Situational requirements engineering for the development of Content Management System-based web applications

Jurriaan Souer*

GX Creative Online Development,
Wijchenseweg 111, 6538 SW Nijmegen, The Netherlands
E-mail: jurriaan@gx.nl
*Corresponding author

Inge van de Weerd, Johan Versendaal and Sjaak Brinkkemper

Institute of Information and Computing Sciences
Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
E-mail: i.vandeweerd@cs.uu.nl
E-mail: j.versendaal@cs.uu.nl
E-mail: s.brinkkemper@cs.uu.nl

Abstract: Web applications are evolving towards strong content-centred information systems accessible through the web. The development processes and implementation of these applications are unlike the development and implementation of traditional information systems. In this paper, we propose a Web Engineering Method (WEM): a method for developing Content Management System (CMS)-based web applications. Critical to a successful development of CMS-based web applications is adaptation to the dynamic business. We first define CMS-based web applications and identify their specific characteristics. Combining these characteristics with situational factors in projects, we show that by taking parts of proven methods, such as UML-based Web Engineering (UWE) and the Unified Process (UP), a unique method can be assembled for situational development of CMS-based web applications. We successfully validated the method at GX, a web-technology company specialised in developing and implementing CMS-based web applications.

Keywords: requirements engineering; web application; Content Management System; CMS; method engineering.

Reference to this paper should be made as follows: Souer, J., van de Weerd, I., Versendaal, J. and Brinkkemper, S. (2007) 'Situational requirements engineering for the development of Content Management System-based web applications', *Int. J. Web Engineering and Technology*, Vol. 3, No. 4, pp.420–440.

Biographical notes: Jurriaan Souer is a Consultant and Researcher at GX Creative Online Development. He is responsible for the requirements analysis of CMS-based web applications. He co-established the requirements management team at GX and is Supervisor of the GX graduate programme. He

holds an MSc in Computational Science from Utrecht University and is an industrial PhD candidate in the group of Professor Dr. Sjaak Brinkkemper, Utrecht University. His research interests include business-IT alignment, content management systems, web applications and organisational issues of software implementations. Before GX, he held a research position at Deloitte & Touche.

Inge van de Weerd is a PhD student at the Institute of Information and Computing Sciences at Utrecht University, the Netherlands. She holds an MSc in Information Science from Utrecht University. Her research interests comprise method engineering, metamodelling, product management, information systems development and knowledge management. Her PhD research focuses on the development of a knowledge infrastructure for product software methods that enable situational capability evolution.

Dr. Johan Versendaal is an Assistant Professor at the Institute of Information and Computing Sciences at Utrecht University. He is co-architect of the newly developed Master's programme Business Informatics at the university. His research interests include business-IT alignment, and human factors and organisational issues of (e-)procurement. Prior to joining the university, he was Product Manager at Baan for purchasing applications, and Development Manager in the area of usability; he established the usability team at Baan. Furthermore, he has worked as a Business and Usability Consultant at Atos Origin.

Dr. Sjaak Brinkkemper is a Professor of Organization and Information at the Institute of Information and Computing Sciences at Utrecht University, the Netherlands. Before that, he was a Consultant at the Vanenburg Group and a Chief Architect at Baan. He has held academic positions at the University of Twente and the University of Nijmegen. He holds an MSc and a PhD from the University of Nijmegen. His research interests are software product development, information systems methodology, metamodelling and method engineering.

1 Web application development and implementation

Nowadays, organisations rely on the web to support their business processes and use the internet as a way to create competitive advantage, global collaboration and integration with external partners (Lee and Shirani, 2004). Therefore, applications based on web technology are considered of strategic importance (Turban *et al.*, 1999). These web applications should be agile enough to respond to the dynamic business and the ever-changing customer demands (Arch-int and Batanovv, 2003). The changing business landscape has also impacted on the requirements of systems development approaches (Standing, 2002). However, existing methods for the requirements engineering (*i.e.*, the elicitation, specification and validation of the requirements) of web applications often fail to capture and specify the business dynamics, fail to implement the desired system, and often lack web focus (Lee and Shirani, 2004). Moreover, little attention in requirements engineering has been paid to the development of frameworks and

methodologies for coping with the requirements analyses of web applications (Bolchini and Paolo, 2002). The literature gives a plausible explanation by suggesting that the web applications differ from conventional information systems (de Castro *et al.*, 2004; Ceri *et al.*, 2000; Lee *et al.*, 2004). In fact, a new discipline called 'Web Engineering' has emerged, for the development of web applications (Murugesan *et al.*, 2001).

With the unrestrained growth of unstructured digital information, the need for a system to control the information emerges. Content Management System (CMS)-based web applications are applications that utilise web technology and manage the unstructured information. CMS-based web applications are therefore implemented to support the organisation with the creation, management and publication of information in an efficient and effective way. Examples of CMS-based web applications are Mediasurface, Tridion, Vignette, Documentum, Microsoft Content Management Server, GX WebManager, and the open source products Zope, Mambo and Joomla. Current CMS-based web applications not only contain an extensive collection of functions for the management of digital information, but also transaction, collaboration and portal functions for extending organisational back-end applications and business processes. Figure 4 gives a functional overview of a CMS-based web application.

Since there is no literature on or methods described for CMS, we developed a method specific for CMS-based web applications, constructed by using components of two existing methods: UML-based Web Engineering (UWE) (Koch and Kraus, 2002) and the Unified Process (UP) (Booch *et al.*, 1999). We integrated the method into a generic development method consisting of the phases Acquisition, Orientation, Definition, Design, Realisation and Implementation. To include situationality to deal with the dynamic business, we developed a route map with three different routes, one route for each of the distinguishable project types that we identified: standard projects, complex projects and migration projects.

Our method helps to answer the research question, what is a 'good' method to develop and implement CMS-based applications in different (customer) organisations? Subsequently, we pose the following questions: how do we organise the requirements engineered in the route maps and how can we adapt the method to meet the organisational needs? The paper develops three main contributions:

1 a definition of CMS-based web applications, derived from Enterprise Content Management Systems and Web Information Systems

2 a Web Engineering Method (WEM) as a specific development method for CMS-based web applications

3 a notation that we introduce (process-data diagram) for specifying the development method.

We start by providing an overview of the differences of developing web applications and traditional information systems given their characteristics. In Section 3, we give an overview of our method, starting with the construction. We validate the method in four projects, which we describe in Section 4. Section 5 provides an overview of related work. Finally, we present our conclusion in Section 6.

2 Defining CMS-based web applications

In developing a method for specifying CMS-based web applications, we first clarify the concepts we are discussing. Moreover, we identify their relationships and the differences, particularly the characteristics that influence the requirements engineering process. We first provide an overview of the concepts, whereupon we give a definition.

