Development of Content Management System-based Web Applications

SIKS Dissertation Series No. 2012-04
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Development of Content Management System – based Web Applications

Cover design: Esther Souer
Cover photo: Marie Pauline Batiller

ISBN: 978-90-393-5730-9

Development of Content Management System-based Web Applications

Ontwikkelen van op Content Management Systeem-gebaseerde Webapplicaties

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit Utrecht op gezag van
de rector magnificus, prof. dr. G.J. van der Zwaan,
ingevolge het besluit van het college
voor promoties in het openbaar te verdedigen op
28 februari 2012 des middags te 4.15 uur
door

Jurriaan Souer

geboren op 28 februari 1980
te De Bilt

Promotor:	Prof.dr. S. Brinkkemper
Co-promotor:	Dr.ir. R.W. Helms

Dit proefschrift werd mede mogelijk gemaakt met financiële steun van GX Software.

Acknowledgements

Many colleagues, friends, and family members have helped me to complete this dissertation. I would like to express my gratitude to these individuals for their support and assistance.

First, there are my colleagues at GX Software that gave me the opportunity to combine working on my dissertation with the dynamics of a fantastic software company: a perfect blend of scientific research and business. Arnoud, Bart, and Martijn, thank you for your trust, time and support along the way, and helping me to accomplish my ambitions at a beautiful company. Rene, Hans, and EJ, thank you for your flexibility, enthusiasm, and understanding as I took my time to work on my research.

I am grateful too for the support and advise from my friends and colleagues at Utrecht University. I need to thank especially Sjaak Brinkkemper for his guidance, his patience and his encouragement throughout the development of the project. He not only helped me realize my scientific ambition and allowed me to have as much freedom as I needed to accomplish it, but he also introduced me to GX Software in the first place and organized my research project with Arnoud. Remko Helms and Johan Versendaal, thanks for your daily supervision, the good conversations we had and your many contributions to this work. I thank the other members of the research group as well for the research environment and support. Thanks also to my colleagues, friends and Master's students I supervised that provided valuable input: Inge, Paul, Lútzen, Thijs U., Thijs K., Dirk-jan, Martin, Bram, Slinger, Matthew, and Vincent.

My time was made enjoyable in large part due to the many friends that became a part of my life. I am grateful for the trips, drinks, games, sushi, Kenpo, a place to sleep once in a while, and some memorable moments at the 'Vierdaagsefeesten'. Marc en Gerd-Jan, thanks for being great friends, for some good discussions, whiskey festivals and barbecues. Without question I wanted you guys to join me at my defense.

Last but not the least, I deeply thank all my family. My brothers, sisters and their families for all the beautiful moments. I thank my parents, and parents-in law especially for their faith in me, and allowing me to be as ambitious as I wanted – and taking great care of my children while I was accomplishing it. But most of all I want to thank Ilse, the one going through all the highs and lows of a researcher, for her love, strength, sacrifice, and consistent support. She helped me make all of my efforts possible, moved with me to California and allowed me to pursue my dreams. And of course there are our precious little dragons, my sons Ian and Elias: thank you for inspiring and amazing me every single day.

Contents

Part I

Introduction to this dissertation

Chapter 1

Introduction

This dissertation is about developing Content Management System-based web applications. Using this term, we refer to web applications that utilize a Content Management System (CMS) as its main development, authoring and delivery platform. Most web applications utilize a Content Management System, at least to a certain extent, to allow organizations to *manage* the *content* of the application with a software *system*. Any content-intensive site that requires regular maintenance, for example to update the website with news or updating product information benefits from a CMS because it allows the non-technical users that are responsible for the content to do so themselves without the help of a development team or webmaster. CMS-based web applications are found everywhere, in all sorts of applications and in every industry: e-commerce sites, bank and insurance sites, corporate portals, intranets, extranets, communities, news portals, to name just a few.

The concept of Content Management originates in pioneering efforts of

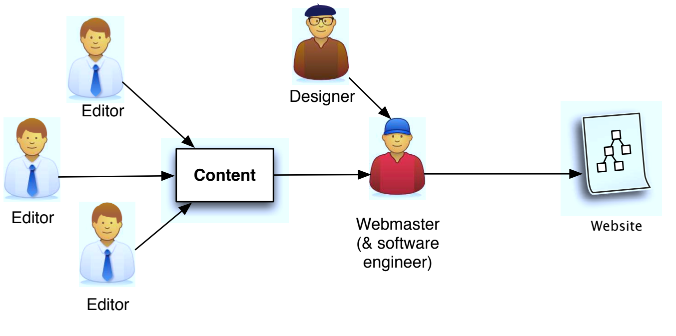

Figure 1.1: Publication process without a WCMS

the mid-nineties to manage corporate web content with in-house systems [Boiko, 2001]. The main problem back then was that anything on the website was created, managed and updated by the web master as illustrated in Figure 1.1 [Jablonski and Meiler, 2002]. A solution came in the form of Content Management, which was a function of web applications that allowed non-technical users to create, edit and delete content on the dynamic parts of the website which removed the 'web master bottleneck'. Over the years, Content Management Systems have evolved into robust development platforms that provide a large set of capabilities out of the box, for example authoring tools, administration tools, document management, authorization and workflow, personalization, multi-lingual publishing, and multi-channel publishing that combine websites, websites for mobile devices, RSS feeds, documents, Facebook, Twitter, and YouTube) and several integrations with back-office systems [Grossniklaus and Norrie, 2002]. The idea and solutions reach beyond plain management of content. The diffusion of content structuring and interchange standards, particularly those based on eXtended Markup Language (XML), and emerging technologies for integrating web pages and multiple access devices with organizational databases and applications [Morrison et al., 2002], makes web engineering a challenging task.

Content Management Systems provide development teams with a standardized platform that they can use to efficiently develop CMS-based web applications. From a deployment point of view, a Content Management System could be the single development, authoring and delivery platform, or it could be part of a larger enterprise architecture where other applications extract content or functionalities from the Content Management System and reuse it in a separate web application. Developing CMS-based web applications is a complex, multidisciplinary process and requires a variety of disciplines that are necessary to address the different areas of the development process. However, an important consequence of using a Content Management System is that you have to choose for a standardized product of platform, also known as a software product. A software product is defined as *a packaged configuration of software components or a software-based service, with auxiliary materials, which is released for and traded in a specific market* [Xu and Brinkkemper, 2007].

This means that the data structures for storing content, the conception of interfaces, content management activities and the basic architecture are defined by the characteristics of the CMS. For example, Päivärinta and Munkvold Paivarinta defined that the core of any content management driven solution resides in understanding of the content itself and its role in the organizational context. This is known as the content model and consists of *(1)* content structure, view, and presentation models, *(2)* content lifecycles, *(3)* metadata, and *(4)* corporate

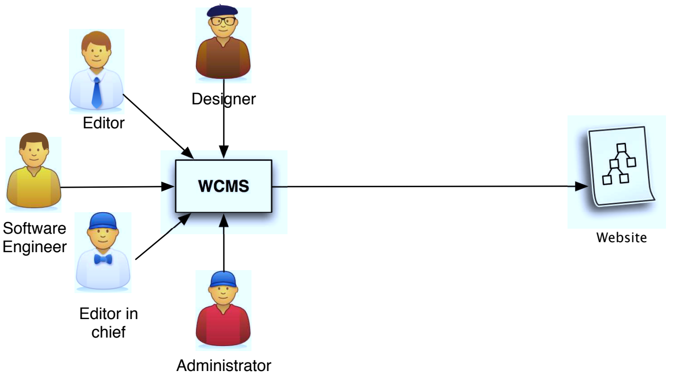

Figure 1.2: Publication process with a WCMS

taxonomy. These aspects of the content model are fundamental aspects of a Content Management System and in order to avoid high customization costs, high maintenance and support costs and future upgrade problems, organizations try to limit customizations or to isolate them if possible. Therefore, developing CMS-based web applications requires a thorough understanding of the CMS that is used for development because during every step of the development cycle, the product experts have to decide what the best way is to cope with a certain problem within that CMS. If not managed correctly, organizations may face high development costs, and the inability to meet business requirements and project deliverables. Depending on the organization and the strategic importance of the web application, the consequences can be huge.

