

## Situational Requirements Engineering of Web Content Management Implementations

Inge van de Weerd<sup>a</sup>, Jurriaan Souer<sup>a</sup>, Johan Versendaal<sup>b</sup> and Sjaak Brinkkemper<sup>b</sup>

<sup>a</sup> <GX> Wijchenseweg 111, 6538 SW, Nijmegen, The Netherlands

<sup>b</sup> Department of Information and Computing Sciences, University of Utrecht, The Netherlands

### Abstract

The development of complex, data-intensive web applications is becoming simpler due to the usage of content management systems. Conventional information systems development methods as well as web application development methods do not cover the needs of a method for web content management implementations. In this research we apply an assembly-based situational method engineering approach to develop a new design method, called WebEngineering Method (WEM). First, implementation situations are identified, then, candidate methods are selected and finally, a new method of useful method fragments is assembled. By using route map configuration, we tune the method fragments to obtain situationality. A meta-modeling technique is proposed that integrates UML activity diagrams and class diagrams for the purpose of analyzing existing methods and assembling the new method. The method developed was validated in a case study, which consisted of a technology testing web application at a large telecommunication organization in the Netherlands. The case study results were very promising, but more case studies are required to refine the method.

**Keywords:** Project situation assessment for RE process, RE for web applications, Method engineering techniques for situational RE methods construction, Experience and case studies of situation-specific RE, Content management

### 1 Introduction

A large number of information system development methods are available. Next to established methods like entity-relationship modeling (Chen, 1976) and the more recent Unified Process (Jacobson, Booch & Rumbaugh, 1999), several methods and techniques for developing *Web* applications have been developed. Examples are WebML (Ceri, Fraternali and Bongio, 2000), UWE (Koch, 2001) and W2000 (Baresi, Garzotto, and Paolini, 2001). These methods show significant influences from information systems development methods. This is not surprising since web applications can be seen as a subtype of information systems (Souer, et al. 2005). Gnaho (2001) acknowledges this in his definition of Web applications: a Web application is an Information System providing facilities to access complex data and interactive services via the Web which changes the state of business.

Using data-intensive Web applications raised new problems concerning consistency, navigation, data duplication, content audit and control, tracking of content and mapping the website workflows on the business processes (Vidgen, Goodwin and Barnes, 2001). The solution to these problems was found in content management. A content management system (CMS) makes it possible to create, archive, search, control and publish information from within a flexible and integrated environment (Burzagli, et al., 2004). A special type of content management systems

are CMS-based Web applications, which are defined as Web applications for the management and control of information (Souer, et al., 2005). Examples are GX WebManager, FatWire Content Server and Tridion Web Content Management Edition.

In this paper we concentrate on the implementation of CMS-based Web applications. To our knowledge, in academic and professional literature no specific methods on this subject exist. Currently, established information system and Web application development methods are being used, but these methods are not able to cover specific content management aspects.

### 1.1 Method Engineering

Kumar and Welke (1992) state that “there is no detailed information systems methodology which is the best in all situations”. They introduced a solution to this problem, method(ology) engineering, which describes the engineering of information systems development methods, by taking into account the uniqueness of a project situation. In addition, Brinkkemper (1996) defined method engineering as “the engineering discipline to design, construct and adapt methods, techniques and tools for the development of information systems”.

Van Slooten and Brinkkemper (1993) introduced route maps, where method fragments are combined to form new methods. Route maps can be used to tune the method into situational methods (Van Slooten and Hodes, 1996; Aydin and Harmsen, 2002).

Rossi et al. (2000) mention the development of UML extensions as a reaction to the abundance of variants of UML for special purposes as a prime example of successful situational method engineering. Also Dietsch (2002) showed that situational method engineering can be used as an appropriate approach for solving the problem to finding the right method.

Recent research in the method engineering area is done by Ralyté, Deneckère and Rolland (2003). They developed a generic process model for situational method engineering. This process model contains three approaches: (a) the assembly-based strategy, based on the reuse of method components extracted from existing methods and stored in some method base; (b) the extension-based strategy, used for extending a method by applying extension patterns; and (c) the paradigm-based strategy, when a new fresh method must be constructed either by abstracting from a given model or by instantiating a meta-model (Ralyté, Deneckère and Rolland, 2003).

Karlsson and Ågerfalk (2004) use method configuration to adapt a particular method to various situated factors. The difference with assembly-based method engineering is that the focus is on one method which is configured in a particular situation, instead of using a set of methods as a base for assembly.

### 1.2 Existing Web Application Development Methods

Since no methods for the development of implementations of CMS-based Web applications exist, we studied the field of web application methods.

The first method we describe is the Unified Software Development Process (Unified Process), which is “a generic process framework that can be specialized for a very large class of software systems, for different application areas, different types of organizations, different competence levels, and different project sizes” (Jacobson, Booch & Rumbaugh, 1999). The Unified Process is actually an information systems

development method, instead of a Web application design method. Nevertheless, we mention it since a) its commercial instance, the Rational Unified Process, has emerged as a de facto standard software development process (Larman, Kruchten and Bittner, 2001); and b) it has had a significant influence on later developed web application design methods. Distinguishing features are captured in the following key words: use-case driven; architecture centric; and iterative and incremental. The lifecycle consists of 4 phases, namely inception, elaboration, construction and transition. In the four phases five core workflows are addressed, which are: requirements, analysis, design, implantation and test.

Since 1998, several methods and techniques specifically for designing Web applications have been developed. First of all, in 1998 the Website Design Method was developed by De Troyer and Leune. WSDM is a user-centered method for the design of kiosk Web Sites. A kiosk Web site mainly provides information and allows users to navigate through that information (De Troyer and Leune, 1998). The two basic characteristics of WSDM are the audience driven approach, and the explicit conceptual design phase. The conceptual design can be performed by using techniques like OMT or E-R modeling.

