

Situational Requirements Engineering for the Development of Content Management System-based Web Applications

*Jurriaan Souer, Inge van de Weerd, Johan Versendaal and
Sjaak Brinkkemper*

Institute of Information and Computing Sciences, Utrecht University
Technical Report UU-CS-2005-050
www.cs.uu.nl

Situational Requirements Engineering for the Development of Content Management System-based Web Applications

Jurriaan Souer¹, Inge van de Weerd², Johan Versendaal² and Sjaak Brinkkemper²

¹ GX

Wijchenseweg 111, 6538 SW,
Nijmegen, The Netherlands
{jurriaan}@gx.nl
<http://www.gx.nl>

² Department of Information and Computing Sciences
Utrecht University

Utrecht, The Netherlands
{i.vandeweerd, j.versendaal, s.brinkkemper}@cs.uu.nl
<http://www.cs.uu.nl>

Abstract: Web applications are evolving towards strong content-centered Web applications. The development processes and implementation of these applications are unlike the development and implementation of traditional information systems. In this paper we propose WebEngineering Method; a method for developing content management system (CMS) based Web applications. Critical to a successful development of CMS-based Web applications, is the 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 Software Development 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 specialized in developing and implementing CMS-based Web applications. Future research focuses on optimizing the development method, and creating reference models and modeling tools for the development of CMS-based Web applications.

Keywords: Requirements Engineering, Web application, Content Management System, Method Engineering.

1 Web Application Development and Implementation

Nowadays, organizations rely on the Web to support their business processes and use Internet as a way create competitive advantage, global collaboration and integration with external partners [19]. Applications based on Web-technology are considered of strategic importance [28]. With the unrestrained growth of unstructured information, the need for a system to control the information emerges. Content Management System-based Web applications are applications which combine both enabling the Web technology and managing the unstructured information. CMS-based Web applications are therefore implemented to support the organization with the creation, management and publishing of information in an efficient en effective way. Examples of CMS-based Web applications are Mediasurface, Tridion, Vignette, Documentum, Microsoft Content Management Server, GX WebManager, and the open source product Zope. These Web applications should be agile enough to respond to the dynamic business and the ever-changing customer demands [1]. However, existing methods for the requirements engineering (i.e. the capturing and specification of the requirements) of Web applications often fail to capture and specify the business dynamics, fail to implement the desired system, and often lack the Web focus [19]. The literature gives a plausible explanation by suggesting that the Web applications differ from conventional information system [8], [9] and [18]. Since there is no literature on specific CMS, we developed a method specific for CMS-based Web applications, constructed

using components of two pre-existing methods: UML-based Web Engineering (UWE) [16] and the Unified Development Process (UP) [6]. The method is integrated into a generic development method consisting of the phases: Acquisition, Orientation, Definition, Design, Realization, 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 following research questions: what is a 'good' method to develop and implement CMS-based applications in different (customer) organizations.. Subsequently we pose the following questions. How are the requirements captured and specified in the route maps, and how is a certain method adapted to meet the organizational needs.

We start by providing an overview of the differences of developing Web applications and traditional information systems with their characteristics. In section 3 we give an overview of our method, starting with the construction. We validated 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 relationship and the differences, particular the characteristics that influence the requirements capturing process. We first provide an overview of the concepts whereupon we give a definition.

There are two developments in the last ten years, which gradually lead towards CMS-based Web-applications. The first development was the growing use and utilization of the World Wide Web. Information systems needed to be expanded beyond the boundaries of the organizations, resulting in Web information systems and Web applications. The other development was the unrestricted growth of digital content, which resulted in lack of information control and loss of data due to the large amount of digital content. New information systems were built to cope with the digital content. These information systems evolved over the years towards Enterprise Content Management Systems (ECM), which encompasses Digital Asset Management (DAM), Document Management (DM) and Web Content Management (WCM). The latter was developed specifically cope with Web content. WCM and Web applications were gradually merged into a single application, based on Web technology for the management of Web content. The figure below illustrates a categorization from Information Systems to CMS-based Web Applications.