In this dissertation, we will refer to the process of developing CMS-based web applications as CMS-based web engineering. Kappel et al. [Kappel et al., 2006] defined web engineering as *the application of systematic and quantifiable approaches (concepts, methods, techniques, tools) to cost-effective requirements analyses, design, implementation, testing, operation, and maintenance of high quality web applications*. We use this definition to structure our research. The aim of this research is to design the web engineering processes to do the requirements analyses, design, implementation, testing, operation, and the maintenance of high quality web applications, using a Web Content Management System as its main development, authoring and delivery platform. From a process point of view, implementing Content Management Processes requires a broader scope than just a technology-focused solution, which is what Content Management Systems in essence are. Organizational structures, people and processes are key to the

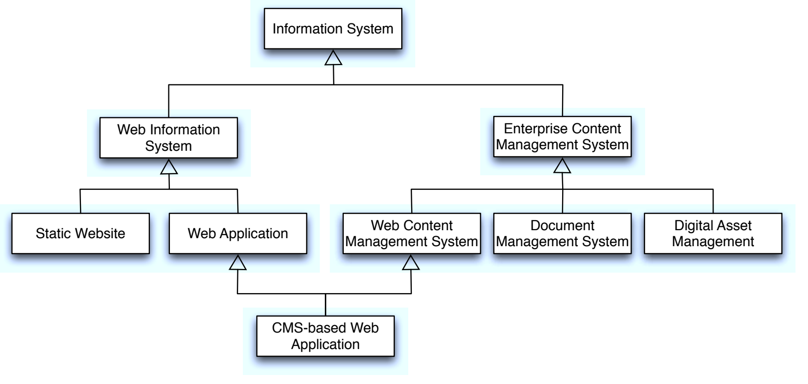

Figure 1.3: Positioning of CMS-based web applications

successful adoption of content management processes. Surprisingly, no literature can be found on the development processes of developing Content Management System based web applications.

By looking at the way existing web engineering processes are defined and structured, we realized that the engineering principles and methods do not provide all the necessary characteristics to cope with Content Management System-based web engineering. For example, designers often construct web applications based on best practices and methods they have learned from experience in other domains or software projects. Such practices work well when designing customizations for completely new components, however, they do not address designing applications based on product software and Content Management Software in particular, where you have to take into account both web-related specifics such as personalization, a hypertext front-end, caching; and the Content Management Software specific characteristics such as the product architecture, existing modules, that is, the way the personalization, hypertext front-end and caching is implemented and configured by the product.

This dissertation proposes the Web Engineering Method (WEM), an integral method for CMS-based web engineering. We provide a process description and deliverables for every step in web engineering, that is, requirements engineering, design, realization, implementation, operation and maintenance. We build upon existing web engineering theories and practices in order to learn what aspects are relevant in specific situations and then we combine them into a single method using Situational Method Engineering processes as defined in [Brinkkemper, 1996]. Moreover, we provide two methods to innovate our Web Engineering Method based on a model-driven paradigm, and a product family approach.

The main objective is to provide researchers and practitioners a complete method for developing CMS-based web applications. A second goal is to provide strategies and guidelines to innovate the web engineering processes.

1.1 Contribution

The main contribution of this dissertation is the Web Engineering Method (WEM). WEM consists of concepts, notations, process descriptions and techniques for the development and implementation of Content Management System-based web applications, which can be used by both researchers and practitioners. We combine fragments from known web engineering methods with new practices identified at a software company. However, the main contribution consists not of the individual aspects of the approach, but a systematic method for developing web applications using a CMS as the main development platform, including processes, tasks, and deliverables. Furthermore, existing courses and literature that are focused on developing web applications from scratch can use this dissertation to adjust their materials to be more inline with the best practices for building web applications based on components and Content Management Systems.

1.1.1 Scientific Relevance

Web engineering is a well-established and broad area of research regarding topics concerning technologies, methodologies, tools, and techniques used to develop and maintain web-based applications and to enable and improve the broadcast and use of web-related content. Research in the area of Content Management Systems is still relatively limited, although the last years have shown increased interest in Web Content Management. Well-respected conferences in the web engineering community, including the International World Wide Web Conferences (WWW) and the International Conference on Web Engineering (ICWE) introduced Web Content Management as a research topic at their conferences. Both conferences added Web Content Management as one of the topics in their Call for Papers, starting in 2007. It is therefore interesting to note that the number of publications on (Web) Content Management is very limited. One of the earlier publications on Content Management is by Hysell in [Hysell, 1998] where web content management is mentioned as a requirement for their 'next-generation website'. In year 2000, some relevant articles were published such as [Arlitt et al., 2000], [Schuster and Wilhelm, 2000] and also the industry came with publications like [Gilbane, 2000]. What's more, web engineering as a research topic became more visible with publications by [Koch, 1999], and [Ceri et al., 2000]. They all address certain parts of the web engineering process or details on Content Management Systems. At the same time, there were some publications addressing the fact that web engineering requires a new approach to software development, such as [Lowe, 1999], [Murugesan et al., 2001], and later [Kappel et al., 2006]. These approaches do address web engineering specific challenges, but they do not address the development approach using a Content Management System as the foundation.

The distinguishing focus of this research is the emphasis on a Content Management System as the main development platform. We demonstrate that a CMS provides a solid foundation for web engineering that copes with the most identified web engineering challenges. This dissertation contributes to the scientific knowledge by studying web engineering processes from a content management perspective. We identify Content Management System specific challenges, provide new methods that combine existing methods with new insights, and show how a model-driven approach and a product family engineering approach can be applied in the context of Content Management System-based web engineering.

1.1.2 Practical Relevance

Organizations are aware that they can use product software to develop web applications in an efficient way. A Content Management System is an often-used solution for developing web applications since it allows non-technical users to update the website and it provides developers with a standardized development platform. However, there are still many questions that need to be answered regarding how to develop, implement and maintain a Content Management System. The main contribution is a detailed process description of the implementation process of CMS-based web applications. We provide two routes each with a set of unique processes, depending on which route you take.

The insights obtained in this dissertation provides practitioners with knowledge in the form of methods, processes, and deliverables to develop and implement Content Management System-based web applications and to give them guidance in shaping their web engineering approach. We provide detailed descriptions on how to structure requirements management, design and development and implementation processes. Moreover, we provide developers of Content Management Systems with insights into how they can innovate their Content Management System. First of all, we demonstrate how a model driven approach can be applied to configure a Content Management System. A second way to innovate the Content Management System is based on a product family engineering approach where the developers are provided with processes and methods in order to identify commonalities in their implementations. The result of that analysis can be a pre-configured CMS for a specific market (an ‘Industry Template’).

1.2 Research Area

1.2.1 Web Engineering

Web engineering is an established research topic that has been around since 1998 [Ginige and Murugesan, 2001b]. There are many publications on web engineering. Below we mention a few research initiatives that served as inspiration, resources for theories and practices for web engineering.

The Object-Oriented Hypermedia Design Method (OOHDM) as proposed by Schwabe and Rossi [Schwabe and Rossi, 1995] [Schwabe et al., 1996] is comprised of four steps: conceptual modeling, navigational design, abstract interface design and implementation. These activities are performed in a mix of incremental, iterative and prototype-like ways. In OOHDM a clear separation is made between navigation on the one hand and on the other hand presentation. OOHDM has introduced navigational concepts for contexts, personalization and user interaction.

The UML-based Web Engineering (UWE) approach is described in [Koch and Kraus, 2002], [Koch, 1999], and [Baumeister et al., 1999]. UWE is an object-oriented, iterative and incremental approach for the development of web applications. The development process of UWE consists of five phases: inception, elaboration, construction, transition and maintenance.

The Business Informatics Group (BIG), headed by Kappel [Kappel et al., 2006]. The research areas of BIG cover model-driven engineering, data engineering, process engineering, web engineering, and services engineering. Kappel published a book on web engineering which we reference throughout this dissertation

The Object-Oriented Web-Solutions Modeling approach [Pastor et al., 2001], [Pastor et al., 2006]. OOWS provides mechanisms to deal with the development of hypermedia information systems and e-commerce applications in web environments. OOWS strongly focuses on the generation of the required web Application and less on managing the content and the application afterwards. OOWS extends OO-method by means of a navigation model, a presentation model and a user categorization. OOWS is comprised of two main activities: system specification and solution development.

The Web Information Systems (WIS) research group by Houben [Houben et al., 2008]. The Web Information System group is focused on engineering and the science of the adaptive web, including semantics-based adaptation, personalization, and user modeling. Among others, they developed Hera, a methodology that supports the design and engineering of web information systems. It is a model-driven methodology that distinguishes three parts in the design: integration, data retrieval, and presentation generation.

The Web Modeling Language approach in [Ceri et al., 2003], [Ceri et al., 1999], [Ceri et al., 2000]. WebML is a notation for defining complex websites at a conceptual level. In line with the definition of web engineering, the WebML approach consists of seven phases: requirements specification, data design, hypertext design, architecture design, implementation, testing and evaluation and maintenance and evolution. The WebML method is supported by a commercial model-driven development environment called WebRatio that allows modeling and the automatic generation of web interface applications.

The Web and Information Systems Engineering (WISE) Laboratory research group by De Troyer [de Troyer, 2008]. The activities of WISE concentrate on innovative information systems, such as next generation web applications. The WISE research group is focused on new engineering methodologies, tools and software frameworks for the rapid prototyping and efficient realization of innovative information environments.

Some of the mentioned web engineering methods address Content Management as part of their approach. For example, WebML addresses the Content Management Model [Ceri et al., 2003]. However, they address the content management function: allowing content creation, update and delete operations. That by itself does not provide much explanation of how to develop CMS-based web applications. Another limitation in the existing literature is that most web engineering models are based on developing web applications from the ground up.