There are two developments in the last ten years that have gradually led towards CMS-based web applications. One is the growing use and utilisation of the World Wide Web as a significant communication channel. Organisations needed to expand their information systems beyond the boundaries of their organisation, resulting in web information systems and web applications. The other development leading towards CMS was the unrestricted growth of digital content, the large amount of which resulted in a lack of information control and loss of data. New information systems were built to cope with the digital content. These information systems evolved towards Enterprise Content Management Systems (ECM), which encompass Digital Asset Management (DAM), Document Management (DM) and Web Content Management (WCM). The last was developed specifically to cope with web content. WCM and web applications merged gradually into a single application. Figure 1 illustrates a categorisation from information systems to CMS-based web applications.

The definition of a web information system is not unambiguous. There are several definitions of web information systems according to Holck (2002). Yet there seems to be a consensus that web information systems rely on the web for a correct execution. We therefore define *Web Information Systems* as a special type of information system which utilises the technology of the web. Holck argues that it is unclear in what ways the development of web information systems is supposed to be new and different. He describes the four most-often mentioned characteristics of web information systems:

1 the new incremental development process

2 the time pressure

3 the new professions

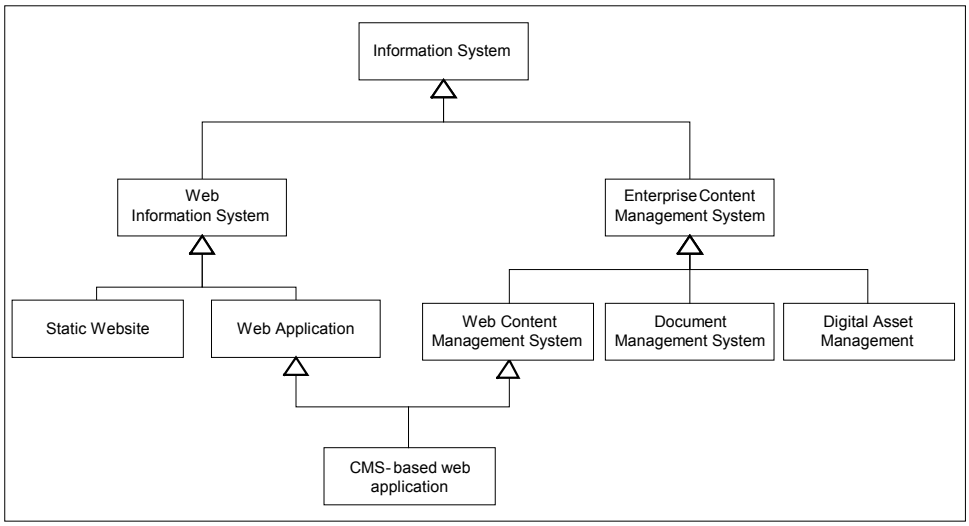
4 a diverse and remote user group.

Taylor *et al.* (2002) found that *ad hoc* development of websites seems to dominate in the industry, which suggests that formal information system development methods are rarely used.

There are numerous web information systems, and not all of them are web applications. In this paper, we use the following definition, derived from the definition of Gnaho (2001): a *web application* is an information system that provides facilities to access complex data and interactive services via the web and changes the state of business. A web application implements business rules and has a certain functionality to either interact with another actor (person or system) or change its own state. In a similar way, De Troyer and Leune identified two kinds of web information systems: a kiosk for presenting plain information (illustrated in Figure 1 as a 'Static Website'), and a web application as a kind of interactive information system (de Troyer and Leune, 1998). De Castro *et al.* (2004) and Jeenicke *et al.* (2003) recognise one of the challenges of

specifying web applications: unlike traditional information systems, in web applications the client and the users are not the same people. Another study recognised that the challenge stems from the fact that the internet, and more specifically, the web, is a completely different computing environment compared to conventional computer-based environments (Arch-int and Batanovv, 2003). There are a few methods and tools available for developing web applications, such as WebML and W2000, as mentioned in Section 5.

Figure 1 Positioning of CMS-based web applications in the information systems categorisation



As the volume of digital content grows, web applications evolve towards more content-centred web applications. Vidgen *et al.* (2001) identified a list of issues arising due to the content growth: information consistency, navigational aspects, data duplication, content audit and control, tracking of content and mapping the website work flows on the business processes. To manage the information, state-of-the-art content-centred web applications rely on a CMS, which can be defined as a tool for the creation, editing and management of information in an integral way (Fernandez-Ilgesias *et al.*, 2005). We define *CMS-based web applications* as a web application for the management and control of content. Typical characteristics of CMS-based web applications are a strict separation of content, structure and graphical design, a content repository for the reuse of information, and an integrated work flow for supporting the content life cycle. With the inclusion of web application in our definition, we imply that CMS-based web applications utilise the technology of the web and implement business logic. As organisations demand more functions to interact with their end users (customers), CMS-based web applications have evolved to a complete set of transactional functions by which multiple users of an organisation can manage their complete online strategy. With a configured CMS-based web application, organisations can provide personalised information to customers; support self-service modules which can be

integrated with back-office applications (*e.g.*, a web service); implement e-commerce functionalities; and manage the information flow for multiple channels (web, PDA, cellphones, RSS, TV screens) in a controlled manner. To elicit, specify and validate all relevant requirements for a CMS-based web application, a thorough understanding of the organisation and its customers is needed. To our knowledge, no specific tools and methods exist for the development of CMS-based web applications.

Note that, as with IS development and implementation of product software like Enterprise Resource Planning (ERP) systems, there are standard modules (commercially) available for CMS-based applications, allowing organisations to implement and customise the applications according to their requirements. In many cases, the development of these applications does not start from scratch.

3 Constructing a new method with method engineering

Current development methods are not capable of coping with the dynamics of CMS-based web applications and engineering the requirements, as described earlier. Therefore, a new method is needed for situational CMS-based web applications. The WEM uses a development method based on six phases: acquisition, orientation, definition, design, realisation and implementation.

3.1 The method engineering process

To construct a new method, we used fragments of existing methods to evaluate earlier work and combined useful parts into our existing method. The description of the method engineering approach and the used method fragments goes beyond the scope of this paper. For an elaboration of the method engineering approach, see Brinkkemper *et al.* (1999) and van de Weerd *et al.* (2006). We will, however, give a brief description of the two main methods we used as a source for our methods: the Unified Process (UP) and UWE.

The UP is a generic process framework that can be specialised for a very large class of software systems, for different application areas, different types of organisations, different competence levels and different project sizes (Booch *et al.*, 1999). The UP is a methodology which developed into a very extensive framework for the development of a software product. Web applications are a specific type of software. Therefore, we used the UP for the construction of our situational method for the development of web applications.

The UWE methodology provides a systematic approach to the development of web applications (Koch and Kraus, 2002). UWE is also based on the UP and is therefore a good starting point for our development method. Some differences with the UP are the specialisation of the UP for the development of web applications; the extension of the development cycle with a maintenance phase; the addition of two supporting work flows, project management and quality management; and extending quality control management with requirements validation and design verification in addition to testing.

