention
× Iteration to a step linked to the intervention within three actions and within 1 h
[a] Subjects were working on relevant design activity and continued after intervention
[b] Subject went forward to relevant design activity

continued working in a step for which the new information from the domain experts was also expected to have impact. It was striking that there were not many questions (two questions from one group and six questions from the other group). Some subjects came up with new questions after the contact with the domain expert had finished; the supervisor advised them to ask any remaining questions to the domain expert by e-mail. One of the supervisors reported that after this intervention some of the subjects were frustrated by the thought of having to modify their design again.

The message from the project manager with new information about the customer's resources at the beginning of the fourth day (Intervention 4) made three subjects iterate immediately and one a bit later. One subject jumped forward to the last step for which this information was also expected to have an impact. During the presentations two subjects explained explicitly how the tasks of instructors had been minimised in their training program design. A third 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, however, did not have time to do this anymore. Finally, five subjects iterated after the chat sessions. One group was finished with their discussion after half an hour. The other group had to be interrupted when the maximum time (one hour) had passed.

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 or facilities to structure the discussion visually. Pena-Shaff et al. (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 program 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.

Conclusions

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 they can make a reasonably good draft of a simulator training program without major problems when they work seriously on the design task with the MASTER-TPD method and the accompanying support. The usability of the MASTER-TPD method is confirmed by the fact that the subjects did not think that working with it was difficult. Moreover, it proved possible to execute the planned procedure without major changes.

Is the amount of iteration related to the quality of the resulting designs?

The results of this study confirm that iteration is inherent to the design process: all subjects iterate. Some subjects iterate more than others (see Table 4), but the number of iterations does not correlate with the quality of the results (see Figure 3). One explanation could be that the average quality ranking is not a very trustworthy measure for quality (see Table 3). However, the concordance between raters is high[6] and in an earlier study we also did not find relationship between the number of iterations and the quality of the results (Verstegen, 2003; Verstegen et al., 2003). Our hypothesis is 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. 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 above).

It is not only the amount of iteration that matters but also the kind of iteration and the timing: should each error be corrected immediately or can new information be saved up to a later time? To what level of detail should activities be executed at different moments in the design process? In order to study the role of iteration in instructional design and to specify support measures we need a much more fine-grained insight in when iteration is desirable and when it is not. The reasons for iteration defined by van Wagenberg (1992) are still not concrete enough to predict and recognise circumstances that can or should cause iteration. In the next section we will, therefore, propose a new theoretical framework for studying iteration in the form of a list of causes or triggers for iteration.

What are the subjects' opinions about the interventions planned to evoke iteration?

Some subjects were frustrated because not all information was immediately available and tried to avoid iteration. It is, however, typical for the 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.

Do subjects iterate when expected, i.e., after the events introduced to evoke iteration?

The explicit scheduling of design cycles was most successful in stimulating subjects to iterate. This was a strong intervention, although subjects were not forced to go through a global design phase first, it was encouraged by explanations in e-mails, in the Manual and in the process-oriented guidelines. This is not unrealistic. In practice, organisational procedures also oblige designers to make several versions of their specifications and send them to superiors and/or stakeholders for approval. 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 last two interventions (new information about resources and discussion in chat sessions) were less successful in provoking iteration. Maybe the timing of these interventions played a role: they took place later in the week (see Table 2) and subjects may not have had enough time to adapt their designs 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 in the peer review were directly related to specific design activities. 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 programs.

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. We have seen evidence of this in the log files and in the subjects' notes. Moreover, the iterations reported in Table 4 are not necessarily caused by the interventions, they can be coincidental. This is most likely for Intervention 2, the peer review. Most of the comments concerned the work done on the first three steps of the MASTER-TPD method. At the time of this intervention, the second day after lunch, subjects were likely to be working on those steps: they had just gone back to the beginning at the start of the detailed design phase that morning.

Limitations of this study

The study reported in this article has some limitations: a limited amount of subjects and some restrictions on the data that were gathered. Furthermore, 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. 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. The subjects are in some ways similar to the people who are usually responsible for designing simulator training programs: they have no experience in the specification of training simulators or the use of the MASTER method and tool. In other ways they are different: they probably have more theoretical background knowledge about instructional design and development models than most of the target users, but little or no experience in applying this knowledge. On the other hand, they have less domain knowledge and knowledge about training simulators.

Further research

In order to study the role of iteration in instructional design and to specify support measures we need a more fine-grained insight in the circumstances that can or should evoke iteration. For this purpose, we propose a new theoretical framework in the form of a list of causes or triggers for iteration. We believe that these triggers also play a role during the specification of other instructional products and instructional design in general.