Subsequently, Sauer and Engels (1999) proposed the UML Extension for Modeling Multimedia Applications. They define a multimedia application as an application that combines at least two media objects and shows time-dynamic behavior. Aspects of the application which are covered in this extension are: (a) logical structure; (b) spatial presentation; (c) predefined temporal behavior; and (d) interactive control. Another extension was developed by Baumeister, Koch & Mandel (1999). They propose this UML Extension for Hypermedia Design, because the diagrams of UML are not sufficient to model aspects as navigational space and graphical representation.

Koch (2001) proposed the UML-based Web Engineering (UWE) approach. This approach is object-oriented, visualized with UML and based on the Unified Process. UWE is a systematic, prescriptive, user-centric, UML-based, iterative and incremental method for adaptive hypermedia systems (Koch, 2001). Brusilovsky, as cited in Koch (2001), provides the following definition of adaptive hypermedia systems: “hypermedia systems which reflect some features of the user in a user model and use this model by adapting various visible aspects of the system to the user”. UWE covers the full development process, divided in requirements analysis, conceptual, navigation and presentation design (Koch, 2001).

WebML is a notation for specifying complex web sites at the conceptual level (Ceri, Fraternali and Bongio, 2000). Its specification consists of four perspectives: (a) the structural model; (b) the hypertext model; (c) the presentation model; and (d) the personal model. It is not based on UML, but it is compliant with it. It does not propose a new language for data modeling, but is compatible with existing notations such as E-R modeling and UML. Also, WebML supports XML syntax, which can be used by software generators.

Finally, W2000 is a framework for designing web applications based on the preexisting assets UML and HDM (Baresi, Garzotto, and Paolini, 2001). According to the authors, the integration between UML and HDM consists in (1) defining several stereotypes and customizations of diagrams to render HDM with UML; (2) specifying guidelines to use UML as a way to specify some of the dynamic and operational aspects of web applications; and (3) refining use case diagrams to describe high-level

user requirements, related to both functional and navigational aspects (Baresi, Garzotto, and Paolini, 2001).

### 1.3 Research Question

Over the years Web applications have evolved in, and make use of, CMS-based Web applications. Since the existing web application development methods do not cover content management aspects, there is a need for methods for developing CMS-based Web applications. This led us to our research question: *“How should a design method be constructed for the process of implementation of CMS-based Web applications?”*

We use assembly-based situational method engineering to develop the new design method. The advantage of this method is that we can reuse relevant, established method fragments of existing methods. In this way, an optimized method for every implementation situation is being developed. A modeling technique that integrates activity diagrams and class diagrams is developed for the purpose of analyzing existing methods and assembling new methods.

The primary measure of success of a software system is the degree to which it meets the purpose for which it was intended (Nuseibeh and Easterbrook, 2000). Hence, requirements analysis is an important part of the development process. Also in implementing CMS-based Web applications, requirements analysis is important. Lowe and Henderson-Sellers (2001), for example, highlighted the need for a design-driven requirements process that structures the way in which design activities for Web systems can be linked to the clarification of requirements through an appropriate model of domain uncertainty. Also, Escalona and Koch (2004) state that “Web applications require a more extensive and detailed requirements engineering process due to the number of stakeholders involved and due to the diversity of the requirements including among others requirements on the navigation and on the business processes as well as Web usability”. Therefore, the focus in this article is on the first phases of the development process where requirements are identified, described and validated.

The paper is organized as follows. In section two the process of method engineering is described. Then, in section three, we propose our situational method for CMS-based Web applications. In section four, the validation is described and finally, the conclusions are covered in section five.

## 2 Assembly-Based Situational Method Engineering

### 2.1 Approach

The approach to situational method engineering described in most literature is quite clear. Brinkkemper (1996) recognized the following steps: (1) characterization of the project, (2) selection of method fragments (that are stored in a method base), and (3) assembly of method fragments. The experience gained in this process is new input for the method base. Saeki (2001) states that the simplest way to construct a new method is first to put meaningful method fragments in a method base, then to select useful method fragments from this method base, and finally adapt and integrate them in a new method. Ralyté, Deneckère and Rolland (2003) have developed the assembly-based process model for situational method engineering. This model describes three steps to develop a new situational method. The steps are: (1) specify method requirements, (2) select method fragments and (3) assemble method fragments.

In the described research it is either assumed that the method base with method fragments is already filled, or that the methods that are to be stored in the method base are already selected. In case of developing methods for a relatively new information systems field, in this case CMS-based Web applications, the method base needs to be filled first. Therefore, we propose the following steps to develop a new situational method for implementing CMS-based Web applications:

1. Analyze implementation situations and identify needs.
2. Select candidate methods that meet one or more aspects of the identified needs.
3. Analyze candidate methods and store relevant method fragments in a method base.
4. Assemble a new method from useful method fragments and use route map configuration to obtain situational methods.

This process is supported by a meta-modeling technique, especially developed for method engineering purposes. This technique, in which a process-data diagram is built, is used in analyzing, storing, selecting and assembling the method fragments.

First, we will outline the meta-modeling technique that is used to support the selection and assembly process. Then, in the next section, the steps of the process model are exemplified by bringing them into practice.

## 2.2 Meta-Modeling Technique

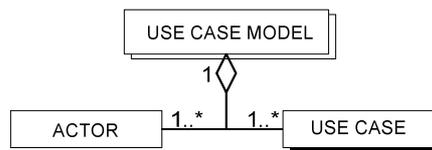
The technique developed to model the activities and artifacts in the development process is a meta-modeling technique, expressed in process-data diagrams. Saeki (2003) proposed the use of a meta-modeling technique for the purpose of attaching semantic information to the artifacts and for measuring their quality using this information. In this research the meta-modeling technique is adopted and adjusted to reveal the relations between activities (the process) and data (the deliverables that are produced in the process) of the development method. This makes it possible to fragmentize methods and to configure both the process and data part of a (situational) method.