3.2 Project categorisations

We categorised CMS-based web application project situations based on distinguishing characteristics of the project requirements, such as the ability to realise the requirements of the customer with standard functionalities of the CMS-based web application. Other requirements need some customisation of the CMS-based web application. The projects can be divided into three categories, based on project type and complexity:

1 *Standard projects* – projects that are mostly based on existing standard functionalities of CMS-based web applications

2 *Complex projects* – projects based on existing functionalities with lots of customisation or new build functionalities

3 *Migration projects* – projects that involve an upgrade of an older version of a CMS-based web application to a newer version.

Standard and complex projects are not that clearly defined and are arbitrarily distinguished by their characteristics. In this paper, we focus on the standard and complex projects.

Table 1 Elements of standard and complex projects

Phase	*Standard*	*Complex*
Acquisition	Acquire customer information	Acquire customer information
	Describe solution	Feature list (UP)
		Describe solution
Orientation	Risk management (UWE)	Risk management (UWE)
Definition	Product vision (UP)	Product vision (UP)
	User and domain modelling (UWE)	User and domain modelling (UWE)
	Application model (UWE)	Use-case modelling (UP)
	Nonfunctional requirements (UP)	Application model (UWE)
		Nonfunctional requirements (UP)
		Requirements validation (UWE)
Design	Web application architecture (UP)	Custom architecture (UP)
		Technical design in detail (UP)
Realisation	Implement graphical design	Implement graphical design
	Configuration of web application	Configuration of web application and extensions
	Functional testing	
		Iterative development of custom functions in components (UP)
		Iterative development of interfaces (UP)
		Product quality assurance (UP)
		Functional and integration testing
Implementation	Production deployment	Staging deployment
	Acceptance	Production deployment
		Acceptance

Ideally, a standard project is a solution completely based on the existing CMS-based web application. In this case, the CMS-based web application should meet the customer's needs. In a standard project, requirements analysis consists of the identification of the required standard functionalities and the configuration of these components. Projects within the complex route map can be very special. Existing functionalities are not sufficient and customisation is needed to create the required functionalities. The requirements capturing and specification is more challenging in complex projects.

Table 1 gives an overview of the standard and complex route maps and in fact depicts our new method. On the left, a general development method is described, defining the rows of the table. At the top, the standard and the complex route map define the two columns. Each cell of the matrix gives an overview of the applied methods in the route map per phase. The actual requirements specification is done in the definition phase.

3.3 The resulting web engineering method

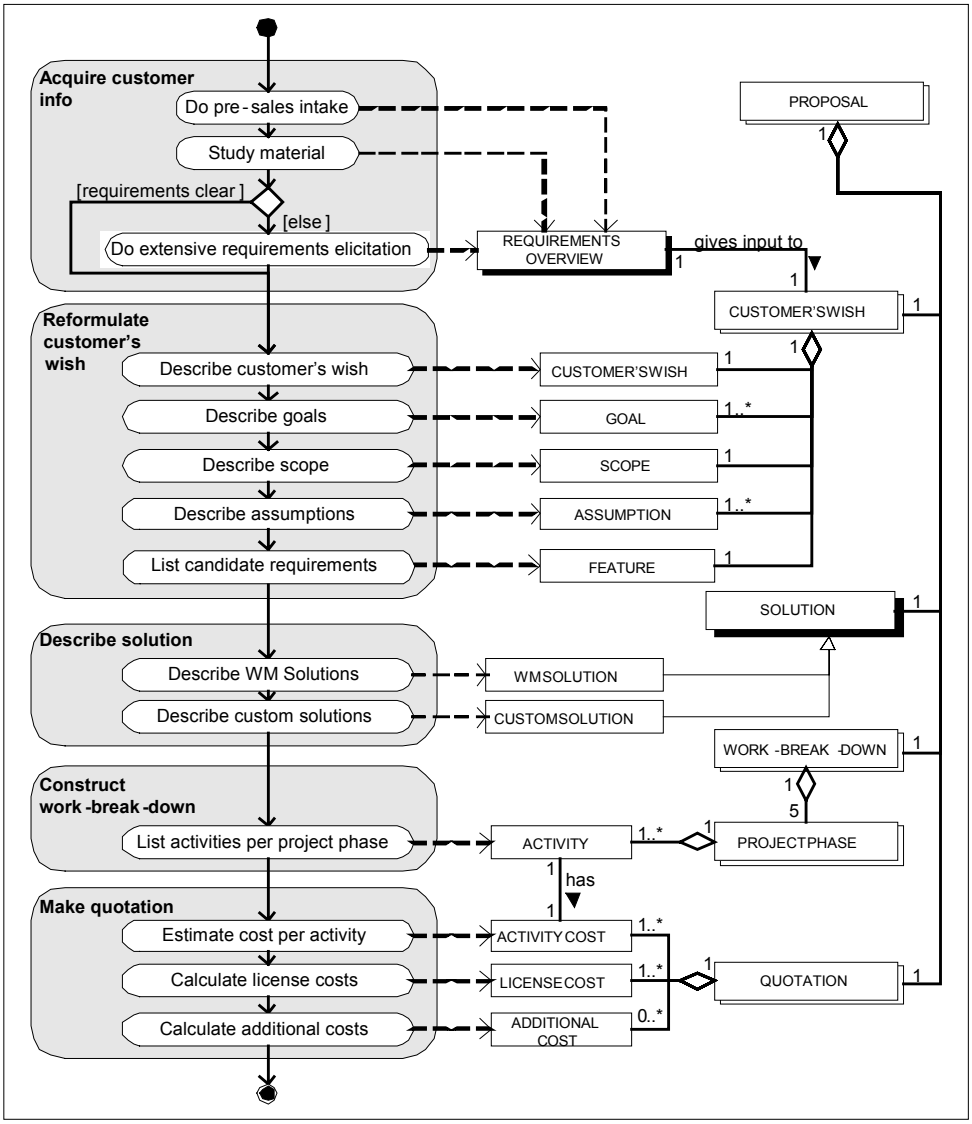
We refer to the method resulting from the method engineering process as the Web Engineering Method (WEM). In the following, we give an overview of the different phases of WEM. We focus on the first three phases (acquisition, orientation and definition).

3.3.1 Acquisition phase

The acquisition phase focuses on outlining the customer's wish into a proper solution. Through interviews with the customer (or workshops, documents, *etc.*), a primary understanding of the desired web application is gathered and an idea of the project environment arises. Two of the earlier mentioned shortcomings of traditional implementation methods are the inability to specify the right requirements of CMS-based web applications and trouble mapping the application work flow with the business processes. To cope with this issue, we introduced a fragment of the UP into our existing development method, consisting of a feature list. If the project appears to be complex, a feature list identifies the candidate key requirements and business processes in an early stage, which need to be addressed by the web application. CMS-based web applications provide an extensive set of standard functionalities, and therefore some candidate requirements match or resemble available functionalities. Based on the information acquired from the customer, the customer's wish is then reformulated in an unambiguous way, understandable to all project members. It is then clear whether the project is standard or complex. With that knowledge, a solution for the project is described. The figure below illustrates the introduction of a feature list in the acquisition phase.