1.2.2 Web Content Management

According to Rothfuss and Ried, Content Management is '*the systematic and structured procurement, creation, preparation, management, presentation, processing, publication and reuse of content*' [Rothfuss and Ried, 2002]. A similar definition is defined by Boiko in [Boiko, 2001]: *the overall process for collecting, managing, and publishing content to any outlet* [Boiko, 2001].

- In collection: You either create or acquire information from an existing source. Depending on the source, you may or may not need to convert the information to a master format (such as XML). Finally, you aggregate the information into your system by editing it, segmenting it into chunks (or components), and adding appropriate metadata.

- In management: You create a repository that consists of database records and/or files containing content components and administrative data (data on the system's users, for example).

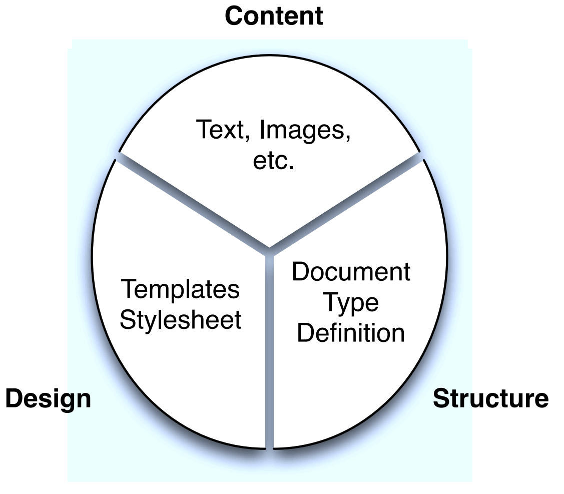

Figure 1.4: Separation of Content, Structure and Design

- In publishing: You make the content available by extracting components out of the repository and constructing targeted publications such as websites, printable documents, and e-mail newsletters. The publications consist of appropriately arranged components, functionality, standard surrounding information, and navigation

This can be adjusted to the contents of web applications such as text, images, flash, movies, etc., represented in a web enabled format (e.g. XHTML, RSS, XML). In this case it would become *Web Content Management* [Jablonski and Meiler, 2002]. The systems to support these processes are called *Web Content Management Systems*. One of the main principles of Web Content Management Systems is the separation of *content*, *structure* and *design* as illustrated in Figure 1.4 [Schuster and Wilhelm, 2000]. That way, the content can by managed independently where it is rendered (structure) and in what template (design).

A similar set of processes on on web content management is identified by Nakano [Nakano, 2002]. Nakano describes two concepts for applying web content management: collaboration operations and work cycle development. Collaboration operations include the following five operations: *Submit* – when you move web assets from a work area to a staging area; *Compare* - identifies whether the assets that were submitted were new, modified or deleted assets to the staging area;

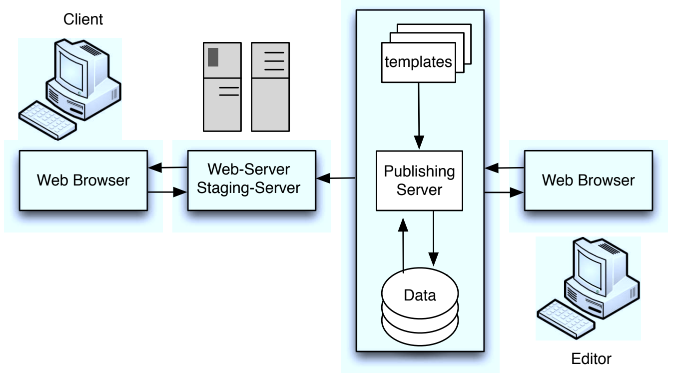

Figure 1.5: Example of a Content Management System

Update copies new, modified or deleted assets from the staging area to the given work area; *Merge* resolves conflicts between the work area and staging area; and *Publish* is a snapshot of the staging area.

An example of a how a Content Management System works is detailed by Schuster and Wilhelm [Schuster and Wilhelm, 2000] and illustrated in Figure 1.5. Editors use a browser to access the publishing server where they create the content. the content is stored in the data repository. In this example, the publishing server assembles static HTML using the content, the templates and the additional files they use such as cascading style sheets, javascript, and images. This static HTML is then 'pushed' towards the web server (or staging server) where it is served to any client visiting the site. This is one of the two most popular delivery mechanisms. The other mechanism renders a page on the visitors requests instead pre-building the page. For any realistic degree of personalization, for interaction with data systems (transactions, catalogs), or for dynamic updates (such as changing news stories or stock quotes), you need the processing power on the web server that a dynamic system gives you. A CMS-based web application is rarely 100% static or 100% dynamic.

The need for content management is also mentioned by Lowe [Lowe and Henderson-Sellers, 2001]: '*The actual authoring of the content itself is also a significant development issue that is often overlooked*.' Furthermore, '*With web development, the generation of* data *(i.e. content authoring) is fundamentally part of the development process*'. We need to ensure that content authoring is appropriately integrated into the development process.

The financial impact of Content Management Systems reaches beyond the few cases we used in this research: IT Research and advisory firms predict the software market for (commercial) Web Content Management will exceed 1 billion dollars in 2010 and will have an expected annual growth of 14% from 2009 to 2014 [Gartner, 2010]. And that is just in software licenses alone. If you consider that implementation services are three times the licensing on average and then add the services for free open source licensed Content Management Software, one could imagine that a small improvement of the development process has a huge potential saving.

Kappel et al suggests that (Web) content management is a buzz-term not used consistently by the industry with reference to [Gilbane, 2000]. We have to agree that the term is not used consistently by the industry – moreover, in this dissertation we use the terms ‘Content Management System’ (CMS), and ‘Web Content Management System’ (WCMS) interchangeably. I intend to make clear in this dissertation that Web Content Management is more than just a ‘buzz-word’ and that it has evolved from a small part of the web engineering process (the part where non-technical staff to manage the content of a website) into an important part of the modern day web engineering processes. Furthermore, it can be used as the main development platform for modern web applications and that it thereby becomes a driver of new web engineering processes that require a different approach.

1.2.3 GX Software’s Web Content Management System

This section gives a short introduction into the CMS that is developed by the organization where most of our research took place: GX Software[1]

The Content Management System used in this in this research is called ‘GX WebManager’ and provides web content management and online functionalities such as personalization, application integration, multi-channel publishing, search engine optimization (SEO), digital asset management, online transaction processing, and WebTV for streaming media. Most of the web application functionality built on the content management platform is configured without customization. However, online marketers always want some new functionalities (e.g. Web 2.0, Wiki) and therefore CMS’s are often customized to meet the business requirements. Customization is one of the most troublesome processes within WCM implementations. Customization can lead to higher implementation

[1] This work is based on our short-paper as published in the Proceedings of the 8th International Conference on Web Engineering (ICWE), entitled ‘A Component Based Architecture for Web Content Management: Runtime Deployable WebManager Component Bundles’ in 2008 [Souer and van Mierloo, 2008]. The work is co-authored with Martin van Mierloo.

costs, longer implementation time, and could give problems when upgrading to a more recent version of the software.

Confronted with these challenges with the previous CMS system – as described in [van Berkum et al., 2004] – we developed the design goals for a new CMS platform:

- Ease of Development
- Ease of Maintenance
- Ease of Deployment

We will elaborate on these design goals below.

Ease of Development

Developing custom functionalities should be easy for developers. Ease of development includes good debugging support, minimize the effort for adding new functionalities, use standards for development (i.c. Java, JSP), out-of-the-box support from software development kit Eclipse + Webtools, Java and Tomcat, and it should be easy to develop in teams within larger projects. To conclude, it should be easy to change the graphical user interface of the CMS based on standards.

Ease of Maintenance

A lot of organizations rely on their web applications for a large part of their business. Downtime for deployment, upgrades and failure should be minimized. Ease of maintenance consists of good support to analyze bottlenecks and to see if the issue (e.g. bug or performance) is part of the CMS platform or the custom developed functionality. It should be possible to change, uninstall or upgrade functionalities without having to create a new deployment.

Ease of Deployment

Still, there are situations where new deployments are needed. However, these deployments should be as easy as possible to minimize the downtime and to allow different hosting providers to manage the deployments themselves. Moreover, easy of deployment encompasses easy deployment over a complete Development, Test, Acceptance and Production environment (DTAP support): when a custom functionality is developed in a development environment it should be very easy to install it in the Test environment without any configuration.

Implementing the Design Goals

To implement the design goals we introduced a new concept called a *WebManager Component Bundle* (or WCB): a plug-in system (or Component) for adding new functionality at runtime to the product. WCBs are single (JAR)-files that can be easily exchanged and deployed on different CMS deploys. WCBs are developed using common Java development tools such as Eclipse and Maven. Developers can use archetypes to create various types of WCBs with a single command. This, in combination with a set of tutorials, videos, forums and a wiki on online development community makes it easy to get started.

WCBs incorporate three open technologies which allows for ease of development, ease of maintenance and easy of deployment:

- **OSGI**: The open standard OSGi [2], with the implementation Apache Felix [3]
- **JCR**: The open standard JCR or Java Content Repository (JSR-170/JSR-283[4]), with the implementation Apache Jackrabbit [5].
- **Spring MVC**: the Model View Controller framework of the Spring Application Framework [6]

The following sections elaborate on these three technologies.