Triggers for iteration

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. A third trigger from outside can be the 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 specifications, as shown in Figure 4.

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 when these are not elaborated in sufficient detail and/or information about the difficulty and criticality of the tasks is missing. This trigger is easy to recognise. It is possible to describe the input required for each design activity and to provide guidelines that direct designers back if vital input is missing.

Trigger 2: The need to repair errors

Another cause for iteration can be that designers discover that they have made an error earlier on. Often this trigger is related to others: errors can be discovered when designers evaluate the input for the next design activity (Trigger 1) or they can be identified by other people (Trigger 6 or Trigger 7). Dealing with errors can be supported in

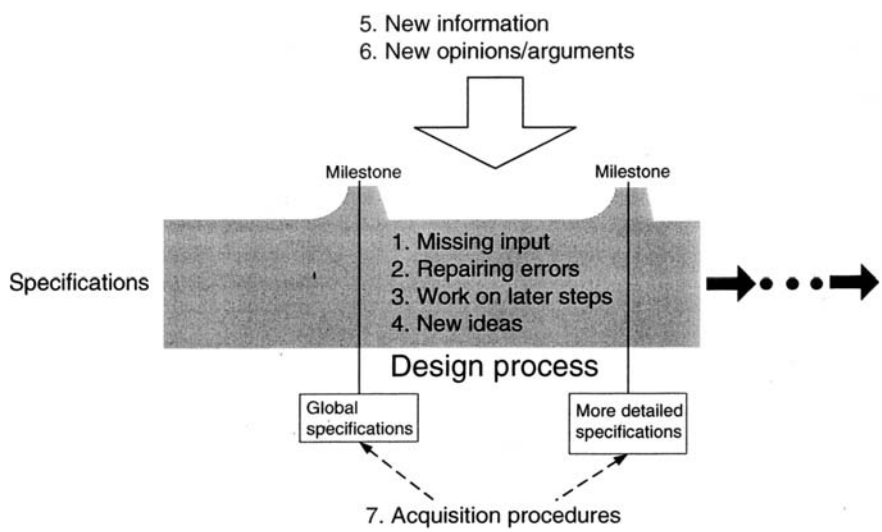

Figure 4. Seven triggers for iteration.

similar ways, e.g., by describing demands on 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 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 exercises, they might discover that training goals that they had allocated to simulator training could be better achieved in another way (e.g. on the job or in the classroom). 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 likely to occur. Their input can be used to develop support, e.g. in descriptions of design activities, guidelines and warnings.

Trigger 4: New ideas of the designer(s)
Designers can get new ideas at any time, triggered by many events. Generating and maintaining ideas makes the design process difficult to manage but will cause more iteration. Presumably, however, it will lead to a better solution. There are design activities where considering alternatives is especially useful, for example the definition of innovative training activities. Design styles and experience will play a role as well. Experienced designers always generate a number of alternative ideas, and do not commit to one particular solution immediately. Experienced designers will have a better overview of the design process, and can plan their design process better. In consequence, they will be better able to recognise triggers for iteration and plan moments to contemplate and generate new ideas.

Trigger 5: New information
New information can trigger iteration, such as new information about the design problem, e.g., about the training goals, the background of trainees, the available resources or other constraints. The problem is that designers might not know what information is available and that their colleagues will not always know which information is relevant for the instructional design process. Guidelines can indicate which kind of information can be relevant for

specific design activities and can help designers to decide whether information is relevant and, if so, what they should do with it. Guidelines can also encourage designers to actively look for additional information and to check whether the information they have is still up-to-date.

Trigger 6: New opinions/arguments from other people

Other parties not directly involved in the design team can bring forward arguments that the designers were previously unaware of. Even when the designers do not agree with these arguments, they will have to spend some time to defend their own decisions. 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. 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. 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. Because it is defined beforehand, this kind of iteration is easy to support with guidelines, warnings and explicit descriptions of the design cycles.

Trials with users

Our model describes the probable causes for iteration within the design phase. In the context of the entire instructional development process there is one other, very obvious trigger for iteration: trials of prototypes with users. When a rapid prototyping approach is used, this trigger for iteration will also occur early in the process.

Last remarks

In practice, not reacting to important triggers for iteration will almost certainly lead to spending time and effort on a nonoptimal 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. Furthermore, it will take time and, thus, cost money. 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 & Frick, 1999). A first attempt in that direction are the case studies presented in Verstegen (2003, Section 9.3.2). Another option is to combine empirical studies similar to the one reported in this article 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, difficult to organise and control, and that it will be hard to isolate critical variables and compare the results between studies.