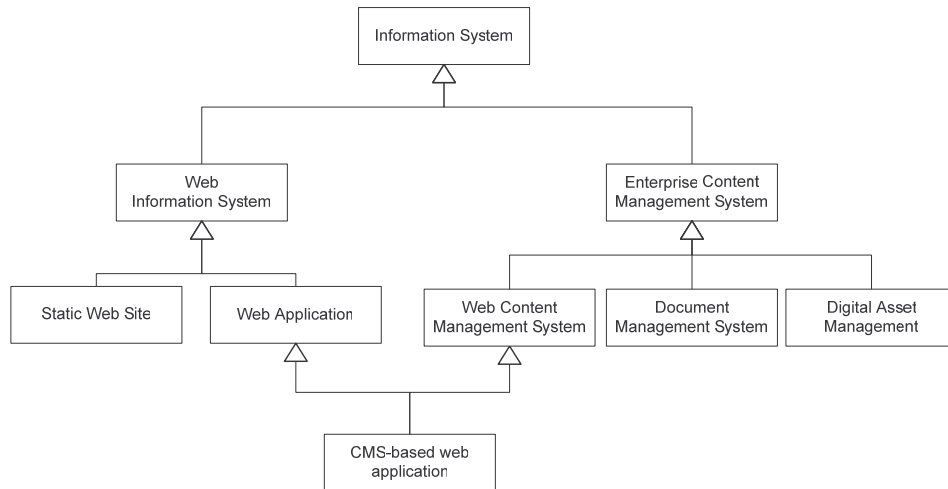


Fig. 1. Positioning of Web-based CMS-applications in the Information Systems categorization.

Defining a Web Information System is not unambiguous. There are several definitions of Web Information Systems according to Holck [14]. 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 Systems, which utilizes 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 does describe the four most often mentioned characteristics of Web information systems: (1) the new incremental development process; (2) the time pressure; (3) the new professions; and (4) a diverse and remote user group. Taylor et al found that *ad hoc* development of Websites seems to dominate in the industry [27] which suggests that conventional information system development methods are rarely used.

There are numerous Web Information Systems, not all of them are Web applications. In this paper, we use the following definition, derived from the definition of Gnaho [12]: a *Web application* is an Information System providing facilities to access complex data and interactive services via the Web and changes the state of business. Web applications implement business rules and have a certain functionality to either interact with another actor (person or system) or change its own state. De Troyer and Leune identified a comparable subdivision of Web information systems: a kiosk for presenting plane information (illustrated in figure 1 as a ‘Static Web Site’), and a Web application as a kind of interactive information system [29]. Castro et al [8] recognize 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 recognized challenge stems from the fact that Internet, and more specifically, the Web, is a completely different computing environment compared to conventional computer-based environments [1]. There are a few methods and tools available for developing Web applications such as WebML and W2000 as mentioned in section 5.

As the volume of digital content grows, Web applications evolve towards more content-centered Web applications. Vidgen et al. identified a list of issues as a result of the content growth: information consistency, navigational aspects, data duplication, content audit and control, tracking of content and mapping the website workflows on the business processes [32]. State-of-the-art content centered Web applications rely on a content management system to manage the information, which can be defined as a tool for the

creation, editing and management of information in an integral way [11]. 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 integrated workflow for structuring the process of creation and publication of information. With the inclusion of Web application in our definition, we implicitly state that CMS-based Web applications utilize the technology of the web and implement business logic.

Note that, as with IS development and implementation (Enterprise Resource Planning), there are standard modules (commercially) available for CMS-based applications, allowing organizations to implement and customize those according to the requirements. In many cases, the development of these applications does not start from scratch. Although we argued that CMS-based web applications justifies a specific development and implementation method, to our knowledge, no specific tools and methods exist for the development of CMS-based Web applications.

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 specifying the requirements, as described earlier. Therefore, a new method is needed for situational CMS-based Web applications. The Web Engineering Method (WEM) uses a development method based on five phases: orientation, definition, design, realization, and implementation.

3.1 The Method Engineering Process

To construct a new method, we used parts of existing methods to value 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 [7] and [30]. 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 UML-based Web Engineering (UWE).

The Unified Process 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 [6]. UP is a methodology, which is 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 Unified Process for the construction of our situational method for the development of Web applications.

The UWE methodology provides a systematic approach for the development of Web applications [16]. UWE is also based on the Unified Process and is therefore a good starting point for our development method. Some differences with the Unified Process are the specialization of the Unified Process for the development of web applications, the extension of the development cycle with a maintenance phase, the addition of two supporting workflows, project management and quality management, and extending quality control management with requirements validation and design verification in addition to only testing.