Components

OSGi technology provides a service-oriented, component-based environment for developers and offers standardized ways to manage the software lifecycle. OSGi is a well-known technology in the mobile industry, but is rarely used as a server side technology. OSGi is used within the CMS to manage the lifecycle of the WCBs, it allows starting, updating, stoping and uninstalling WCBs at runtime. The OSGi guarantees the stability (it stops a WCB when its corrupt and makes sure that the platform is not influenced by the WCB). OSGi also includes version control that allows for automatic updates when a new version of the custom functionality is available. With OSGi, developers can easily develop their WCB locally and activate it on the Test environment using OSGi. The runtime deployment aspect and the stability contribute to the ease of deployment and ease of maintenance.

Content Repository

The Content Repository for Java Technology specification, developed under the Java Community Process as JSR–170 (currently being enhanced in its successor JSR–283) separates infrastructural services from application services. This unified API allows access to any compliant repository in a vendor- or implementation-neutral manner, which will prevent possible vendor lock-in. The JCR API is a

[2] http://www.osgi.org
[3] http://felix.apache.org/
[4] http://www.jcp.org/en/jsr/detail?id=283
[5] http://jackrabbit.apache.org/
[6] http://www.springframework.org/

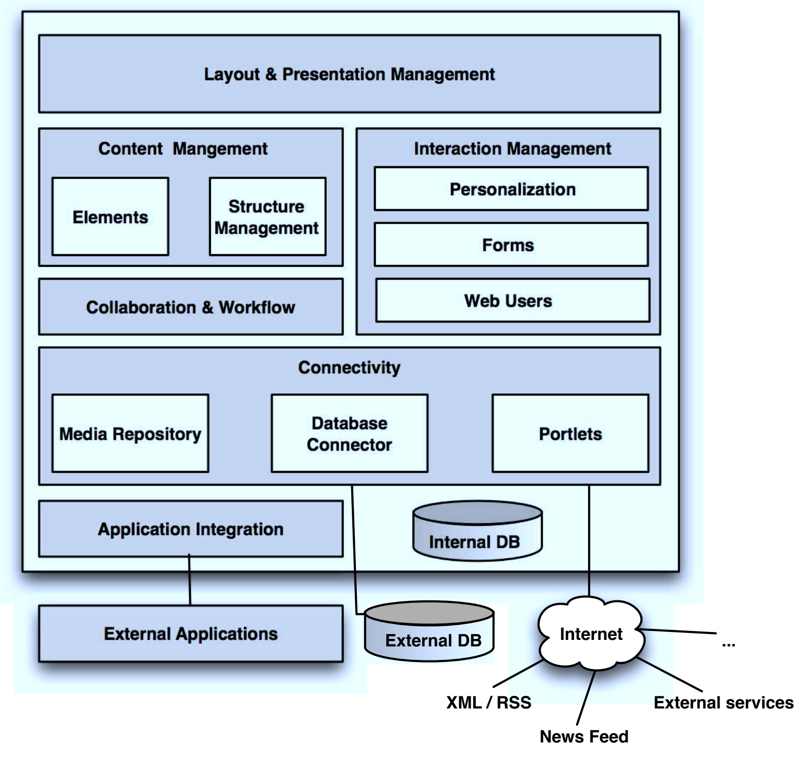

Figure 1.6: High-level Functional Architecture of GX WebManager

type of Object Database that stores, searches and retrieves hierarchical data and allows true content repository infrastructure so that different applications can use the same interface for many purposes, making it universally accessible. A major advantage of JSR-170 is that it is not tied to any specific architecture. The back-end data storage for a JSR-170 implementation could be a file system, a WebDAV repository, an XML- backed system, or even an SQL database.

Within the CMS, the JCR is used as an persistence layer, including the data storage for the WCBs. Developers will be able to avoid the effort associated with learning the particular API of each repository vendor. Instead, programmers will be able to develop content-based application logic independently of the underlying repository architecture or physical storage. Using a common interface reduces both times and risk, in so much as a company will no longer need to rely on any one proprietary repository.

Model View Controller MVC (Model-View-Controller) is a design pattern that helps to separate user interface logic from business logic. Usage of a MVC based library offers a good approach to render the edit interface of the CMS in a way that is supported by many open source standards. The current proprietary scripting language can contain both business logic and user interface logic in one and the same script. For the implementation we use the open source Spring MVC Framework. The inclusion of Spring MVC is the implementation of the design goal that developers should be able to easily change the user interface of the CMS.

In summary, the platform is separated from its functional components in allowing developers to easy create new functionalities and deploy and maintain them at runtime. Modifying the platform implies checking if the other functionalities of the platform are still intact. Making changes to a component however, implies only checking if that specific component still behaves as expected. As a result, the engine of the CMS will have a higher stability and will have a non-changing interface with the components. Only individual WCBs can be modified and therefore be corrupted. This will have no impact on the complete website in general. In this demonstration we will show how to build and deploy a WCB based on an archetype and elaborate on the various forms of applications.

Combining the three technologies

This combination of three open standards and technologies results in a number of interesting characteristics. From a technical point of view the three technologies provides Lifecycle management, Persistence, Controller framework, Runtime deployment, Dependency management, Headless Services, View and edit presentation, OTAP support and merge conflict resolution with UUIDs, and Schema evolution.

But also the different roles in organizations that deal with web content management can benefit in various ways from WCBs. The business owner or marketing can quickly launch new functionality and be innovative. They are no longer dependent on the supplier's product roadmap and resources. When a WCB is already available they are even not dependent on developers. The IT department take advantage from the fact that managing a web content management system is easier and costs are lower since the platform is standard and additional functionality is added as standards based bundles, which have a predictable and protected behavior. In DTAP environments functionality and content can easily be transferred to other servers for testing purposes. And also developers have advantages: working with WCBs is easy to learn because of the use of common tooling (e.g. Eclipse, Maven, Spring MVC, JSPs) and because there is an active development community. Development of WCBs is aimed at working in teams and reusing components and code.

Figure 1.7 shows the application architecture of the CMS. It consists of five layers: the user interface layer, web layer, service layer, domain layer and the data access layer. As illustrated, the JCR is used as the persistency storage. The OSGi is part of the service layer and Spring MVC is used in the web layer as the MVC framework. WCBs can access all layers of the application and can be used for creating user interface (graphical design of a website), the web layer (for functions and panels), the service layer (for services and components), the domain layer (to access platform functions and other WCBs) and the data access layer. Each WCB has its own version number and certification that is centrally managed.

1.3 Research Description

1.3.1 Research Questions

Web engineering is an existing research area with several groups and methods. However, none of them specifically address Web Content Management Systems as a foundation for web engineering. For reasons we described in the previous sections, we argue that a Web Content Management System is a are very suitable platform for web engineering. An overview of the web engineering process is illustrated in Figure 1.8 where we show the different phases acquisition, orientation, definition, design, realization, implementation and maintenance). The main development phase starts with the orientation phase and stops after the implementation where it will move to the operations and maintenance phase. Besides the processes, we provided an overview of its main activities such as preparation, requirements analysis, design, development and implementation before it goes into operation and maintenance. Our research questions are structured to define these processes with respect to Content Management Systems.

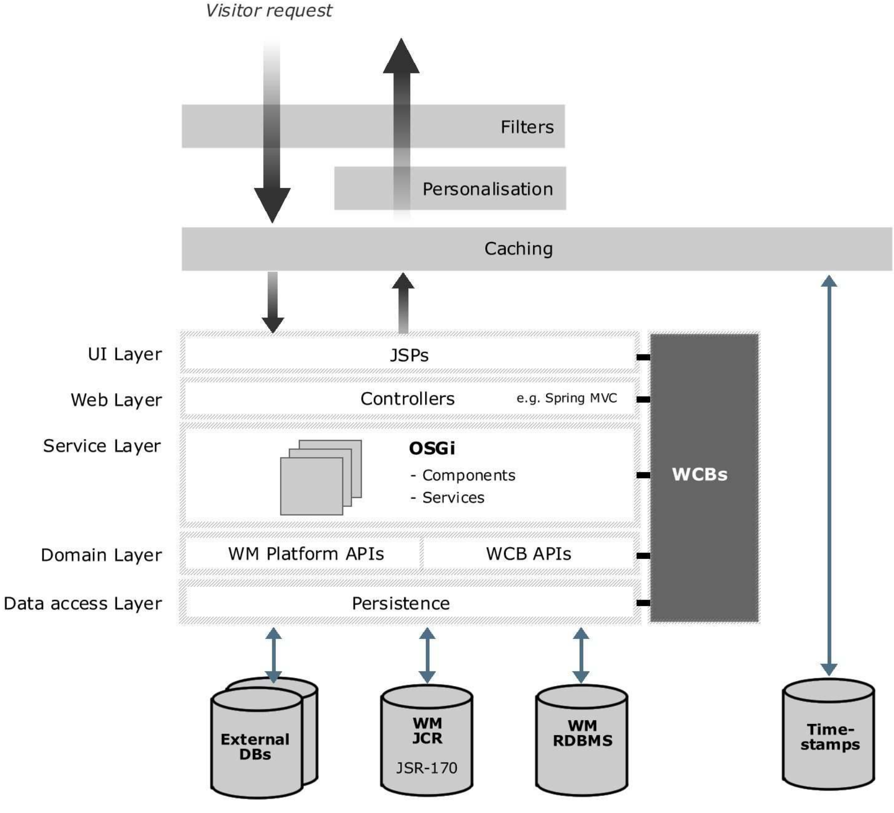

Figure 1.7: Application Architecture

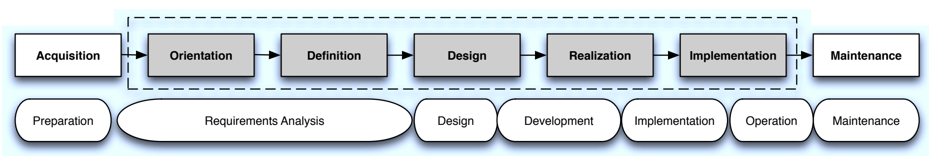

Figure 1.8: Web Engineering Method process overview

Therefore, the first research question is:

RQ1: What are the concepts and processes of web engineering using a web content management system as a foundation?

Developing web applications can be a complex process, especially now that they are becoming full-blown information systems with more often than not tight integrations into back-office systems such as ERP, CRM, and Workflow systems. For the reasons defined in the introduction, modern web applications often utilize a content management system. Before this research, there was no integrated view on how to develop and implement Content Management System-based web applications. Well known web engineering methods such as WebML, UWE, OOWS, and OOHDM only peripherally address the content management and usually as a single function of a web application that allows users to manage content on a website.

In this research we use the definition by Kappel et al. to structure our research method. Kappel defines web engineering as: '*the application of systematic and quantifiable approaches (concepts, methods, techniques, tools) to cost-effective requirements analyses, design, implementation, testing, operation, and maintenance of high quality web applications*'. Our research uses this definition and provides a method using a CMS as a platform to implement high quality web applications. Therefore, based on this definition, we defined the following sub questions (SQs):

SQ1.1: What is a web content management system?

Before we can define the concepts and processes of web engineering based on a Web Content Management System we require a thorough understanding of the concept. The definition of a web content management system provides insight into the origin of content management systems and is a starting point for our Web Engineering Method.

SQ1.2: What is an appropriate method for requirements engineering and designing web applications using a web content management system?

Following the definition of Kappel et al, we want to know what the requirements engineering and design processes are for our Web Engineering Method. The first phase consists of requirements engineering and the second phase of the process of designing the CMS-based web application. Note that we refer to the information system design, not the graphical design of a website. A CMS provides the base architecture for a CMS-based web application but it can also be adjusted to meet specific requirements. To answer this question, the concepts of assembly-based situational method engineering are applied to find appropriate method fragments for implementing content management systems using 14 identified key concepts of content management systems.

SQ1.3: What does the realization and implementation phase look like in content management system based web engineering?.

The realization and implementation phase of CMS-based web applications consists of the actual configuration and customization of the CMS and then implementing it in the production environment. We argue that developing CMS-based web applications is challenging and different. With this research we identify what the CMS-specific challenges are when realizing and implementing a content management system and describe a method to cope with those challenges. The resulting method addresses the realization and implementation phase for CMS-based web applications.

SQ1.4: How to organize operations and maintenance in content management system based web engineering?.

Organizations lack the experience, processes and structure to effectively maintain the Web Content Management System once the CMS-based web application has been developed and implemented. To answer this question, we describe a generic IT Management framework and the inclusion of Web Content Management processes on a strategic, tactical, and operational level. Within each level, processes are designed that address the relevant activities.

RQ2: How can process innovations be included in the web engineering method?

A second motivation for this research is to find ways to innovate the web engineering method in terms of efficiency (less time or fewer steps to implement the system), reusability (reusing solutions or parts the solutions in similar situations), and ease of implementation (less configuration or coding). The goal is to provide researchers and practitioners with process models, guidelines and prototypes to improve the web engineering processes, which are reflected by our second research question. The research question is broken down into two sub questions:

SQ2.1: Can a model driven approach innovate the web engineering method?

A CMS allows a non-technical user to update and maintain a web application. With this research question we try to bridge the gap between business requirements as defined by the end-user and the realized web application as developed by the developer. We investigate and define a model-driven approach for the configuration of CMS-based web applications that aims to reduce the complexity and increase the transparency of implementations. We developed a prototype of the model-driven approach that allows business users to configure the business processes without the need of technical support.

SQ2.2: Can a product family line approach innovate the web engineering method?

The goal of this sub research question is to optimize the implementations of CMS-based web applications from a product family approach. We focus on the identification of software components that can be reused in similar situations. This leads to more efficient and effective implementations. The idea is that if organizations within a certain domain have similar requirements, the CMS implementation could be optimized by means of pre-configuration of the software product, or so called ‘Industry Templates’.

1.3.2 Research Approach

The aim of this research is to develop an artifact, the Web Engineering Method, for developing and implementing CMS-based web applications. The main research approach we use for that is *Design Science research* and we used *Method Engineering* to assemble the artifact.

Design Science Research – Most of the research we conducted is design science research [Vaishnavi and Kuechler, 2005], [Peffers et al., 2007], where solutions are designed and evaluated in different settings. Design-science research must produce a viable *design artifact* in the form of a construct, a model, a method, or an instantiation [Hevner et al., 2004]. Some examples of design artifacts in our research are the Web Engineering Method as described in chapters 2, chapter 3, chapter 4 and chapter 5, the model-driven development approach in chapter 6, and the approach for product verticalization in chapter 7.

Method Engineering – The artifacts we developed in this research are assembled using Method Engineering. Method Engineering is defined as ‘the discipline to construct new methods from existing methods’ [Brinkkemper, 1996]. That definition by itself says something about this work: we do not disregard existing methods but instead select useful fragments to create a suitable web engineering method based on Web Content Management Systems. Situational method engineering is a subset of method engineering that allows for the creation of methods that are tuned towards the unique development project situation [Ralyté et al., 2007], [Ralyté et al., 2003]. We use the situational Method Engineering approach as specific form of method engineering in order to assemble our Web Engineering Method. For example, in chapter 2, we assembled fragments from three methods into the requirements engineering process for WEM.

Prototype Evaluation – A prototype was built in order to demonstrate the feasibility of the model-driven approach to configure a CMS and to establish that it would configure a CMS-based Application based on the models as described in chapter 6. The prototype has been developed using MetaEdit+.

Case study Evaluation – The designed methods have been applied in case studies using the method provided by Yin [Yin, 2002]. The case studies have been used to validate the artifacts designed with the Design-research paradigm. Most of the research took place at GX Software, where the researcher was employed during the research.

1.4 Dissertation Outline

The different research questions and sub questions are answered throughout the sections. We focus on two topics: designing a new method for CMS-based web engineering and establishing methods and guidelines to improve the processes, which is reflected by the two main research questions and the two main parts of the this dissertation. Each individual section is based on a published paper and this dissertation contains the original contents of each paper. Only the layout has been changed to present the sections in a consistent design.

- This introductory part provides an overview of the dissertations content, lays out the principal concepts of this research, the main research questions, the approach and provides a general outline of this dissertation.

- The second part of this dissertation develops the approach to CMS-based web engineering. We define the process steps and deliverables using the definition of web engineering

- The third part extends the developed web engineering method by enriching the concepts and deliverables with a model-driven web engineering approach and prototype. Furthermore, we develop an approach to improve the web content management system product line for future implementations by identifying software commonalities in CMS-based web applications.

A overview of this dissertation is visualized in Figure 1.9

1.4.1 Part 1: Introduction

This introduction provides an overview of the dissertations content, lays out the principal concepts of this research, the main research questions, the approach and provides a general outline of this dissertation.

1.4.2 Part 2: Developing Web Content Management System based web applications

Chapter 2 reviews and analyses literature on web engineering en web information systems engineering and provides a definition of a CMS. Moreover, we design the requirements engineering processes of CMS-based web engineering by combining existing methods using a situational method engineering approach. Since a CMS is product software, the requirements engineering processes should take into account the CMS-specific modules and features. Our method introduces two routes, ‘standard’ and ‘complex’, that result in different processes depending on the complexity of the CMS-based web application. The new implementation model is much more appropriate for selecting and defining the requirements for CMS-based web engineering.