Notes

1. Different terms are used for these models in the literature and they are used inconsistently. We use Reigeluth's (1983, p. 24) 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.
2. Comparing the quality of intermediate and final versions of the same subject (i.e. within subject design) was considered. We did not execute this analysis because it does not make sense to compare the quality of an unfinished or very global design with the quality of a finished and more detailed design. We know, from an earlier study, that the amount of time spent on the design task has a large influence on the quality of the design (Verstegen, 2003; Verstegen et al., 2003). In this study, we wanted to rule out this effect. By comparing the final designs (between subjects), we know that all subjects had spent the same amount of time on the design task.
3. Friedman ANOVA: Chi Sqr. (n = 3; df = 7) = 14.1; $\alpha < 0.05$.

4. 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 4 may be an underestimate. For some steps the log files only show when objects are added or removed from corresponding tables in the database (for details see Verstegen, 2003). If subjects have gone back to one of these steps without making any changes, it does not show in the log files.
5. Spearman Rho Correlation Coefficient 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.
6. There are also no relations between the ratings of individual raters and the number of iterations.

Acknowledgements

The research reported in this article was carried out at TNO Human Factors, Soesterberg, the Netherlands. We would also like to thank the reviewers for their constructive comments.

References

van den Akker, J. (1999). Principles and methods of development research. In van den J. Akker, R. Branch, K.L. Gustafson, N. Nieveen & T. Plomp, eds, *Design approaches and tools in education and training*, pp. 1–14. Kluwer Academic: Dordrecht, the Netherlands.

Alessi, S.M. & Trollip, S.R. (2001). *Multimedia for learning: Methods and development*. Needham Heights, MA: Allyn & Bacon.

Andrews, D.H. & Goodson, L.A. (1980). A comparative analysis of models of instructional design. *Journal of Instructional Development* 3: 2–16.

van Berlo, M. (2002). *Experimental validation of instructional design guidelines supporting the analysis of team tasks*. Soesterberg, the Netherlands: TNO Human Factors Report No. TM-02-B004.

Blessing, L.T.M. (1994). A Process-Based Approach to Computer-Supported Engineering Design. Doctoral dissertation, University of Twente, Enschede, the Netherlands.

van den Bosch, K., Barnard, Y.F. & Helsdingen, A.S. (1999). *Taak- en trainingsanalyse beeldanalist SPERWER (Task and Training Analysis Image Interpreter SPERWER)*. Soesterberg, the Netherlands (in Dutch): TNO Human Factors Report No. TM-99-A018.

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.

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).

Dijkstra, S. (2001). Epistemology, psychology of learning and instructional design (Special issue). *Instructional Science* 29(4–5).

Farmer, E.W., Jorna, P.G.A.M., Riemersma, J.B.J., van Rooij, J.C.G.M. & Moraal, J. (1999). *Handbook of simulator-based training*. London: Ashgate.

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.

Greeno, J.G., Korpi, M.K., Jackson III, D.N. & Michalchik, V.S. (1990). *Processes and knowledge in designing instruction*. Stanford, CA: Stanford University Report No. N00014-88-K-0152.

Gustafson, K.L. (1991). *Survey of instructional development models*. (2nd edn.) 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. Educational Technology Publications: Englewood Cliffs, NJ.

Hamel, R. (1990). Over het denken van de architect: Een cognitief psychologische beschrijving van het ontwerpproces bij architecten (On Designing by Architects: A Cognitive Psychological Description of the Architectural Design Process). Doctoral dissertation, University of Amsterdam, the Netherlands (in Dutch).

Holcomb, C., Wedman, J.F. & Tessmer, M. (1996). ID activities and project success: Perceptions of practitioners. *Performance Improvement Quarterly* 9(1): 49–61.

Hoogveld, A.W.M., Paas, F., Jochems, W.M.G. & van Merriënboer, J.J.G. (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.

van der Hulst, A.H., de Hoog, R. & Wielemaker, J. (1999). *BOOT: Decision support for the selection of facilities for education and training*. The Hague the Netherlands: TNO Physics and Electronic Laboratory Report No. FEL-99-A188.

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).

Kearsley, G. (2003). *Explorations in learning and instruction: The theory into practice (TIP) database*. Retrieved September 20, 2003, from http://tip.psychology.org.

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.