Figure 2 illustrates the metamodel of the acquisition phase in WEM. For the method visualisation, a metamodelling technique is used (van de Weerd *et al.*, 2006) based on UML.[1] The processes are modelled on the left-hand side in UML activity diagrams, and the corresponding data on the right-hand side in UML class diagrams. An important adjustment of the standard UML concerns the use of different types of concepts to indicate whether a concept is simple or compound. A simple concept (*e.g.*, GOAL) does not contain any subconcepts, whereas a compound concept is an aggregate of subconcepts. Compound concepts can be closed (*e.g.*, REQUIREMENTS OVERVIEW) or open (*e.g.*, PROPOSAL), depending on the relevance of showing the subconcepts.

Figure 2 Metamodel of the acquisition phase in WEM



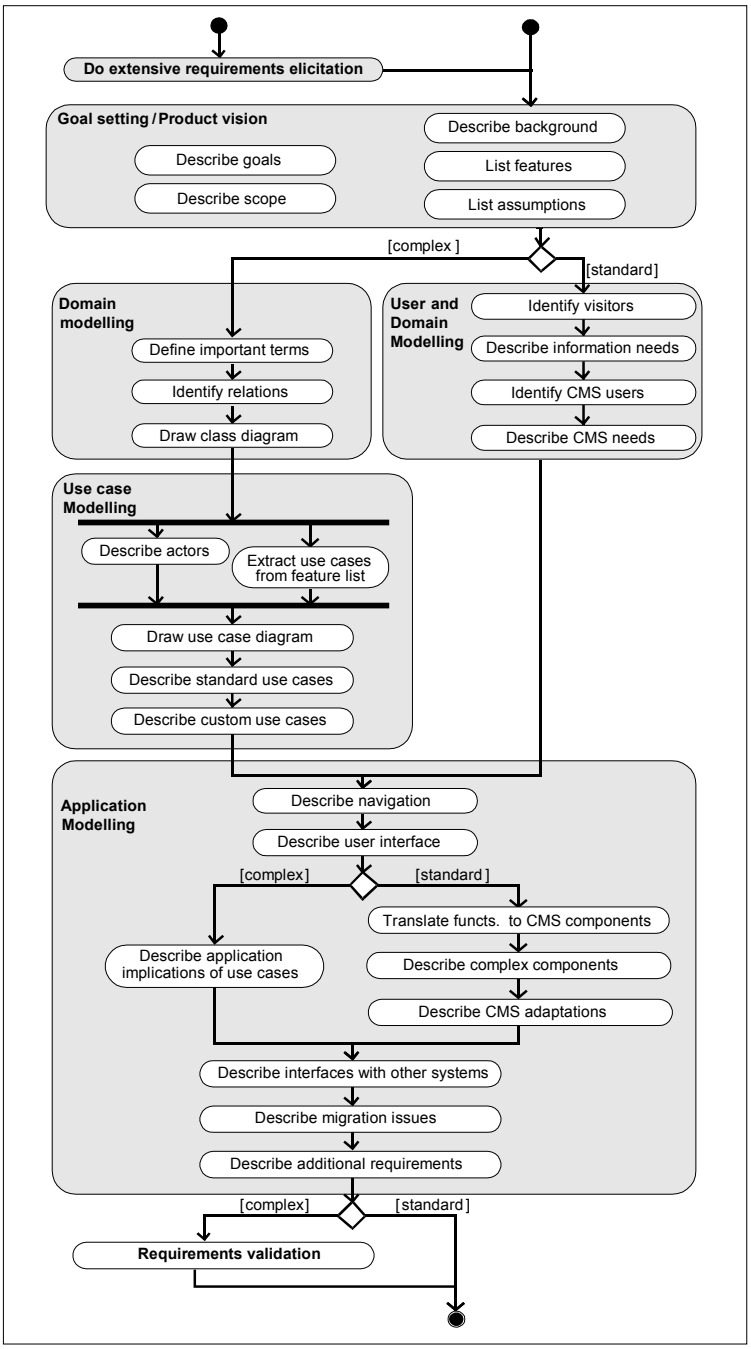
3.3.2 Orientation phase

The project starts in the orientation phase. Organisational aspects are defined, such as participants, targets, products, scope and assumptions. Furthermore, project management aspects are installed, including planning and control of the project, communications, activities and responsibilities, risk management, problem management and change management.

3.3.3 Definition phase

The definition phase of WEM consists of the actual requirements elicitation, analysis and specification. A metamodel of the definition phase is illustrated in Figure 3. Both the standard and the complex projects are integrated. For readability, we left out the data side of the diagram.

Figure 3 Metamodel of definition phase in WEM



In the definition phase, the analyst focuses on defining what should be built. The requirements analysis starts with the creation of a product vision. We used the product vision of the UP to document the purpose and the goal of the web application. A product vision includes a background description, the goals of the web application, the scope of the project, and some assumptions and limiting conditions. If available, the feature list is detailed and further specified. To cope with the lack of web focus, we adapted user and domain modelling from UWE to define the users and the environment. We identified the different types of users (*e.g.*, visitors, registered visitors, editors, content managers) and their information need. Functions derived from the users and their needs are compared to functions in the CMS-based web application. This application modelling results in a configuration of the CMS-based web application. An application model in WEM consists of a navigational description, user interface description, functional mapping, work-flow modelling and a utilisation of content reusability. Finally, the nonfunctional aspects are defined. Typical examples are user management, security, scalability, performance, design conditions, backup and logging.

For the complex components, we added domain modelling and use case modelling to specify the requirements of complex issues. Use case modelling originates from the UP and describes the requirements as an interaction between actors and the system. Therefore, all actors and their functions are identified. To simplify communication, a use case diagram was developed to show all the functions of the actors in one diagram. This gives a customer an overview of the web application and minimises the chances of implementing the wrong requirements. In complex projects, the requirements are regularly discussed with the client to specify the requirements in the right way.

3.3.4 Design phase

During the design phase, it is determined how the requirements are realised. Based on the requirements, a suitable architecture is created. Standard projects will ideally be fully integrated into the CMS-based web application; hence, the architecture of the web application is then the actual architecture of the CMS-based web application. In complex projects where customised components are developed, a customised architecture is necessary. Still, complex architectures utilise the standard web application architecture. We used the 4 + 1 view architecture of the UP, *i.e.*, the logical view, process view, implementation view and deployment view.

3.3.5 Realisation phase

During the realisation phase, the actual web application is created. The graphical user interface design is then integrated in the CMS-based web application and the relevant functions in the web application are configured to meet the customer's requirements. Depending on the complexity, one or more iterations are used to realise the desired functions. All the realised components are eventually tested, based on the test plan. If the components succeed in the functional and integration tests, the realisation phase is concluded.