Chapter 2 was originally published as:

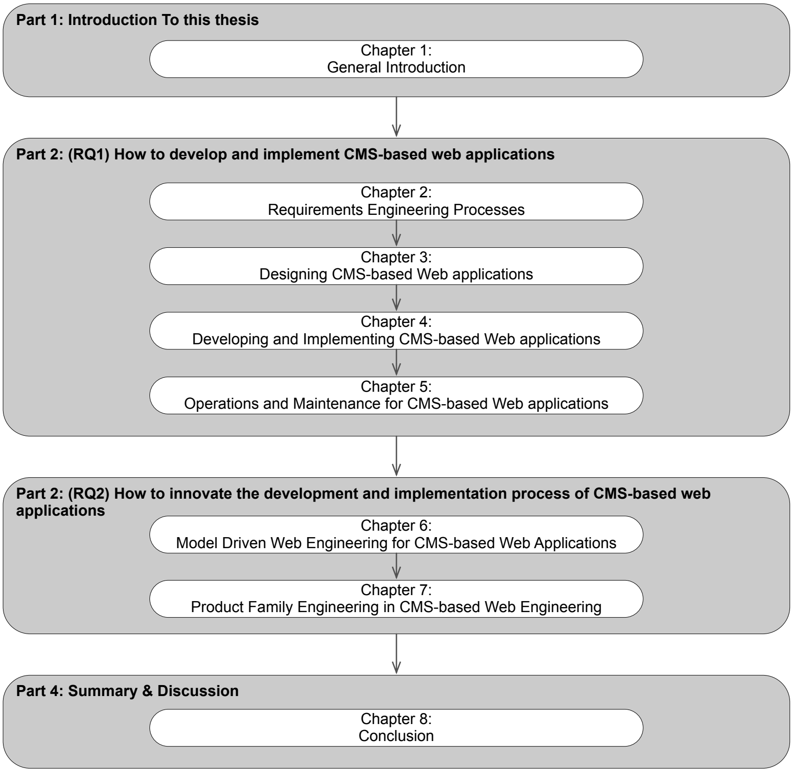

Figure 1.9: General Outline

(I) *Souer, J., Weerd, I. van de, Versendaal, J. and Brinkkemper, S. (2007) 'Situational requirements engineering for the development of Content Management System-based web applications', International Journal of web engineering and Technology (IJWET) 2007 - Vol. 3, No.4 pp. 420440 [Souer et al., 2007b]*

Chapter 3 addresses the next step of our approach to implement CMS-based web engineering: the Design Phase. The design phase addresses fourteen key CMS concepts and consists of the main activities: 'Conceptual Design', 'Architecture Design', 'Presentation Design' and 'Detailed Component Design'. The proposed method is validated through an expert validation and applied in a customer situation.

Chapter 3 was originally published as:

(I) *Souer, J., Luinenburg, L., Versendaal, J., van de Weerd, I., and Brinkkemper, S. (2008), 'A Design Method and Application for Web Content Management Implementations', in Proceedings of the 10th International Conference on Information Integration and Web-based Application & Services (iiWAS08), Linz, Austria, November 2008. pp. 351-358. [Souer et al., 2008b]*

Chapter 4 describes the next two phases of the development processes: the Realization and Implementation phase. Several researchers have investigated the differences between 'traditional software engineering' and 'web software engineering'. We demonstrate how a CMS copes with most of the known challenges in web engineering. However, *CMS-based* web engineering has led to a new set of challenges, which we will elaborate upon. Moreover, we designed the Realization and Implementation phase based on those challenges specifically related to CMS-based development and deployment.

Chapter 4 was originally published as:

(I) *Souer, J., Urlings, T., Helms, R., and Brinkkemper, S. (2011), 'Engineering Web Information Systems: A Content Management System-based Approach', Accepted for publication at the 13th International Conference on Information Integration and Web-based Application & Services (iiWAS11) [Souer et al., 2011b]*

Chapter 5 Investigates the Operations and Maintenance processes of Content Management Systems. We propose a WCMS Process Framework for the operation and maintenance phase of web engineering. In this paper we elaborate on the construction of the framework. It encompasses the description of a generic IT Management framework and the inclusion of Web Content Management processes. The WCMS Process Framework conformed to the subdivision of three managerial levels: the strategic, tactical and operational level. This also complies with the five identified issues where strategy and policy issues dictate a strategic management level, coordination issues point to a tactical management level and the production and maintenance of content designate an operational level.

Chapter 5 was originally published as a conference paper (I) and later extended on invitation to a journal version (II):

(I) Souer, J., Paul, H., Versendaal, J. and Brinkkemper, S. (2007), 'Defining Operation and Maintenance in web engineering: a Framework for CMS-based web applications', in Proceedings of the Second IEEE Internatoinal Conference on Digital Information Management (ICDIM07), Lyon, France, pp. 430-435. [Souer et al., 2007a]

(II) Souer, J., Paul, H., Versendaal, J. and Brinkkemper, S. (2008), 'A Framework for Web Content Management System Operation and Maintenance', in Journal of Digital Information Management (JDIM), Vol. 6, Nr. 4, ISSN 0972-7272, pp. 324-331. [Souer et al., 2008a]

1.4.3 Part 3: Process Innovation

Chapter 6 investigates the model-driven paradigm to improve the quality and efficiency of the implementation process. We develop a modeling tool for the automated configuration of Web Content Management Systems which aims to reduce the complexity and increase the transparency of implementations. It allows business users to configure the business processes without the need of technical support. We combine fragments of existing web engineering methods and specify an abstract and concrete syntax based on a domain model and end user analysis. The resulting WebForm Diagram has been implemented in a prototype and has been validated by subject matter experts.

Chapter 6 was published as a conference paper (I) and adjusted on invitation for a workshop (II):

(I) *Souer, J., Kupers, T., Helms, R., and Brinkkemper, S. (2009), 'Model Driven Web Engineering for the Automated Configuration of Web Content Management Systems', in Proceedings of the 9th International Conference on web engineering (ICWE09), San Sebastian, Spain, June 2009, pp. 121-135. [Souer et al., 2009]*

(II) *Souer, J., and Kupers, T. (2009), 'Towards a Pragmatic Model Driven Engineering Approach for the Development of CMS-based web applications', in Proceedings of the 5th Model Driven web engineering Workshop (MDWE09), San Sebastian, Spain, June 2009, ISSN 1613-0073, pp. 31-45. [Souer and Kupers, 2009]*

Chapter 7 investigates the product verticalization paradigm to optimize the implementations of CMS-based web applications. We focus on the identification of software components that can be reused in similar situations. This leads to more efficient and effective implementations. The idea is that when organizations within a certain domain have similar requirements, the CMS implementation could be optimized by means of pre-configuration of the software product, or so called Industry Templates. We present a new approach to identify software commonalities in CMS Product Family Engineering using atomic e-Business models. Moreover, we propose an adjusted approach for the derivation of generic requirements from atomic e-Business models to requirements where we combine goal modeling.

Chapter 7 was originally published as a conference paper (I) and later extended on invitation to a journal version (II):

(I) *Souer, J. and Joor, D. (2010), 'An Approach to Identify Commonalities In web application Engineering for a Web Content Management System', in Proceedings of the 12th International Conference on Information Integration and Web-based Application & Services (iiWAS10), Paris, France, November 2010, OCG ISBN 978-3-85403-272-4, pp. 556-563. [Souer and Joor, 2010]*

(II) *Souer, J. and Joor, D., and Helms, R. and Brinkkemper, S. (2011), 'Identifying Commonalities in Web Content Management System Engineering', in International Journal of Web Information Systems (IJWIS), Vol. 7 Iss: 3, pp. 292-308. [Souer et al., 2011a]*

1.4.4 Part 4: Conclusion and Discussion

The last part contains a summary of this dissertation and answers our research questions and the associated sub-questions. We also discuss the limitations of this research and suggest some topics for future research.

1.5 List of Acronyms

WCM Web Content Management

CMS Content Management System

WCMS Web Content Management System

CRM Customer Relationship Management

ERP Enterprise Resources Planning

ECM Enterprise Content Management

DAM Digital Asset Management

SAAS Software As A Service

WEM Web Engineering Method

PDD Process Delivery Diagram

DTAP Development, Test, Acceptance, Production

MDE Model Driven Engineering

DSML Domain Specific Modeling Language

CASE Computer Aided Software Engineering

Note that CMS and WCMS are used interchangeably. Over the last years, the industry (and researchers) moved from CMS as an all-encompassing abbreviation of Content Management Systems to WCM as Web Content Management processes and WCMS to refer to a Web Content Management System. In this dissertation we use both CMS and WCMS interchangeably.

Part II

Developing Content Management System based Web Applications

CHAPTER 2

Requirements Engineering for CMS-based Web Applications

Web applications are evolving towards strong content-centered 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 specialized in developing and implementing CMS-based web applications.[1]

[1]This work was originally published in the International Journal of Web Engineering and Technology, entitled 'Situational requirements engineering for the development of Content Management System-based web applications' in 2007 [Souer et al., 2007b]. The work is co-authored with Inge van de Weerd, Johan Versendaal, and Sjaak Brinkkemper.