3.2 Project Categorizations

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

- *Standard projects*: projects which are mostly based on existing standard functionalities of CMS-based Web applications.
- *Complex projects*: projects based on existing functionalities with lots of customization or new build functionalities.
- *Migration projects*: Migration projects are 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 clear defined and arbitrary distinguished by their characteristics. In this paper, we focus on the standard and complex projects.

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 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 a customization is needed to create the required functionalities. The requirements capturing and specification is more challenging in complex projects. The table below 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. On the top side, 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.

	Standard	Complex
Acquisition	Acquire Customer Information Describe solution	Acquire Customer Information Feature list (UP) Describe solution
Orientation	Risk Management (UWE)	Risk Management (UWE)
Definition	Product vision (UP) User and domain modeling (UWE) Application model (UWE) Non-functional requirements (UP)	Product vision (UP) User and domain modeling (UWE) Use-case modeling (UP) Application model (UWE) Non-functional requirements (UP) Requirements Validation (UWE)
Design	Web application Architecture (UP)	Custom Architecture (UP) Technical design in detail (UP)
Realization	Implement Graphical design Configuration of Web application Functional testing	Implement Graphical design Configuration Web application & extensions Iterative development of custom functions in components (UP) Iterative development of interfaces (UP) Product quality assurance (UP) Functional and integration testing
Implementation	Production Deployment Acceptance	Staging deployment Production Deployment

		Acceptance
--	--	------------

Table 1. Elements of Standard and Complex Projects.

3.3 The Resulting Web Engineering Method

We refer to the method, resulted from the method engineering process as the Web Engineering Method, in short: WEM. In the following, we give an overview of the different phases of WEM. We focus on the first three phases (Acquisition, Orientation, Definition) since the Acquisition and the Orientation phases influences and the Definition phase exists of the requirements capturing and specification.

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. One of the earlier mentioned shortcomings of traditional implementation methods is the inability to specify the right requirements of CMS-based Web applications and the trouble mapping the application workflow with the business processes. To cope with this issue, we introduced a fragment of the Unified Processes 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 needs to be addressed by the Web application. CMS-based Web-applications provide an extensive set of standard functionality, therefore some candidate requirements matches or resemble available functionality. Based on the information acquired from the customer, the customer's wish is then reformulated in an unambiguous way, understandable for 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. An overview of the meta-modeling techniques used in the figure is elaborated in [30].

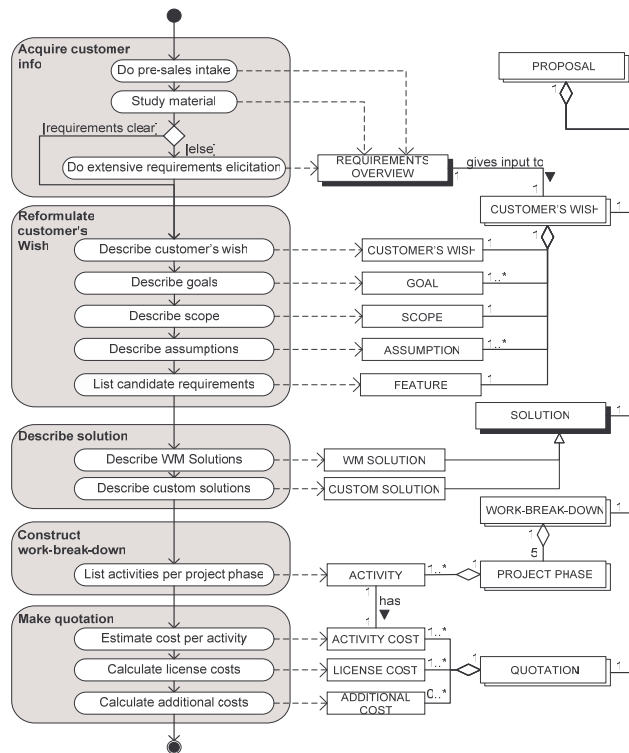


Fig. 2. Meta-model of the acquisition phase in WEM.

The left part of the figure illustrates the processes of the phase, the right part shows the deliverables, which are created in the corresponding process.