A process-data diagram consists of two integrated meta-models. The left-hand side is a meta-process model based on a UML activity diagram (OMG, 2003), and the right-hand side is a meta-data model based on a UML class diagram (OMG, 2003). In Figure 1 the integration of both meta-models, which results in a process-data diagram, is shown. The dotted arrows indicate which concepts are created or adjusted in the corresponding activities.



concepts. However, it is possible that an actor can possess several attributes. Another example of a standard concept is a term in a glossary.

- *Open concept*, which is an expanded complex concept that consist of a collection of (sub) concepts. An open concept is visualized with an open shadow. In Figure 2, *use case model* is represented as an open concept. The use case model is an open concept and consists of one or more actors and one or more use cases. Other examples of open concepts are a domain model, consisting of several terms(or concepts) and relations.
- *Closed concept*, which is an unexpanded complex concept that consist of a collection of (sub) concepts. A closed concept is visualized with a closed shadow. *Use case* in Figure 2 is an example of a closed concept. Note that the context of which it is modeled is important. A use case is detailed in a use case description, which has a flow of events, conditions, special requirements, etc. Also, a use case can have a priority. Because in this case we decided it is unnecessary to reveal that information, the use case is illustrated with a closed concept. Another example of a closed concept is a class diagram. For this concept, the same holds as for the use case diagram.



**Figure 2:** Example of standard, open and closed concepts

### 3 GX WebEngineering Method

#### 3.1 Introduction

The research described in this work is carried out at GX creative online development, a web technology company in the Netherlands. GX is active in the fields of content management, online application development, and integration of backend systems in portal solutions. The company implements web applications, using GX WebManager, a generic CMS-based web application tool. GX WebManager enables people without a specific technological background in creating, maintaining and integrating several dynamic websites and portals. In addition to their product, the company also provides a service, which is creating a web application ‘around’ the CMS-based Web application. The development of this web application is currently carried out by a proprietary method. However, the need exists to optimize this method in order to save time and money. Also, the need for a standardized web application development method exists, which can be used by implementation partners of the company. Therefore we developed the the GX WebEngineering Method (WEM).

In the next sections, the development of WEM is outlined, following the four steps as described in section 2.1.

### 3.2 Implementation situation analysis

Brinkkemper (1996) as well as Kumar and Welke (1992) stress the importance of distinguishing development situations. In this research, we use the term *implementation situations*, since the project deals with the implementation of a CMS-based Web application. Consequently, the first step in the method engineering process is to analyze the projects, categorize them in implementation situations, and identify their specific requirements. The categorization of implementation situations is based on their distinguishing characteristics.

Karlsson (2002) followed a similar process as Brinkkemper (1996) and Kumar and Welke (1992) in abstracting projects into development situations, but for the purpose of method configuration. He defined a characteristic of a development situation as: “a delimited part of a development situation, focusing on a certain problem or aspect which the method configuration aims to solve or handle”. We use this definition to define a characteristic of an implementation situation, that is: “a characteristic is a delimited part of an implementation situation, focusing on a certain problem or aspect which the method aims to solve or handle”.

In 2004 GX completed about 80 implementations. These projects vary in size, sector and type. The number of employees of the customer organizations ranges from a few to tens of thousands of employees. These implementations have been executed in a range of industries, like services, sports organizations, publishing companies, media, government, education, knowledge centers, and health care. Resulting from artifact research and semi-structured interviews, we concluded that three kinds of implementation situations exist at GX: standard, complex and migration projects. The latter one was easy to identify. When a large amount of content from an existing Web application needs to be migrated to the new Web application, this is classified as a migration development situation. However, the difference between standard and complex implementation situations is more ambiguous.

Kumar and Welke (1992) and Van Slooten en Hodes (1996) mention several characteristics for the categorization of development projects that are of importance. In general one can state that these factors are deduced from the context, organization or from technical aspects from the project (Karlsson, 2002). In Table 1 the adopted characteristics per area are described.

**Table 1:** Implementation situation characteristics

Area	Characteristic
Context	Dependency (to external activities & conditions)
	Level of innovation (of the applied technology, methods, tools and techniques)
Organization	Number of stakeholders
	Uncertainty of customer's expectations by management team
	Uncertainty of development activities by customer
Technique	Complexity (of functional components)
	Number of relationships (to existing systems)

The characteristics listed in Table 1 can be used to categorize the development situations. Every characteristic can be labeled with a value: high or low. In general we state that the complexity of an implementation situation depends on the amount of characteristics that is labeled with a high value. When more than three of the values are high, it should be categorized as a complex development situation. Otherwise, it is a standard development situation.

In this work we focus on standard and complex development situations. Requirements were obtained by conducting semi-structured interviews with consultants, project managers and software architects; and artifact analysis of existing requirements specifications and project evaluation documents. Several problems were found and translated into overall needs, standard implementation situation needs and complex implementation situation needs. In Table 2 example needs are given for each implementation situation.

**Table 2:** Implementation situation needs

Situation	Need
Standard & Complex	The method should deliver a requirements document that is understandable for the customer and informative for the stakeholders at GX.
Standard	Standard project often have a small budget. This implies that the amount of time for specifying the requirements is limited. Therefore, the method should make it possible to translate the requirements quickly into WebManager solutions.
Complex	A solution has to be found to the problem of changing requirements after the contract is signed. Although one can expect the requirements to change during the requirements analysis, the customer often does not understand that this affects the budget.