2.1 Web application development and implementation

Nowadays, organizations 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, 2004a]. Therefore, applications based on web technology are considered of strategic importance [Turban et al., 2000]. These web applications should be agile enough to respond to the dynamic business and the ever-changing customer demands [Arch-int and Batanov, 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, 2004a]. 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 Paolini,]. The literature gives a plausible explanation by suggesting that the web applications differ from conventional information systems [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 utilize web technology and manage the unstructured information. CMS-based web applications are therefore implemented to support the organization 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 organizational 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) [Jacobson et al., 1999]. We integrated the method into a generic development method consisting of the phases Acquisition, Orientation, Definition, Design, Realization and Implementation. To include situationally 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) organizations? Subsequently, we pose the following questions: how do we organize the requirements engineered in the route maps and how can we adapt the method to meet the organizational 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.

2.2 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 utilization of the World Wide Web as a significant communication channel. Organizations needed to expand their information systems beyond the boundaries of their organization, 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 2.1 illustrates a categorization 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 the author in [Hoick, 2003]. 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 utilizes the technology of the web. Holck argues in 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. 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 2.1 as a ‘Static website’), and a web application as a kind of interactive information system [Troyer and Leune, 1998]. [Castro et al., 2004] and [Jeenicke et al., 2003] 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 study recognized that the challenge stems from the

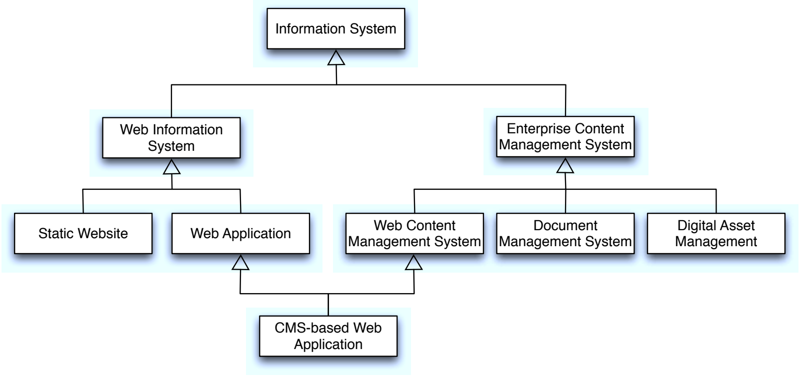

Figure 2.1: Positioning of CMS-based web applications in the information systems categorization

fact that the internet, and more specifically, the web, is a completely different computing environment compared to conventional computer-based environments [Arch-int and Batanov, 2003]. 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 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 [Vidgen et al., 2001]. To manage the information, state-of-the-art content-centered 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-Iglesias 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 utilize the technology of the web and implement business logic. As organizations 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 organization can manage their complete online strategy. With a configured CMS-based web application, organizations can provide personalized 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 organization 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 organizations to implement and customize the applications according to their requirements. In many cases, the development of these applications does not start from scratch.

2.3 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, realization and implementation.

2.3.1 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 specialized for a very large class of software systems, for different application areas, different types of organizations, different competence levels and different project sizes [Jacobson 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 specialization 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.

2.3.2 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:

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 customization 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.

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 customers 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 customization 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

Phase	*Standard*	*Complex*
Acquisition	Acquire Customer Information	Acquire Customer Information
		Feature list (UP)
	Describe solution	Describe solution
Orientation	Risk Management (UWE)	Risk Management (UWE)
Definition	Product Vision (UP)	Product Vision (UP)
	User and Domain modeling (UWE)	User and Domain modeling (UWE)
		Use Case modeling (UP)
	Application Model (UWE)	Application Model (UWE)
	Nonfunctional requirements (UP)	Nonfunctional Requirements (UP)
		Require 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	Development of customizations
		Development of Interfaces (UP)
	Functional Testing	Product Quality Assurance (UP)
		Functional and integration testing
Implementation	Product deployment	Staging deployment
		Production Deployment
	Acceptance	Acceptance

Figure 2.2: Elements of standard and complex projects

matrix gives an overview of the applied methods in the route map per phase. The actual requirements specification is done in the definition phase.

2.3.3 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).

Acquisition phase

The acquisition phase focuses on outlining the customers 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 customers 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 2.3 illustrates the meta-model of the acquisition phase in WEM. For the method visualization, a meta-modeling technique is used [van de Weerd et al., 2006] based on UML. The processes are modeled 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.

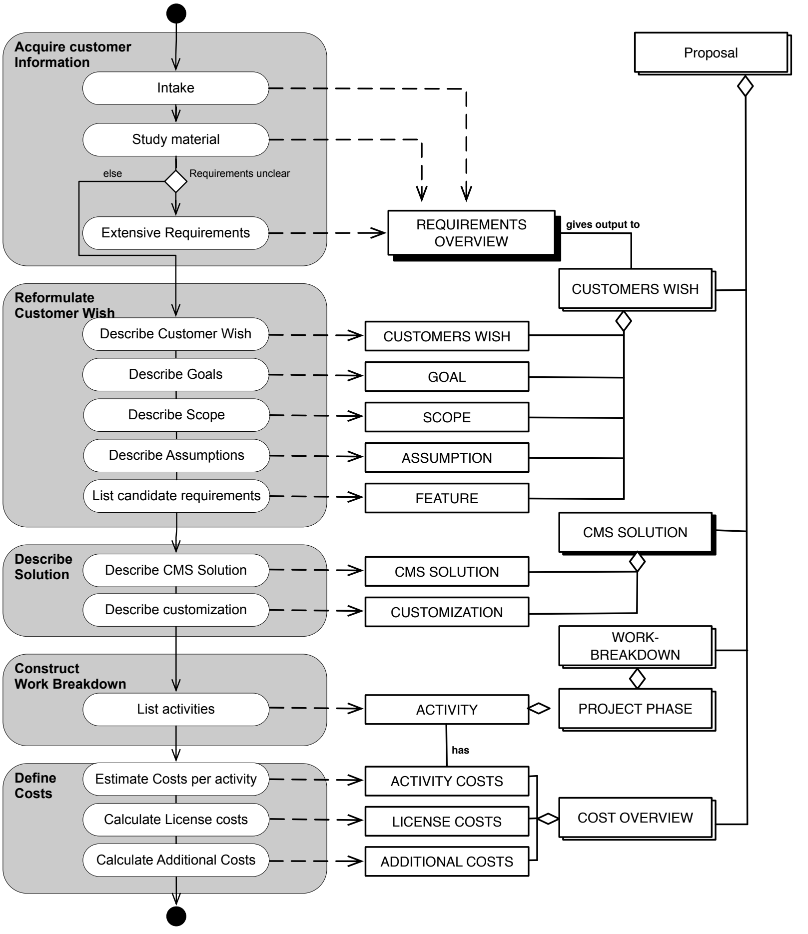

Figure 2.3: Meta-model of Acquisition Phase in WEM

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 are installed, including planning and control of the project, communications, activities and responsibilities, risk management, problem management and change management.

Definition Phase

The definition phase of WEM consists of the actual requirements elicitation, analysis and specification. A meta-model of the definition phase is illustrated in Figure 2.4. 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 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 modeling 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 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, work-flow modeling and a utilization 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 modeling and use case modeling to specify the requirements of complex issues. Use case modeling 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 minimizes 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.