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

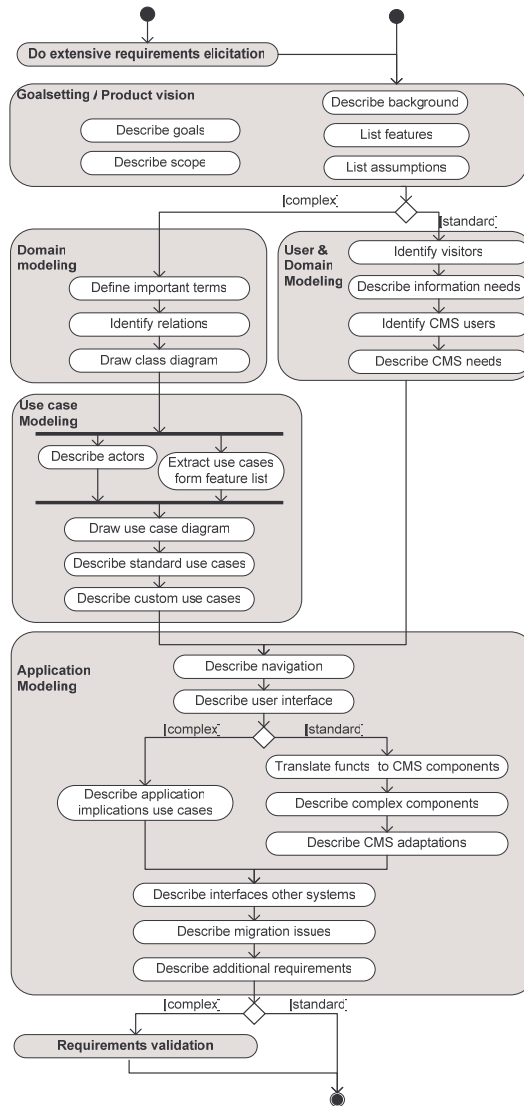


Fig. 3. Meta-model of Definition Phase in WEM.

Definition phase. The definition phase of WEM exists of the actual requirements analysis and specification. A meta-model of the definition phase is illustrated in Figure 4. Both the standard and the complex projects are integrated. For readability, we left out the data-side of the diagram.

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 use the product vision of 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 modeling from UWE to define the users and the environment. We identify the different types of users (e.g. visitors, registered visitor, editors, content managers, etc.) and their information need. Functions, derived from the users and their needs are compared to functions in the CMS-based Web application. This application modeling

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, workflow-modeling and a utilization of content reusability. Finally, the non-functional aspects are defined. Typical examples are user management, security, scalability, performance, design conditions, backup and logging.

For the complex components, we added domain modeling and use case modeling to specify the requirements of complex issues. Use case modeling originates from 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 is developed to show all the functions of the actors in one diagram. This gives a customer an overview of the Web application and minimizes the chance to implement the wrong requirements. In complex projects, the requirements are regularly discussed with the client to specify the requirements in the right way.

Design phase. During the design phase is determined how the requirements are realized. 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 customized components are developed, a custom-made architecture is necessary. Still, complex architectures utilize the standard Web application architecture. We use the 4+1 view architecture of UP, i.e. the logical view, process view, implementation view and the deployment view.

Realization phase. During the realization 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 realize the desired functions. All the realized components are eventually tested, based on the test plan. If the components succeed the functional and integration tests, the realization phase is concluded.

Implementation phase. CMS-based Web applications are generally deployed straight to production. Customers may demand a staging environment before actually deployed 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 small extensions or customizations. 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 the 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 modeling without a domain model is quite difficult to read. An important crosscheck for instance 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 for the system.

4 Validation

To validate the new developed method, we applied WEM in three projects at GX creative online development, a Web technology company in The Netherlands. GX develops and implements their Web-based CMS application *GX WebManager*, which is described in [5]. Figure 4 illustrates a high-level functional architecture of the GX WebManager.