3.3.6 Implementation phase

CMS-based web applications are generally deployed straight to production. Some customers demand a staging environment – or a complete DTAP (Development, Test,

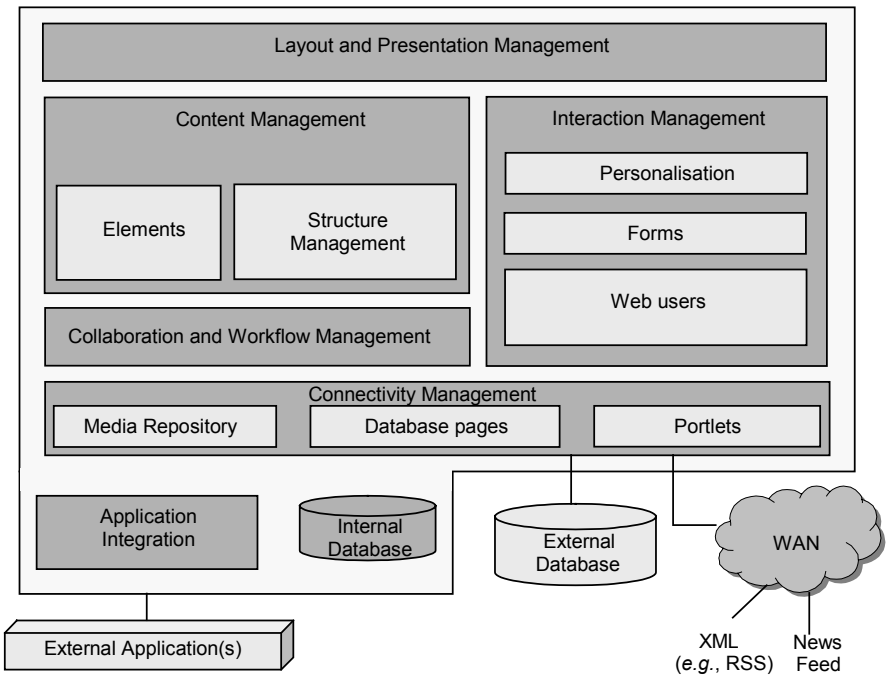
Acceptance, Production) configuration – before actual deployment to production, which can be the case in complex projects. Customers then test the product themselves before they accept the project.

WEM has different elements for standard and complex projects. A project is rarely fully standard or complex. Most projects have certain functionalities which are fully supported by the existing CMS-based web application, but have minor extensions or customisations. Each project should at least contain the activities as described in the standard route map. WEM is then tailored to the situation at hand. Because WEM was created by reusable fragments and in itself is developed in components, one can easily add components from the complex route map to the standard route map, creating an optimal development method in a given situation. WEM is thereby suitable for maturation, adding new components as new development methods arise. Clearly, not all activities in the complex route map can be selected individually. For instance, product quality assurance is not useful if no custom functions or interfaces are developed. In addition, use case modelling without a domain model is quite difficult to read. An important cross-check is the consistency between the domain model and the use cases, as every concept described in the domain model should be mentioned in at least one use case. Otherwise, the described concept is irrelevant.

4 Validation

To validate the newly developed method, we applied WEM in three projects at GX Creative Online Development, a web technology company in the Netherlands. GX develops and implements its Content Management System *GX WebManager* to deliver CMS-based Web Applications, which is described in van Berkum *et al.* (2004). Figure 4 illustrates a high-level functional architecture of the GX WebManager.

Figure 4 High-level functional architecture of GX WebManager



In the validation, we focused solely on the definition phase, comprising the requirements specification part of the overall implementation. We validated WEM by analysing project aspects such as realisation within the time and budget, user satisfaction of internal stakeholders (architects, engineers and project managers of GX) and user satisfaction of external stakeholders (customers, graphical designers). The user satisfaction was determined through interviews with the relevant stakeholders. The questions concerned the process of requirements engineering (the structuring and managing of the process) and the final requirements specification (soundness, completeness and readability). The results were then compared to previous projects, which were not based on WEM. The customers we used in the validation consist of:

- a retail organisation, which operates throughout Europe in the marketing, sales and distribution of home entertainment products
- a large telecommunications company
- a health insurance company (standard project, multiple sites).

In Table 2, an overview of the three cases is shown with the project characterisations. The last two columns describe the number of actors and use cases in the requirements document. Note that case number 3 also has a few use cases, despite the characterisation of a standard project. We will elaborate on one case. The other cases are discussed in Section 4.2.

Table 2 Overview of case studies

Case	*Project type*	*Estimated man-hours*	*Estimated dev. time*	*Actors*	*Use cases*
1	Complex	2000	7 months	9	29
2	Complex	400	2 months	7	17
3	Standard	1500	5 months	*3*	*3*

4.1 Case description

The GX customer operates throughout Europe in the marketing, sales and distribution of home entertainment products. For their clients, the customer wanted to develop an integrated multichannel platform consisting of a web shop with physical and digital products and an in-store marketing application called narrowcasting. This multichannel platform had to be managed in a web application, managing the content on several web shops and in-store marketing of all retail clients. The products for the web shops and the in-store marketing were provided through multiple interfaces with product-data suppliers, and there were two fulfilment partners for order handling. Specifying such a highly dynamic and specific content-driven web application is clearly a complex project.

Based on the new WEM method, the following process steps were applied during the requirements specification:

- goal setting/product vision (background, feature list, assumptions)
- domain modelling (terms, relations, class diagram)
- use case modelling (actors, use case diagram, use cases)

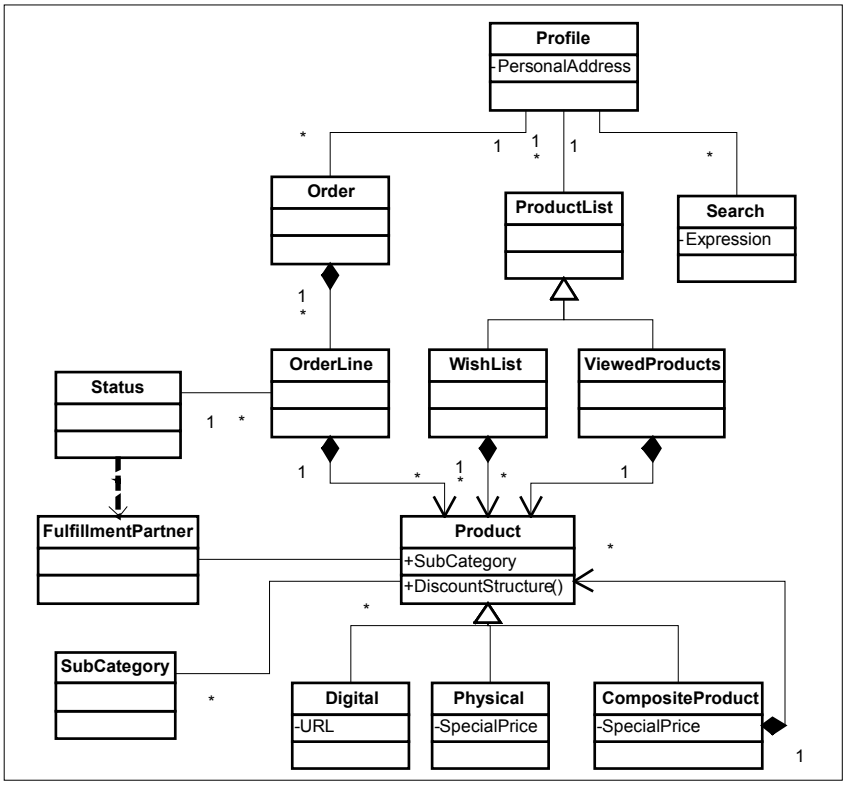
- application modelling (navigation, interfaces, application implications, additional requirements)
- requirements validation.