### 3.3 Candidate method selection

The following step is the selection of candidate methods from which method fragments are extracted and stored in a method base. After conducting a literature research, we chose to make use of Unified Process (Jacobson, Booch & Rumbaugh, 1999) and UML-based Web Engineering (Koch, 2001).

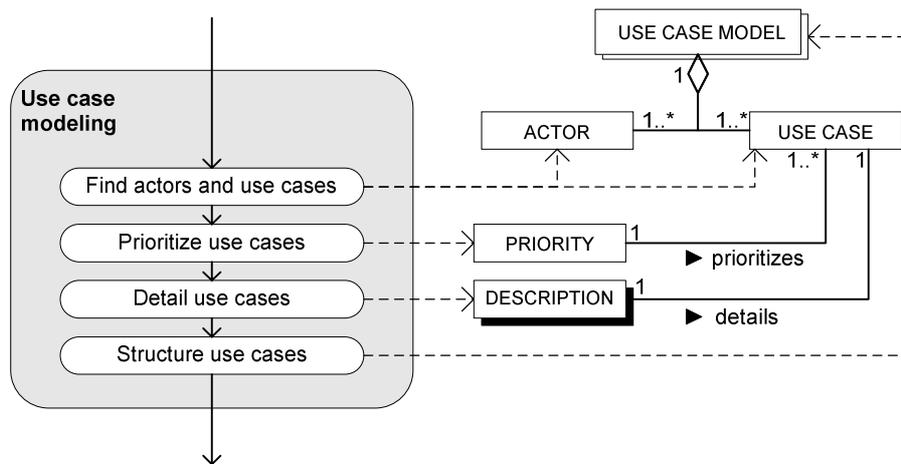
In choosing the candidate methods, the following considerations were taken into account: (a) the Unified Process is very suitable to divide into fragments and store in a method base; (b) UWE combines the strengths of the Unified Process with several Web-specific characteristics; (c) consultants who are going to use the method are already familiar with RUP, the commercial variant of the Unified Process; and (d) both methods use UML as modeling language, which is the standard notation for modeling object-oriented systems and widely accepted by the software engineering community (Baresi, Garzotto and Paolini (2000).

### 3.4 Candidate methods analysis

All three methods, GX development process, Unified Process and UML-based Web Engineering, are analyzed by expressing the process in a process-data diagram. From

every method main and sub activities are identified. Each of these activities results in a deliverable, which is represented at the data-side of the diagram.

In Figure 3 is illustrated how a method fragment is visualized. It represents a fragment of the requirements workflow in UML-based Web Engineering. On the process-side, one activity (use case modeling) and four sub-activities can be found. The first sub-activity, find actors and use cases, results in the concepts actor and use case. Next, a priority is given; the use case is detailed by providing it with a description; and, finally, the use cases are structured, which results in the use case model. The use case description is not expanded in this fragment, since the sub-concepts of this complex concept were of no importance in this context.



**Figure 3:** Implementation situation characteristics

All process-data diagrams that are produced form the method base, from which fragments are selected to assemble the new method. The complete method base can be found in Van de Weerd, 2005.

### 3.5 New Method Assembly

Based on the defined implementation situations and requirements, we chose which method fragments to use in the new method. In order to assemble the right method fragments to the implementation situations, the needs of every situation were examined, after which the method fragments were mapped with the implementation situations. We chose to represent the results of the method engineering process as routemaps. The advantage of using routemaps, compared to using different methods per situation, is that it preserves the method from inconsistencies and that updating of the method is easier. The routemaps are static, that is, when an implementation situation is chosen, the route that is followed is definite. However, if usage of WEM indicates that parts of the method are not useful, the routemap can be adapted. Also, when a new implementation situation is identified, a route can be added to the method.

The complete method is divided in six project phases: acquisition, orientation, definition, design, realization and implementation. In Figure 4 the process-data diagram of the definition phase of WEM is illustrated.