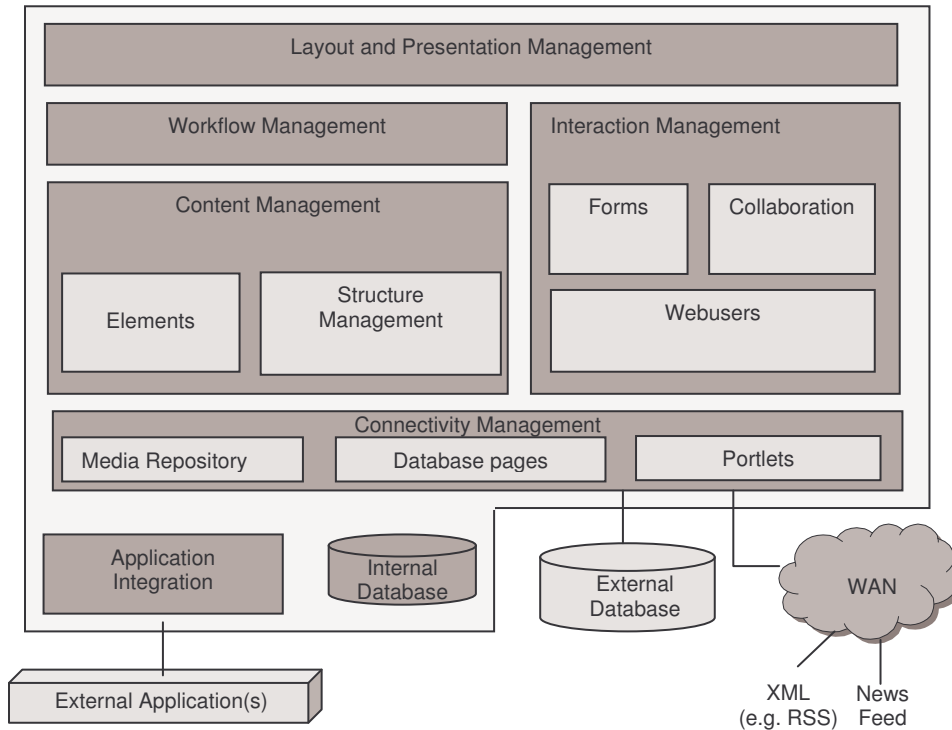


Fig. 4. High-level functional architecture of GX WebManager

In the validation, we solely focus on the definition phase, comprising of the requirements specification part of the overall implementation. We validated WEM by analyzing project aspects such as a realization 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 the 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, which we used in the validation, consist of:

1. A retail organization, which operates throughout Europe in the marketing, sales and distribution of home entertainment products.
2. A large telecommunication company.
3. A health insurance company (standard project, multiple sites)

Case	Project type	Est. Man-hours	Est. 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

Table 2. Overview of cases.

In Table 2, an overview of the three cases is with the project characterizations. The last two columns describe the number of Actors and Use cases in the requirements document. Note that case number three also has a few use cases, despite the characterization of a standard project. We will elaborate on one case. The other cases are discussed in the discussion (section 4.2).

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 multi-channel platform consisting of a web shop with physical and digital products and an in-store marketing application called narrowcasting. This multi-channel 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 modeling (terms, relations, class diagram);
- Use Case modeling (actors, use case diagram, use cases);
- Application modeling (navigation, interfaces, application implications, additional requirements); and
- Requirements validation.

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 of 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 interrelationship were identified to prevent miscommunications. These concepts and their relationships are modeled in a domain model. In Figure 5 is part of the domain model visualized.

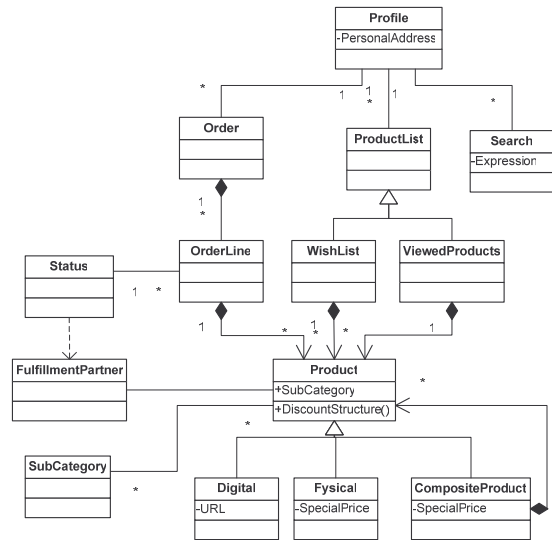


Fig. 5. Part of the domain model

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.

When the use cases were completed, the process continued with application modeling which translated use cases and non-functional requirements into the CMS-based Web application. Navigational design was developed based on the use cases. The use cases then needed to be translated to existing functions of the CMS-based Web application. With the application modeling, certain requirements were adapted to be in line with the existing functions (e.g. the customer wanted a ‘frequently asked questions’ with some specific features. The CMS-based Web application has a standard FAQ-function but exactly in accordance with the requirements. Yet, these choices saved time, effort and money for the customer).