Before using WEM, GX used primarily application modelling for the requirements engineering. In interviews, the graphical design was translated to functions, and additional requirements were specified (such as defining the web forms, user authorisation and work-flow process).

With WEM, all the activities described in the complex route map were applied. Because this project was complex, a feature list was created in an early stage, comprising the core functions of the web application. Some examples are, 'Visitors need to register before they can buy any products', 'Visitors have a wish list' and 'Products in the web shop are both digital and physical'. With the feature list, the scope of the project was defined.

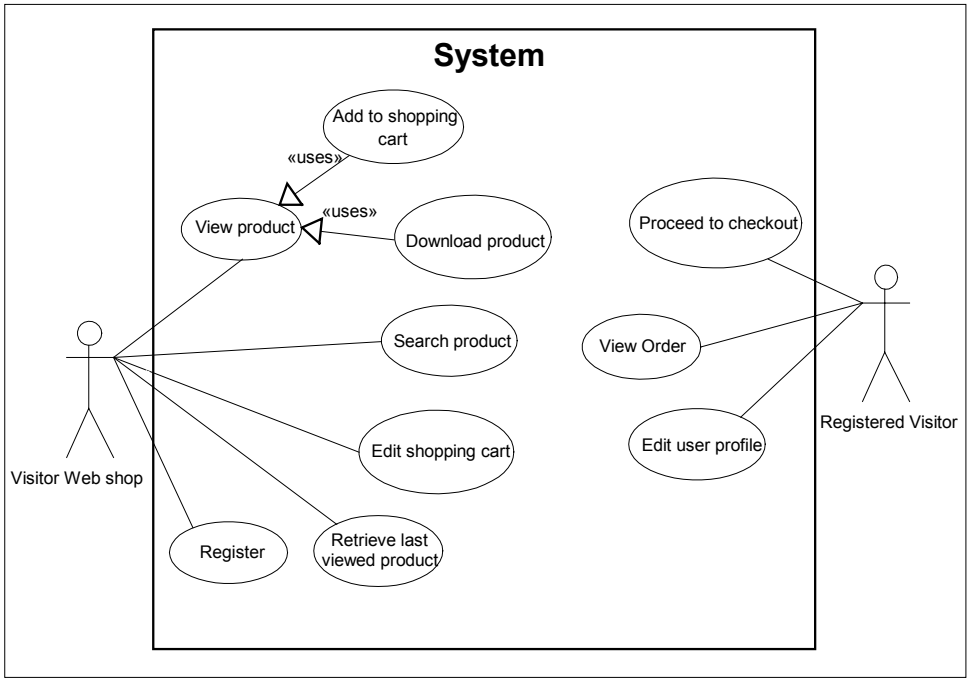
During the requirements analyses in the definition phase, a product vision was defined in collaboration with the customer. Then, all the concepts and their interrelationships were identified to prevent miscommunication. These concepts and their relationships are modelled in a domain model. In Figure 5, part of the domain model is visualised.

Figure 5 Part of the domain model



All the functional requirements were then specified with use cases. First, the actors and a basic set of use cases were identified, based on the feature list. There were, for instance, different types of users in the web shop – distinguished by their status (registered or not). Different users have different functions. Through iterations, the set of actors and use cases grew. By means of use cases, the requirements for each actor could clearly be defined, as specified by the customer. Figure 6 illustrates a component of the use case diagram.

Figure 6 Use case diagram



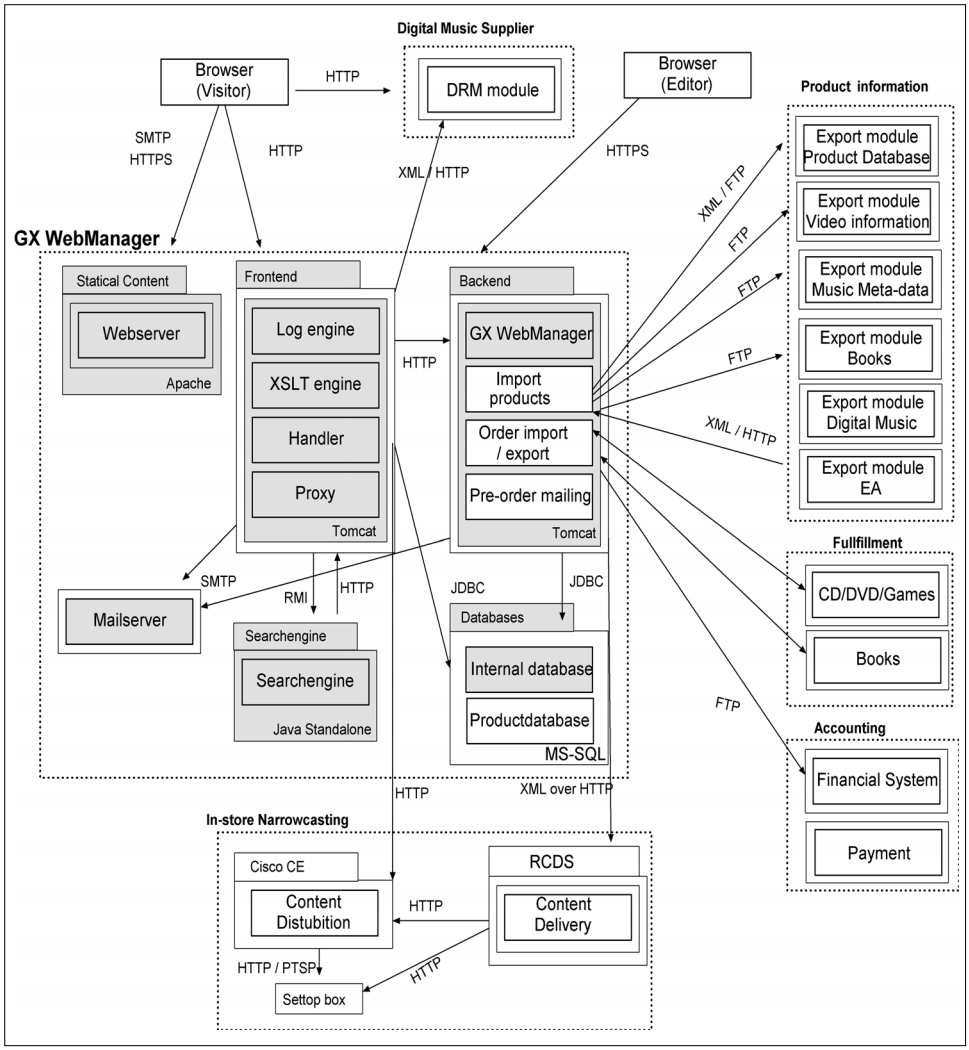
When the use cases were completed, the process continued with application modelling, which translated use cases and nonfunctional requirements into the CMS-based web application. Navigational design was developed based on the use cases. The use cases then needed to be translated into existing functions of the CMS-based web application. With the application modelling, certain requirements were adapted to be in line with the existing functions. (For example, the customer wanted a 'frequently asked questions' function with some specific features. The CMS-based web application has a standard FAQ function but not exactly in accordance with the requirements. Yet, while not fully compliant with the original requirements, choosing the standard functions over customisation saved time, effort and money for the customer.)