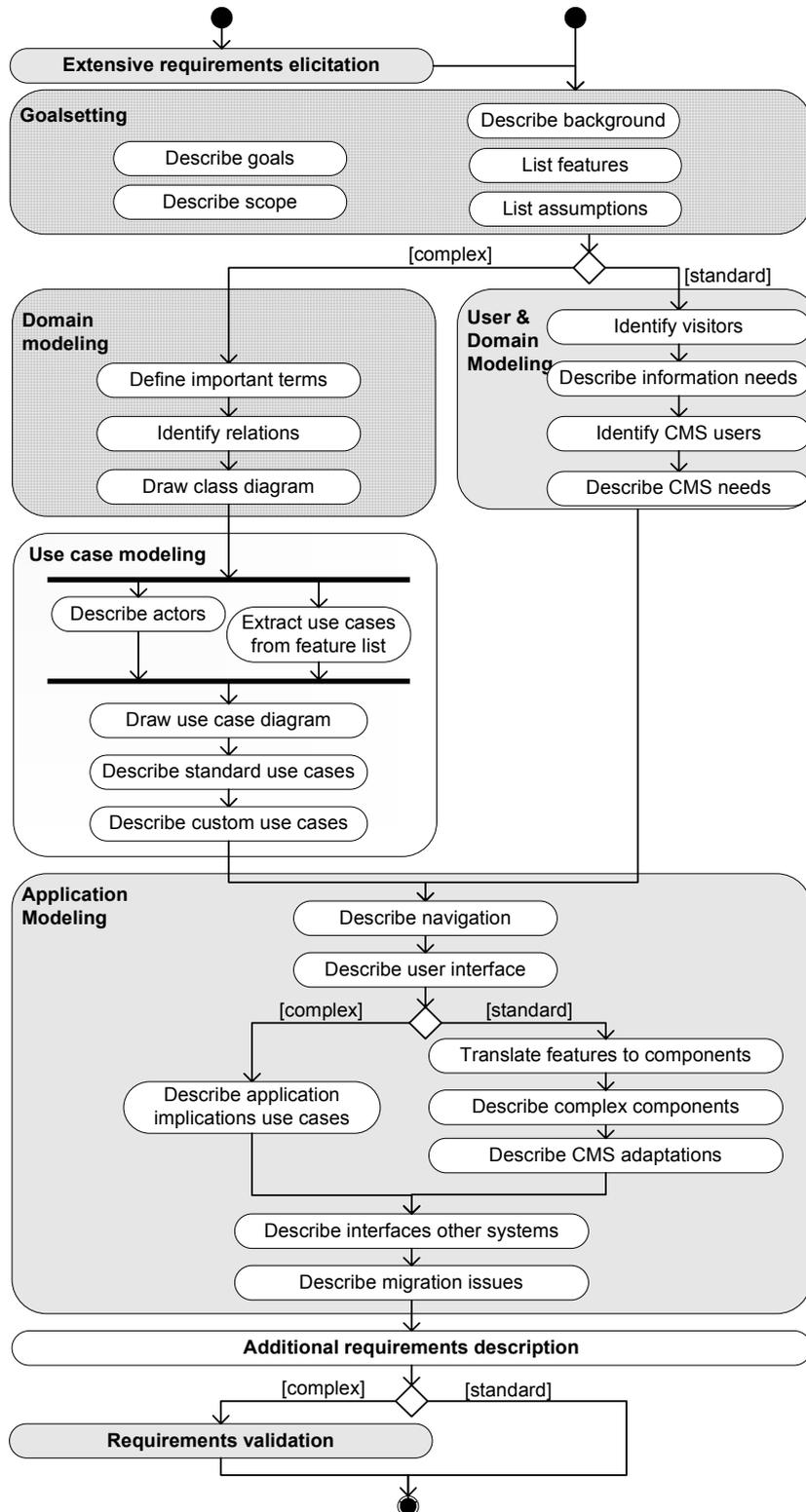


Figure 4: New method - Activities in the definition phase

We preferred to show both standard and complex routes in one diagram, to make clear what the differences are between the two implementation situations. Therefore, we omitted the data-side of the diagram. The main activities in the diagram are marked to indicate from which method they origin. A checked pattern indicates that this method fragment originates from the old method at GX; grey indicates that it is a UWE fragment; and, finally, white indicates a Unified Process origin.

The main difference between the standard and complex route is, next to the extensive requirements elicitation and validation, the use of use case modeling. In the complex route this is used to describe the people who will interact with the Web application, as well as how they will interact with the system. In the standard route this is partly handled in the user and domain modeling fragment and partly in the application modeling. In the appendix, the process-data diagrams of the definition phase of both the standard (Figure 5) and complex (Figure 6) routes can be found.

## 4 Validation

We applied two types of validation: expert validation and a case study. Because of scoping issues, only the definition phase is covered.

### 4.1 Expert Validation

WEM was developed with input of the requirements management workgroup. The goal of this workgroup was an overall improvement in the requirements process at GX. Members of the workgroup were consultants and project managers of GX and one external consultant. WEM was assessed in this workgroup. Two route maps of the method were validated, namely the standard and complex route maps. The results were positive, for the following reasons:

- The distinction of standard and complex implementation situations was perceived as very useful.
- The use of user and domain modeling in the standard route was seen as very practical.
- Use case modeling the complex route was seen as very useful.

### 4.2 Case Study

The route map for complex implementation situations was tested in a case study. In the case study the method was used in a project that consisted of building a Web application for a large telecommunication organization in the Netherlands. The purpose of the Web application was to support the testing of new products and services that are offered to a limited group of customers in a limited period of time. Employees of the organization should be able to develop and test a new offer with the application, without the help of GX. Several connections with existing back office systems had to be realized. Also, online payment of the products and services had to be supported. Finally, a special application for the Customer Care department needed to be developed, in order to support this department with customer service.

The project had an estimated budget of 400 man hours. Several stakeholders were involved in the requirements phase of the project. At the side of GX these were: (a) project manager, (b) consultants, and (c) software architect. Stakeholders of the

customer organization were: (a) business project manager, (b) technical project manager, (c) Web department manager, and (d) Customer Care project manager.

#### *4.2.1 Usage of the Method*

Before the start of the project, a requirements document template was created. Also, a briefing was given to the consultants and project manager to outline the new method. The requirements analysis was carried out by GX consultants and reviewed by the project manager.

The requirements document consisted of thirty-two pages. The use case model consisted of seven actors, who were connected to seventeen use cases. Eight of these use cases were immediately translated to standard GX WebManager components. The others were more complex and were provided with use case descriptions. One part of the method was omitted, namely the drawing of a class diagram to model the domain, since the use was not necessary in this project.

In an interview, the consultants responded positive to the new method. In comparison to the old method, WEM was more structured and better able to describe complex functionalities. Also, the domain modeling was commented on as being clarifying and useful. A remark was made on the use of a feature list, which was not recognized as useful. Another comment was that use case modeling is time-consuming. However, the budgeted hours for this project were not exceeded.

#### *4.2.2 Evaluation of the Requirements Document*

The requirements document was sent to the all stakeholders at GX and the customer. At GX, the requirements document was perceived as 'clear, structured, and with the right level of detail'. All stakeholders agreed that this document was an improvement to former requirements documents. However, the project manager expressed the fear that this method was too time-consuming.

To the project organization project members a survey was sent. The questions were divided in several categories. First, questions on the requirements process were asked in the categories (a) structure, (b) team, and (c) general. Then, questions on the requirements document were asked in the categories (a) understandability, (b) correctness, (c) use case modeling, and (d) general.