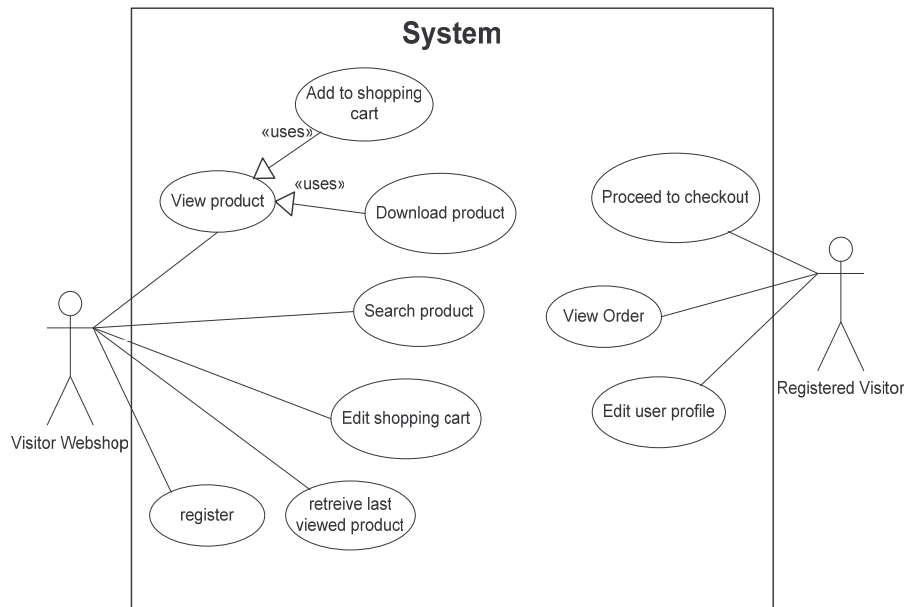


Fig. 6. Use Case Diagram

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, backend 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 workflow 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 of all the changes. A few additional requirements concerning financial transactions were specified and that concluded the requirements specification. To conclude the process, the requirements specification was validated with GX internally and with the customer before it was finalized.

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 system that was thereafter realized and implemented.

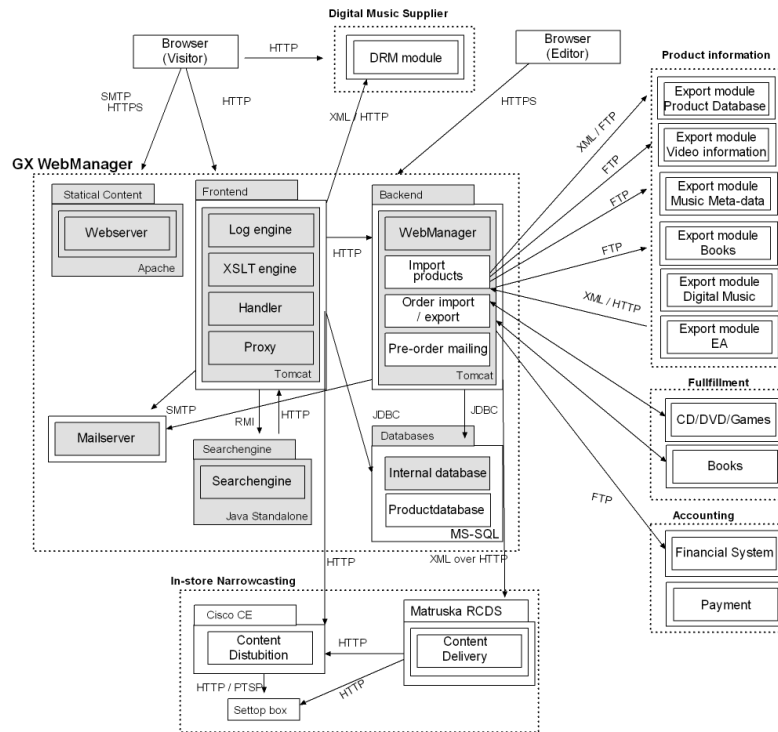


Fig. 7. Process view

4.2 Discussion

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

To start with; 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 was conform their specifications. GX was also very satisfied with WEM. WEM made a significant contribution toward realizing 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. Also, 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 modeling 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 form these cases. The first one is that not everyone can easily interpret use cases. 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 specification, and the 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 learned is when a standard functionality is used or is customized 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 instead of 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 chance of success.

5 Related Work

Several methods and techniques have been developed for designing and implementing web applications. One of these methods is the Website Design Method [29]. WSDM is a user-centred method for the design of kiosk Web Sites. A kiosk Web site 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 by using techniques like OMT or E-R modeling. De Troyer and Leune identified two types of Web sites: 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 the latter.