For each type of interface, we specified an actor and use cases: multiple product-data imports, multiple fulfilment exports, a payment services provider, a web application for the delivery of narrowcasting content, back-end applications (financial system, among other systems) and a music-server for the digital music files. With such an extensive web application, a process of the work flow for the creation and the delivery of content was developed; the editors could create product views and in-store marketing material within

the web application and with a controlled procedure. To support tracking of content and order creation, an audit trail was designed for tracking all the changes. A few additional requirements concerning financial transactions were specified and that concluded the requirements specification. To complete the process, the requirements specification was validated with GX internally and with the customer before it was finalised.

After the definition phase comes the design phase, in which GX creates the architectural design of the CMS-based web application based on the specified requirements. Figure 7 shows the process view of the architecture, illustrated to give an idea of the system, which was thereafter realised and implemented.

Figure 7 Process view



4.2 Discussion

We described one of the three cases in the previous section. Here we will give an overview of the findings of the three cases.

To start, the external and internal stakeholders were pleased with the specifications and the final CMS-based web application. The customers mentioned that the resulting CMS-based web application was what they had hoped for and conformed to their specifications. Internal stakeholders at GX were also very satisfied with WEM. GX acknowledged that, compared to before the development of WEM, the developers had a better understanding of the customer environment because of the product vision and the user and domain modelling. Moreover, they had a better understanding of the overall purpose and the requirements of the system, especially the complex functionalities that were specified with use cases and translated with application modelling. Summarised, WEM made a significant contribution towards realising these complex projects within time and budget, and with good results and satisfied customers.

There were, however, some shortcomings due to the separation of processes for standard and complex situations. For instance, the health insurance company was specified within the standard route, yet they asked for some elaboration of the requirements in the form of use cases, which was actually part of the complex route (hence, the three use cases in Table 2). The reason for this was that although they wanted the standard functionalities, they found it hard to imagine the actual specification without the use cases. In addition, the acceptance of one of the complex projects was challenging because the customer had other expectations concerning the functions in the CMS-based web applications. While discussing this with the customer, we found that these expectations could have been managed with two processes of the application modelling within the standard route: the translation of the functions to the CMS-components and a description of the CMS adaptations.

Some lessons can be learned from these cases. The first one is that not everyone can easily interpret use cases. The customer indicated that they could use a little assistance with interpreting the use cases. Therefore, it is very important that a supplier ensures that the customer really understands the specifications and implications of the use cases. On the other hand, some standard functionalities are so advanced that use cases can help the customer to understand the functionality. It is evident that good communication with the customer is crucial.

Another lesson can be learned from looking at when a standard functionality is used or is customised to the customer's needs. That standard functionality is preferably specified with a standard approach, since it makes the CMS-based web application components and their adaptations explicit compared to using a use case, which essentially can describe any interaction between an actor and the CMS-based web application. Moreover, the functional boundaries are known within the standard approach. These boundaries can be communicated to the customer to manage their expectations and thereby increase the chances of success.

5 Related work

Several methods and techniques have been developed for designing and implementing web applications. For an excellent overview of some different development methods for web applications which include a requirements phase, see Escalona and Koch (2004). One of these methods is the Website Design Method (WSDM) (de Troyer and Leune, 1998). WSDM is a user-centred method for the design of kiosk websites. A kiosk website mainly provides information and allows users to navigate through that information. The two basic characteristics of WSDM are the audience-driven approach and the explicit conceptual design phase. The conceptual design can be performed using techniques like OMT or E-R modelling. De Troyer and Leune identified two types of websites: a kiosk for just presenting information, and a web application, which is a highly interactive information system where the interface is formed by a set of web pages. WSDM focuses on kiosk websites, which are mainly developed to provide information on the web. We, on the other hand, focus on a specific type of web application. WSDM does not include transactional requirements and user interface requirements (Escalona and Koch, 2004), which is an integral part of the CMS-based web applications requirements engineering process. It also concentrates on the end user (or 'visitor') of the system, while CMS-based web applications also need to support the users of the system (or 'editors'). This makes WSDM unsuitable.

Sauer and Engels developed the Unified Modelling Language (UML) Extension for Modelling Multimedia Applications. Aspects of the application, which are covered in this extension, are logical structure, spatial presentation, predefined temporal behaviour and interactive control. Another extension was developed by Baumeister *et al.* (1999). They propose the UML Extension for Hypermedia Design, because the diagrams of UML are not sufficient to model such aspects as navigational space and graphical representation. While they present an interesting modelling tool to visualise Hypermedia design, they do not provide us with a proper development method to capture all relevant requirements.

WebML is a high-level specification language for designing data-intensive web applications (Ceri *et al.*, 2000). Its specification consists of four perspectives: the structural model, the hypertext model, the presentation model and the personal model. It is not based on UML, but it is compatible with such existing notations as E-R modelling and UML. WebML is a tool to design web applications and it can be used in several development methods. CMS-based web applications, however, have their own data structure to manage the information. However, WebML does have comparable fragments in the requirements engineering process, such as the identification of internal and external users, use cases, data requirements, personalisation requirements, nonfunctional requirements and even multichannel requirements. Still, aspects such as application modelling, transactional requirements and migration requirements are not covered. Moreover, WebML is developed for designing data-intensive web applications.

Finally, W2000 is a framework for designing web applications based on the preexisting assets UML and the Hypermedia Design Model (HDM). According to the authors, the integration between UML and HDM consists of four methods: defining several stereotypes and customisations of diagrams to render HDM with UML, specifying guidelines to use UML as a way to specify some of the dynamic and operational aspects of web applications, refining use case diagrams to describe high-level user requirements, and addressing issues related to both functional and navigational aspects (Baresi *et al.*, 2001). W2000 provides some useful methods which

we also use in WEM, such as use case modelling and the information design. Still, W2000 is far from being a complete development method for it does not include data requirements, user interface requirements and nonfunctional requirements (Escalona and Koch, 2004), which are a necessity in the requirements engineering of CMS-based web engineering.