The answers to the survey appeared to be overwhelmingly positive. On a Likert scale of 1 to 5, where 1 was most negative and 5 most positive, a mean score of 4.4 was received. The given answer ranged from 3 to 5. No significant difference in scores was measured between the "process part" and "document part".

In summary, the requirements document was 'understandable and logical', with the right level of detail. Also, the functionalities described in the use cases perfectly matched the functionalities they wanted to be realized. Use cases were considered to be a great way to describe functionalities, since they are understandable for technical and non-technical project members.

#### *4.2.3 Discussion*

Summarizing, the results of the case study were positive. Nevertheless, some comments were made. First of all, only the definition phase was covered, which implies that the acquisition phase was done in the 'old-fashioned' way. The most obvious consequence was that the feature list, which should have been created in the

acquisition phase and used in the definition phase, was seen as redundant by the consultants. The customer, however, did not comment on this.

Secondly, all project members at the customer's organization were familiar with use case modeling. If they were not, the requirements document might have been more difficult to understand.

Using the method may lead to new insights. The developed method is not static, but dynamic. Users of the method should adapt it to their own preferences. When it appears that an activity structurally is omitted, the method should be updated.

## 5 Conclusions

In this paper an improvement is proposed to the existing method engineering process. The described process helps in developing a method base, consisting of candidate methods that are selected based on how they meet the identified implementation situation needs. Secondly, a meta-modeling technique, resulting in a process-data diagram, for method engineering is developed. By modeling the relations between activities and concepts, it is possible to engineer both process and data part of the method. In the future, this technique should be updated, to keep it consistent with the UML standards.

Looking at the delivered results of the research, a method has been developed and tested for the process of implementing CMS-based Web applications: the WebEngineering Method. This method can be used for standard and complex implementation situations, by following the described route maps. Since no such method existed, this research is an important addition to the existing information systems and Web development methods.

A limitation on WEM is that only the acquisition, definition, orientation and analysis phases are covered. In addition, only the definition phase has been validated. However, the results were promising. In future work, the method should be expanded and refined, based on experiences in executed projects. Also, besides the Unified Process and UML-based Web engineering, other relevant methods can be analyzed to improve the method base.

Finally, a challenge lies in the integration of the method in GX WebManager. As the content management system itself is capable to store structured documents, it makes sense to integrate the WEM design method as an extension to the WebManager product. This strategy is similar to the extension of the Oracle DBMS with Oracle CASE tools, or of the Baan ERP software with the Dynamic Enterprise Modeling (DEM) tooling (Brinkkemper, 1998).

## Acknowledgements

The authors wish to thank Roel Bouwmans, René Janssen and Bob Kennedy for giving input to WEM during its development.

We thank Nils ten Broeke, Christiaan Hoogerwerf, Michel Teunissen and the rest of the project team who developed the Web application, as described in the validation, for their contributions on the validation of this paper. Also, the project team members of the customer are thanked for their cooperation in testing and validating WEM.

Finally, we would like to thank the anonymous reviewers for their extensive remarks that led to the improvement of this paper.

## References

- AYDIN MN and HARMSSEN, F (2002) Making a Method Work for a Project Situation in the Context of CMM. *Lecture Notes in Computer Science*, 2559, 158-171.
- BARESI L, GARZOTTO F and PAOLINI P (2001) Extending UML for modeling web applications. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, HI, USA.
- BAUMEISTER H, KOCH N and MANDEL L (1999) Towards a UML extension for hypermedia design. In France, R., Rumpe, B. (eds.), *Proceedings «UML» '99, Lecture Notes in Computer Science*, 1723, 614-629.
- BRINKKEMPER S (1996) Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*, 275-280.
- BRINKKEMPER S (1998) Global Process Management. In *Dynamic Enterprise Innovation: Establishing Continuous Improvement in Business* (Es R, Ed), 4-8 - 4-15, Baan Business Innovation, Putten, The Netherlands.
- BURZAGLI L, BILLI M, GABBANINI F, GRAZIANI P and PALCHETTI E (2004) The use of current content management systems for accessibility. *Proceedings of the 9th International Conference on Computers Helping People with Special Needs (ICCHP 2004), Lecture Notes in Computer Science*, 3118, 331-338.
- CERI S, FRATERNALI P and BONGIO A (2000) A web modeling language (WebML): A modeling language for designing web sites. *Computer Networks*, 33, 137-157.
- CHEN PP (1976) The Entity-Relationship Model - Towards a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9-36.
- DE TROYER OMF and LEUNE CJ (1998) WSDM : A user-centered design method for web sites. In *Proceedings of the Seventh International World Wide Web Conference*, Elsevier, 85-94.
- DIETZSCH A (2002) Adapting the UML to business modelling's needs - Experiences in situational method engineering. In <<UML>> 2002 – *Proceedings of the Unified Modeling Language; Fifth International Conference* (JÉZÉQUEL J-M, HUSSMANN H and COOK S, Eds), Dresden Germany, September 2002. Springer, 73-83.
- ESCALONA MJ and KOCH N (2004) Requirements engineering for web applications: A comparative study. *Journal on Web Engineering, Rinton Press*, 2(3), 193-212.
- GNAHO C (2001) Web-Based information systems development – A user centered engineering approach. *Lecture Notes in Computer Science*, 2016, 105-118.
- HARMSSEN F, BRINKKEMPER S and OEI JLH (1994) Situational method engineering for informational system project approaches. *Proceedings of the IFIP WG8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle*, 169-194.
- JACOBSON I, BOOCH G and RUMBAUGH J (1999) *The unified software development process*. Redwood City, CA: Addison Wesley Longman Publishing Co., Inc.
- KARLSSON F (2002) Bridging the gap between method for method configuration and situational method engineering. Promote IT, Skövde, Sweden.