Sauer and Engels developed the Unified Modeling Language Extension for Modeling 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, Koch & Mandel [4]. They propose the UML Extension for Hypermedia Design, because the diagrams of UML are not sufficient to model aspects as navigational space and graphical representation. As they present an interesting modeling tool to visualize Hypermedia design, they don't provide us with a proper development method.

WebML is a notation for specifying complex Web sites at the conceptual level [9]. 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 existing notations as E-R modeling and UML. Also, WebML supports XML syntax, which can be used by software generators. WebML is 'a high-level specification language for designing data-intensive Web applications'. WebML can be considered a tool, WebML is a tool to design Web-applications, and it could be used in several development methods but is not a development method itself.

Finally, W2000 is a framework for designing Web applications based on the pre-existing assets UML and the Hypermedia Design Model. According to the authors, the integration between UML and HDM consists in four ways. Defining several stereotypes and customizations 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 related to both functional and navigational aspects [2].

6 Conclusion and Future Work

In this paper, we gave a definition and categorization of CMS-based Web applications. We described Web Engineering Method, 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, using components of the 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 optimize the three routes. This can be realized 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 which are relevant within the CMS-based Web applications. In addition, we are developing a modeling technique to configure these type 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. Like we developed three route maps for the different types of projects, likewise, different route maps can be developed for different types of clients. Mapping the client needs to product-extensions is one of the future work directions. This leads to new product extensions and architectures.

References

- [1] Arch-int, S. & Batanovv, D.N. (2003), 'Development of industrial information systems on the Web using business components', *Computers in Industry*, Vol. 50, Issue: 2 (February 2003) pp. 231 – 250.
- [2] Baresi, L., Garzotto, F. & Paolini, P. (2001), 'Extending UML for Modeling Web Applications', in *Proceedings of 34th Annual Hawaii International Conference on System Sciences (HICSS'34)*.
- [3] Barry, C. & Lang, M. (2003), 'A Comparison of `traditional` and multimedia systems development practices', *Information and Software Technology*, Vol. 45, Issue: 4 (March 2003), pp. 217 – 227.
- [4] Baumeister, H., Koch, N., Mandel L. (1999), 'Towards a UML extension for hypermedia design', In France, R., Rumpe, B. (eds.), *Proceedings UML '99, LNCS*, Vol. 1723, pp. 614 – 629.
- [5] Berkum, M. van, Brinkkemper, S. & Meyer, A. (2004), 'A Combined Runtime Environment and Web-Based Development Environment for Web Application Engineering', in *Advanced Information Systems Engineering: 16th International Conference, CAiSE 2004, Riga. Lecture Notes in Computer Science*, Vol. 3084 (2004), pp. 307 – 321.
- [6] Booch, G., Rumbaugh, J., Jacobson, I. (1999). *The Unified Software Development Process*. Redwood City, CA: Addison Wesley Longman Publishing Co., Inc.
- [7] Brinkkemper, S, Saeki, M. & Harmsen, F. 'Meta-modelling based assembly techniques for situational method engineering', *Information Systems*, Vol. 24, Issue: 3, (May 1999), pp. 209 – 228.
- [8] Castro, V. de, Marcos, E and Cáceres, P., 'A User Service Oriented Method to Model Web Information Systems'. *Proceedings of the 5th Conference on Web Information Systems Engineering (WISE 2004)*.