6 Conclusion and future work

In this paper, we gave a definition and categorisation of CMS-based web applications. We described the WEM, a method for developing situational CMS-based web applications. The focus was on the aspects of requirements engineering. We used method engineering to develop WEM, based on components of two existing methods: UP and UWE. We validated WEM and the results show that WEM seems a promising approach for developing complex web applications. To make strong statements on the applicability of WEM, more research is needed.

We are now extending the method with the migration route and will further validate and optimise the three routes. This can be realised by continuing the collection of improving method fragments and eliminating redundant and invalid fragments. Every iteration of method assembly leads to a more suitable implementation method.

Ongoing work is focusing on clearly defining the concepts that are relevant within the CMS-based web applications. In addition, we are developing a modelling technique to configure these types of web applications and thereby attempting to improve the overall implementation process.

Another aspect we are currently researching is the development of reference models for CMS-based web applications. Through the same manner in which we developed three route maps for the different types of projects, different route maps can be developed for different types of clients. Mapping the client's needs to product extensions is one of the future work directions.

References

Arch-int, S. and Batanovv, D.N. (2003) 'Development of industrial information systems on the web using business components', *Computers in Industry*, February, Vol. 50, No. 2, pp.231–250.

Baresi, L., Garzotto, F. and Paolini, P. (2001) 'Extending UML for modeling web applications', *Proceedings of 34th Annual Hawaii International Conference on System Sciences (HICSS'34)*.

Baumeister, H., Koch, N. and Mandel, L. (1999) 'Towards a UML extension for hypermedia design', in R. France and B. Rumpe (Eds.) *Proceedings UML '99*, LNCS, Vol. 1723, pp.614–629.

Bolchini, D. and Paolo, P. (2002) 'Capturing web application requirements through goal-oriented analysis', *Proceedings of V Workshop on Requirements Engineering WER'02*, Valencia, Spain.

Booch, G., Rumbaugh, J. and Jacobson, I. (1999) *The Unified Software Development Process*, Redwood City, CA: Addison Wesley Longman Publishing Co., Inc.

Brinkkemper, S., Saeki, M. and Harmsen, F. (1999) 'Meta-modelling based assembly techniques for situational method engineering', *Information Systems*, May, Vol. 24, No. 3, pp.209–228.

Ceri, S., Fraternali, P. and Bongio, A. (2000) 'Web Modeling Language (WebML): a modeling language for designing web sites', *Computer Networks*, June, Vol. 33, Nos. 1–6.

De Castro, V., Marcos, E. and Cáceres, P. (2004) 'A user service oriented method to model web information systems', *Proceedings of the 5th Conference on Web Information Systems Engineering (WISE 2004)*.

De Troyer, O.M.F. and Leune, C.J. (1998) 'WSDM: a user centered design method for web sites', *Computer Networks and ISDN Systems*, April, Vol. 30, Nos. 1–7, pp.85–94.

Escalona, M.J. and Koch, N. (2004) 'Requirements engineering for web applications – a comparative study', *Journal of Web Engineering*, Vol. 2, No. 3, pp.193–212.

Fernandez-Ilgesias, M.J., Alveraz-Sabucedo, L.M., Lojo-Silva, C., *et al.* (2005) 'Generating high quality printouts from content management systems: a cost-effective approach', *Software Practice and Experience*.

Gnaho, C. (2001) 'Web-based information systems development – a user centered engineering approach', *Lecture Notes in Computer Science*, Vol. 2016, pp.105–118.

Holck, J. (2002) '4 perspectives on web information systems', *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS '03)*.

Jeenicke, M., Bleek, W. and Klischewski, R. (2003) 'Revealing web user requirements through e-prototyping', *Proceedings of the Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE03)*.

Koch, N. and Kraus, A. (2002) 'The expressive power of UML-based web engineering', in D. Schwabe, O. Pastor, G. Rossi and L. Olsina (Eds.) *Second International Workshop on Web-oriented Software Technology (IWWOST02)*, pp.105–119.

Lee, C., Suh, W. and Lee, H. (2004) 'Implementing a community web site: a scenario-based methodology', *Information and Software Technology*, Vol. 46, pp.17–33.

Lee, S.C. and Shirani, A.I. (2004) 'A component based methodology for web application development', *Journal of Systems and Software*, April, Vol. 71, Nos. 1–2, pp.177–187.

Murugesan, S., *et al.* (2001) 'Web engineering: a new discipline for development of web-based systems', *WebEngineering 2000*, LNCS 2016, pp.3–13.

Standing, C. (2002) 'Methodologies for developing web applications', *Information and Software Technology*, March, Vol. 44, No. 3, pp.151–159.

Taylor, M.J., McWilliam, J., Forsyth, H., *et al.* (2002) 'Methodologies and website development: a survey of practice', *Information and Software Technology*, April, Vol. 44, No. 6, pp.381–391.

Turban, E., McLean, E. and Wetherbe, J. (1999) *Information Technology for Management*, 2nd ed., New York: John Wiley and Sons.

Van Berkum, M., Brinkkemper, S. and Meyer, A. (2004) 'A combined runtime environment and web-based development environment for web application engineering', *Advanced Information Systems Engineering: 16th International Conference, CAiSE 2004, Riga. Lecture Notes in Computer Science*, Vol. 3084, pp.307–321.

Van de Weerd, I., Brinkkemper, S., Souer, J. and Versendaal, J. (2006) 'A situational implementation method for web-based content management system-applications: method engineering and validation in practice', *Software Process: Improvement and Practice*, Vol. 11, No. 5, pp.521–538.

Vidgen, R., Goodwin, S. and Barnes, S. (2001) 'Web content management', *Proceedings of the 14th International Electronic Commerce Conference*, Bled, pp.465–480.

Note

1 OMG (2003) 'UML 2.0 superstructure final adopted specification, document. reference ptc/04-10-02, available from the Object Management Group website, www.omg.org.

Appendix

Selection of websites realised with GX WebManager

- Ajax – a consumer-oriented fan site of the famous Dutch football club:
  http://www.ajax.nl/
- KPN.com – a consumer and business-oriented site of the market leader in the major segments of the Dutch telecom market:
  http://www.kpn.com
- Planet Technologies – the web portal of the largest internet services provider in the Netherlands:
  http://www.planet.nl/
- Daimler Chrysler Nederland – a consumer-oriented website for Mercedes:
  http://www.mercedes-benz.nl/
- Talpa TV – a website of a commercial television and multimedia company:
  http://www.talpa.tv
- Gemeente Eindhoven – Municipality Eindhoven website:
  http://www.eindhoven.nl
- ASICS – a company specialised in the manufacturing of sportswear:
  http://www.asicseurope.com
  http://www.asics.nl and {.fr, .be, .co.uk, .de, .es, .dk, .fr, .it, .no, .se}