- KARLSSON F and ÅGERFALK PJ (2004) Method configuration: Adapting to situational characteristics while creating reusable assets. *Information and Software Technology*, (46)9, 619-633.
- KOCH N (2001) *Software engineering for adaptive hypermedia applications* (Phd Thesis, Austria, Munich, 2001). Uni-Druck Publishing Company.
- KUMAR K and WELLKE RJ (1992) Methodology engineering: A proposal for situation specific methodology construction. In *Challenges and strategies for research in systems development* (COTTERMAN WW and SENN JA, Eds.), 257-269, John Wiley & Sons, Washington, DC.
- LARMAN C, KRUCHTEN P and BITTNER K (2001) *How to Fail with the Rational Unified Process: Seven Steps to Pain and Suffering*. Valtech Technologies & Rational Software.
- LOWE D and HENDERSON-SELLERS B (2001) Characteristics of Web Development Processes. In *Proceedings of SSGRR-2001: International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet*, (MILUTINOVIC V, Ed), 21, Scuola Superiore Guglielmo Reiss Romoli, Italy.
- NUSEIBEH R and EASTERBROOK S (2000) Requirements Engineering: A Roadmap. In: A. Finkelstein (ed.), *The Future of Software Engineering*, ACM and IEEE Computer Society Press.
- OMG (2003) *UML 2.0 Superstructure Final Adopted specification*, Document reference ptc/04-10-02 August 2003. Available from the Object Management Group website: [www.omg.org](http://www.omg.org)
- RALYTÉ J, DENECKÈRE R and ROLLAND C (2003) Towards a Generic Model for Situational Method Engineering. *Lecture Notes in Computer Science, Vol. 2681*, Springer-Verlag, 95.
- ROSSI M, SUBRAMAH M, LYYTINEN K, KAIPALA J and TOLVANEN JP (2000) Method rationale in method engineering, *Proceedings of the 33rd annual Hawaii International Conference on Systems Science, IEEE Computer Society Press, USA*.
- SAUER S and ENGELS G (1999) Extending UML for modeling of multimedia applications. *Proceedings of the IEEE Symposium on Visual Languages*, 80.
- SAEKI M (2003) Embedding Metrics into Information Systems Development Methods: An Application of Method Engineering Technique. *CAiSE 2003: The 15th Conference On Advanced Information Systems Engineering*, 374-389.
- SLOOTEN K VAN and BRINKKEMPER S (1993) A Method Engineering Approach to Information Systems Development. In *Information Systems Development Process*. (PRAKSH N, ROLLAND C and PERNICI B, Eds). Elsevier Science Publisher B.V. (North-Holland). 167-186.
- SLOOTEN K VAN and HODES B (1996) Characterizing IS development projects. In *Method Engineering: Principles of method construction and tool support, Proceedings of the IFIP TC8, WG8.7/8.2 Working conference on method engineering*, Atlanta, USA.
- SOUER J, WEERD I VAN DE, VERSEDAAL J and BRINKKEMPER S (2005) Developing Situational Content Management System-based Web Applications. To be published in *Proceedings of the Sixth International Conference on Web Information Systems Engineering (WISE 2005)*, New York City, New York.
- STONEBRAKER M and HELLERSTEIN, JM (2001) Content Integration for E-Business. *Proceedings of ACM SIGMOD.PODS 2002*, Santa Barbara, California.

VIDGEN R, GOODWIN S and BARNES S (2001) Web Content Management. *Proceedings of the 14<sup>th</sup> International Electronic Commerce Conference*, Bled, Slovenia, 465-480.

WEERD, I VAN DE (2005) *Design Methods for WebManager Implementations* (technical report). University of Utrecht, Department of Information and Computing Sciences.