- [9] Ceri, S., Fraternali, P. & Bongio, A. (2000), 'Web Modeling Language (WebML): a modeling language for designing Web sites', *Computer Networks*, Vol. 33, Issue: 1 – 6 (June 2000).
- [10] Conallen, J. (1999). Modeling web application architectures with UML. *Communications of the ACM*, Vol. 42, No. 10, pp. 63 – 70.
- [11] Fernandez-Ilglesias, 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*.
- [12] Gnaho, C. (2001), 'Web-Based Information Systems Development – A User Centered Engineering Approach', *Lecture Notes in Computer Science*, Vol. 2016, pp. 105 – 118.
- [13] Gruhn, V., Mocker, M. & Schope, L. (2002), 'Experience: Development of an electronic commerce portal system using a specific software development process', *Proceedings of the Fortieth International Conference on Tools Pacific: Objects for internet, mobile and embedded applications*, Vol. 10, pp. 93 – 101.
- [14] Holck, J. (2002), '4 perspectives on Web Information Systems', in *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS '03)*.
- [15] Karlsson, F. & Agerfalk, P.J. (2004), 'Method configuration: adapting to situational characteristics while creating reusable assets', *Information and Software Technology*, Vol. 46, Issue: 9 (July 2004), pp. 619 – 633.
- [16] Koch, N., 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.
- [17] Lang, M & Fitzgerald, B (2005), 'Hypermedia Systems Development Practices: A Survey', *IEEE Software*, Vol. 22, No. 2 (March/April 2005), pp. 68 – 75.
- [18] Lee, C. Suh, W. & Lee, H. (2004), 'Implementing a community web site: a scenario-based methodology', *Information and Software Technology*, 46 (2004), pp. 17 – 33.
- [19] Lee, S.C. & Shirani, A.I. (2004), 'A component based methodology for Web application development', *Journal of systems and software*, Vol. 71, Issue: 1 – 2 (April 2004), pp: 177 – 187.
- [20] Liang, C-C. Hsu, P-Y. Leu, J-D. & Luh, H. (2005), 'An Effective Approach for Content Delivery in an Evolving Intranet Environment – A Case Study of the Largest Telecom Company in Taiwan', in *Proceedings of the 6th International Conference on Web Information Systems Engineering (WISE 2005)*.
- [21] Lowe, D. and Tongrungsrojana, R. (2004), 'Web Information Exchange Diagrams for UML', in *Proceedings of the 5th International Conference on Web Information Systems Engineering (WISE 2004)*.
- [22] Merialdo, P, Atzeni, P. & Mecca, G. (2003), 'Design and development of data-intensive web sites: The Areneus approach', *ACM Transactions on Internet Technology (TOIT)*, Vol. 3, Issue: 1 (February 2003), pp. 49 – 92.
- [23] Moro, R.G., Galante, R de M. & Heuser, A.C. (2004), 'Web mining and clustering: A version model for supporting adaptation of web pages', *Proceedings of the 6th annual ACM international workshop on Web information and data management (WIDM04)*, pp. 120 – 127.
- [24] Rahman, Z. (2004), 'Internet-based supply chain management: using the Internet to revolutionize your business', *International Journal of Information Management*, Vol 23, pp. 493 – 505.
- [25] Robbins, S. S. & Stylianou, A.C. (2003), 'Global corporate web sites: an empirical investigation of content and design', *Information & Management*, Vol. 40, Issue: 3 (January 2003), pp. 205 – 212.
- [26] Standing, C. (2002), 'Methodologies for developing Web applications', *Information and software technology*, Vol. 44, Issue: 3 (March, 2002), pp. 151 – 159.
- [27] Taylor, M.J., McWilliam, J., Forsyth, H. *et al* (2002), 'Methodologies and website development: a survey of practice', *Information and Software Technology*, Vol. 44, Issue: 6 (April 2002), pp. 381 – 391.
- [28] Turban, E., McLean, E. & Wetherbe, J. (1999), *Information Technology for Management*, 2nd edition, John Wiley and Sons: New York.
- [29] Troyer, O.M.F. de & Leune, C.J. (1998), 'WSDM: a user centered design method for Web sites', *Computer Networks and ISDN Systems*, Vol. 30, Issue: 1 – 7 (April 1998), pp. 85 – 94.
- [30] Weerd, I. van de (2005). 'WEM: A design method for CMS-based web applications'. *Technical report UU-CS-2005-043*, Institute of Computing and Information Sciences, Utrecht University.
- [31] Weerd, I. van de, Souer, J., Versendaal, J. and Brinkkemper, S (2005), 'Situational Requirements Engineering of Web Content Management Implementations', *Proceedings of the 1st International Workshop on Situational Requirements Engineering Processes (SREPO5)*, Paris, 13 – 30.
- [32] Vidgen, R., Goodwin, S. and Barnes, S (2001). Web Content Management. *Proceedings of the 14th International Electronic Commerce Conference*, Bled, pp. 465 – 480.

[33] Watcher, R.M. & Gupta, J.N.D. (1997), 'The establishment and management of corporate intranets', *International Journal of Information Management*, Vol 17, Issue: 6 (December 1997), pp. 393 – 404.

Appendix: Selection of Websites Realized 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 ISP in the Netherlands:
<http://www.planet.nl/>
- Daimler Chrysler Nederland – 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 specialized 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 }