Kincaid, B. (1997). *A Dinosaur in Whitehall: The true cost of defence procurement bureaucracy*. London: Brassey's.

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.

van Merriënboer, J.J.G. (1997). *Training complex cognitive skills*. Englewood Cliffs, NJ: Educational Technology Publications.

Merrill, M.D., The ID2 Research Group (1996). Instructional transaction theory: Instructional design based on knowledge objects. *Educational Technology* 36(3): 30–37.

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.

Nelson, W.A., Magliaro, S. & Sherman, T.M. (1988). The intellectual content of instructional design. *Journal of Instructional Development* 11(1): 29–35.

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).

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. Plenum Press: New York.

Pieters, J.M. & Bergman, R. (1995). The empirical basis of designing instruction. *Performance Improvement Quarterly* 8(3): 118–129.

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. (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.

Riemersma, J.B.J., van Rooij, J.C.G.M., Just, J., Farmer, E., Paris, Ph., Fuchs, M., Reinschlüssel, R., Jorna, P.G.A.M. & Bermejo Muñoz, J. (1994). *Reference Framework*. Soesterberg, the Netherlands: TNO Human FactorsEUCLID RTP 11.1 MASTER Deliverable Dl.1.

van Rooij, J.C.G.M. (1997). Onderwijskundige richtlijnen ten behoeve van de specificatie van simulatoreisen (Didactic guidelines for the specification of simulator requirements). Report No. TM-97-A077, TNO Human Factors, Soesterberg, the Netherlands (in Dutch).

van Rooij, J.C.G.M. (2002 May). Digitalisatie van het trainingsveld (Digitalisation in the field of training). *Opleiding and Training* 2002(17): 4–8 (in Dutch).

Rowland, G. (1991). Problem Solving in Instructional Design. Doctoral dissertation, Indiana University, Indiana.

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 van den J. Akker, R. Branch, K.L. Gustafson, N. Nieveen & T. Plomp, eds, *Design approaches and tools in education and training*, pp. 29–44. Kluwer Academic: Dordrecht, the Netherlands.

Ryder, M. (2003). *Instructional design models* (d.d. 16–9-2003). Retrieved September 20, 2003, from http://carbon.cudenver.edu/~mryder/itc/idmodels.html.

Saroyan, A. (1993). Differences in expert practice: A case from formative evaluation. *Instructional Science* 21: 451–472.

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. Springer-Verlag: Berlin Germany.

Tripp, S.D. & Bichelmeyer, B. (1990). Rapid prototyping: An instructional design strategy. *Educational Technology Research & Development* 38(1): 31–44.

Verstegen, D.M.L. (2003). Iteration in Instructional Design: An Empirical Study on the Specification of Training Simulators. Doctoral dissertation, Utrecht University, the Netherlands. Can be retrieved from: http://www.library.uu.nl/digiarchief/dip/diss/2004-0130-092005/inhoud.htm.

Verstegen, D.M.L., Barnard, Y.F. & Pilot, A. (2002). *Do training simulators support instruction and practice? Results of field visits to 44 simulators*. Soesterberg, the Netherlands: TNO Human Factors Report No. TM-02-B005.

Verstegen, D.M.L., Barnard, Y.F. & Pilot, A. (2003). *Iterative design of training programs: Results of an evaluation study*. Soesterberg, the Netherlands: TNO Human Factors Report No. TM-03-B005.

Verstegen, D.M.L., Barnard, Y.F. & Pilot, A. (2004). *Second evaluation of iterative design of training programs*. Soesterberg, the Netherlands: TNO Human Factors Report No. TM-04-B006.

Verstegen, D.M.L. & van der Hulst, A.H. (2000). Standardized development of a needs statement for advanced training means. In *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*. Soesterberg, the Netherlands: TNO Human Factors Report No. TM-00-B009.

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*. Soesterberg, the Netherlands: TNO Human Factors IMAT Deliverable R.I.3.

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 van den J. Akker, R. Branch, K.L. Gustafson, N. Nieveen & T. Plomp, eds, *Design approaches and tools in education and training*, pp. 15–28. Kluwer Academic: Dordrecht, the Netherlands.

van Wagenberg, M.J.G.M. (1992). Gericht CAD-ondersteund ontwerpen en organiseren (Goaloriented CAD-Supported Designing and Organising). Doctoral dissertation, Delft University of Technology, the Netherlands (in Dutch).

Wedman, J.F. & Tessmer, M. (1993). Instructional designers' decisions and priorities: A survey of design practice. *Performance Improvement Quarterly* 6(2): 43–57.