Appendix

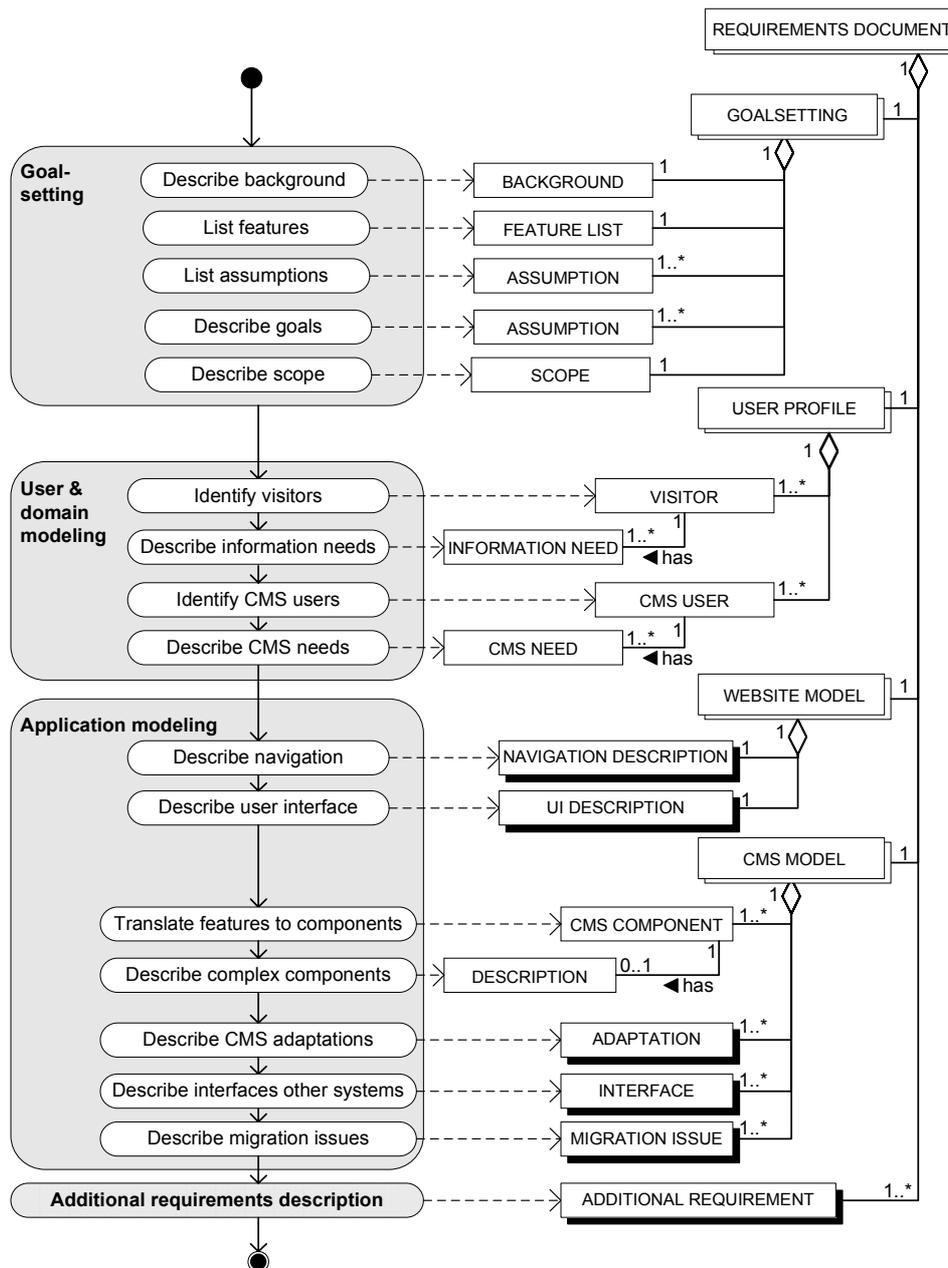


Figure 5: Process-data diagram of the standard definition phase

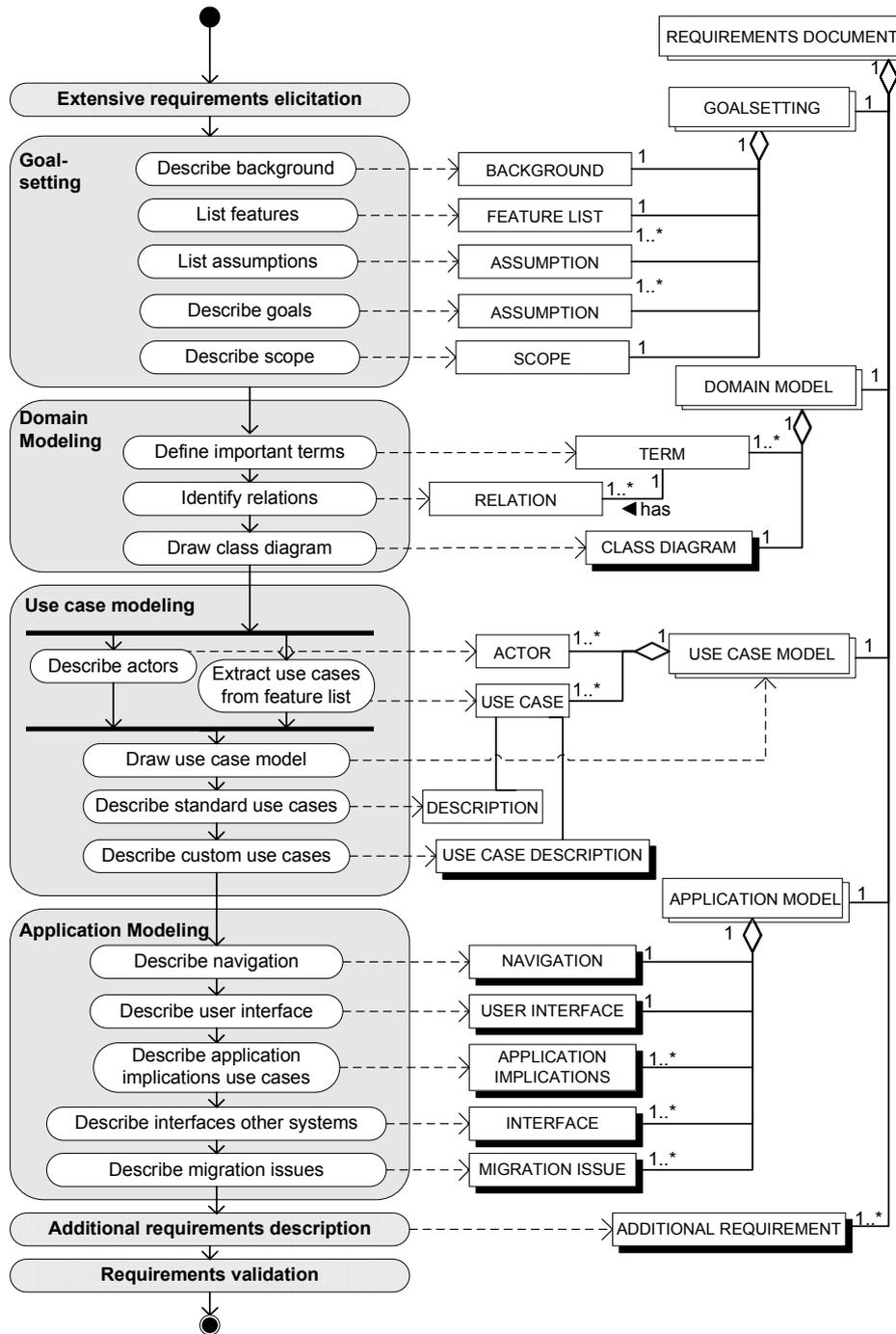


Figure 6: Process-data diagram of the complex definition phase