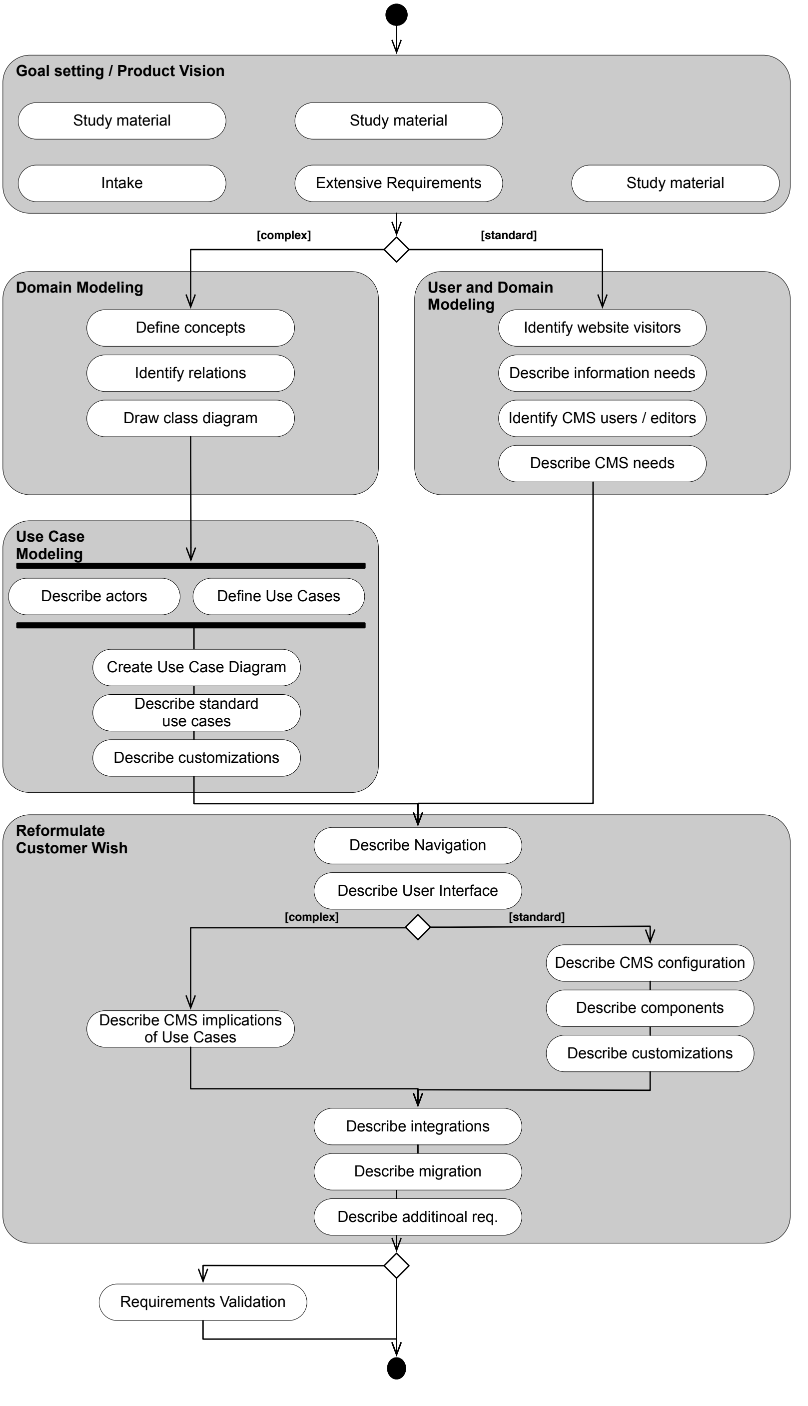

Figure 2.4: Meta-model of Definition Phase in WEM

Design Phase

During the design phase, it 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 customized 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.

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 customers 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 in the functional and integration tests, the realization phase is concluded.

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 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 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 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.

2.4 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 [van Berkum et al., 2004]. Figure 2.5 illustrates a high-level functional architecture of the 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 analyzing project aspects such as 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 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 organization, 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 characterizations. 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 characterization of a standard project. We will elaborate on one case. The other cases are discussed in Section 4.2.

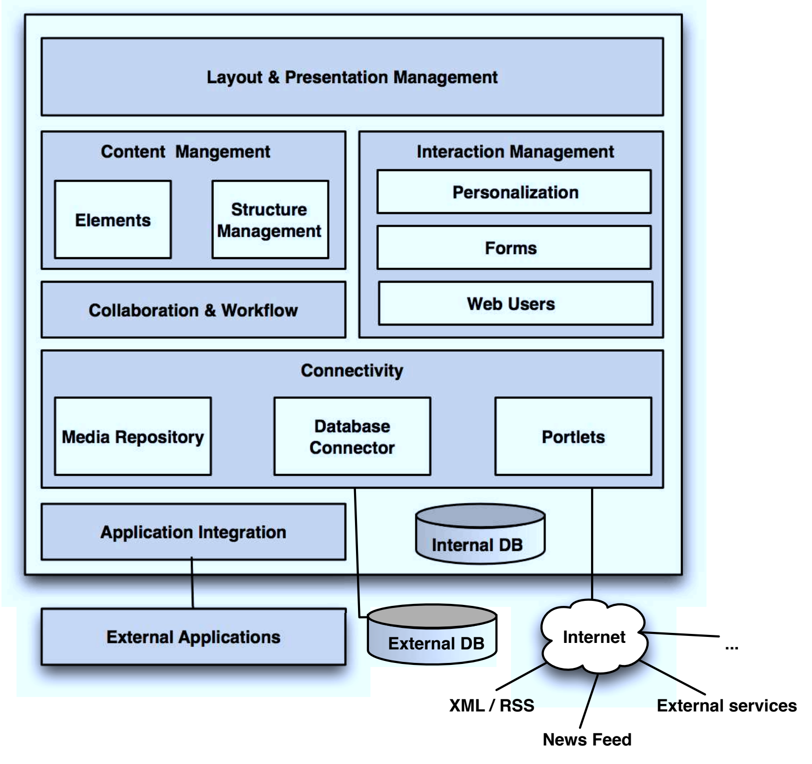

Figure 2.5: High-level Functional Architecture of GX WebManager

Case	Project type	Estimated man-hours	Estimated develop. 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

Figure 2.6: Overview of case studies

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 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 fulfillment 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)
- requirements validation.

Before using WEM, GX used primarily application modeling 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 authorization 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 modeled in a domain model. In Figure 2.7, part of the domain model is visualized.

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

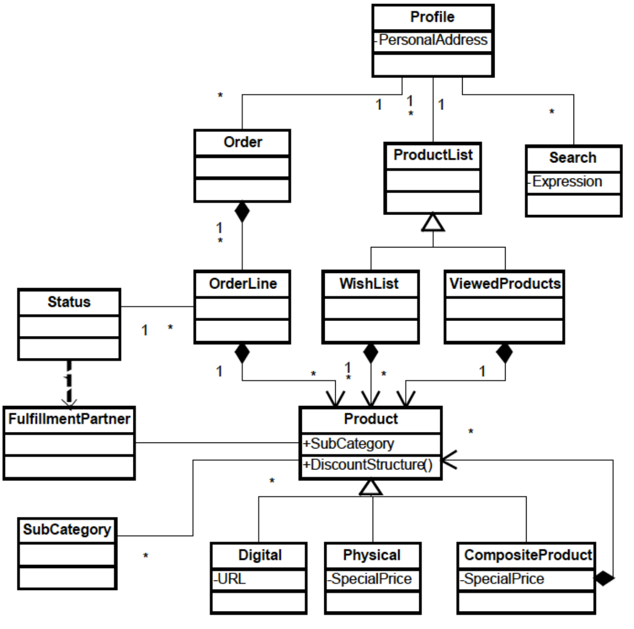

Figure 2.7: Fragment of Domain Model

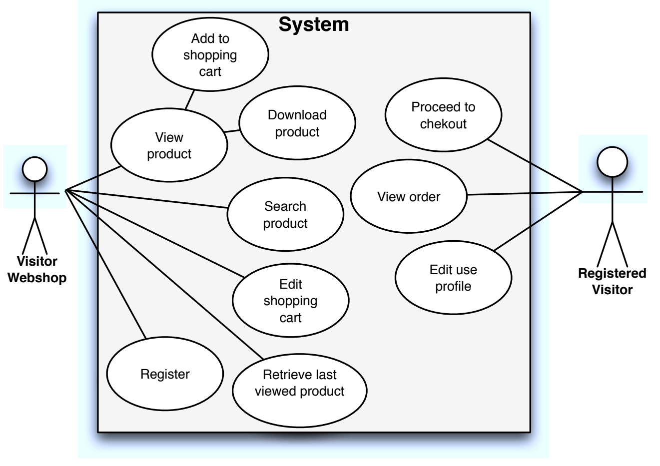

Figure 2.8: Use Case Diagram

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 2.8 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 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 modeling, 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 customization 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 fulfillment 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 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.

2.4.2 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 modeling. 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 modeling. Summarized, WEM made a significant contribution towards 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. 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 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 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 customized to the customers 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.

2.5 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) [Troyer and Leune, 1998]. WSDM is a user-centered 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 modeling. 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, 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 Modeling Language (UML) Extension for Modeling Multimedia Applications. Aspects of the application, which are covered in this extension, are logical structure, spatial presentation, predefined temporal behavior 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 modeling tool to visualize 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 modeling 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, personalization requirements, nonfunctional requirements and even multichannel requirements. Still, aspects such as application modeling, 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 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 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 modeling 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.

2.6 Conclusion

In this paper, we gave a definition and categorization 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 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 that are relevant within the CMS-based web applications. In addition, we are developing a modeling 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 clients needs to product extensions is one of the future work directions.

Chapter 3

Designing CMS-based Web Applications

web applications The process of implementing a Web Content Management System (WCM) can be complex and time consuming. We argue that there is a need for a web engineering methods specifically addressing WCM implementations. In this paper we present the design phase of our Web Engineering Method (WEM). The design phase of WEM addresses fourteen identified key WCM concepts and consists of the main activities: Conceptual Design, Architecture Design, Presentation Design and Detailed Component Design. The proposed method is validated through an expert validation and a case study. Based on the outcome, we developed a prototype of a CASE-application supporting the Conceptual Design activity. Future research includes further development of the CASE-application including the abstract syntax and automation of the transition.[1]

3.1 Introduction

Most public and private sector enterprises faced with challenges in satisfying evolving customer needs and meeting regulatory and compliance dictates, utilize Web Content Management (WCM) software. WCM software acts both as a

[1]This work was originally published in the Proceedings of the 10th International Conference on Information Integration and Web-based Application & Services, entitled ‘A Design Method and Application for Web Content Management Implementations’ in 2008 [Souer et al., 2008b]. The work is co-authored with Lutzen Luinenburg, Johan Versendaal, Inge van de Weerd and Sjaak Brinkkemper.

controlling mechanism and an enabler: it controls the processes of managing the web content with workflow, scheduling, authorization, reuse of content and archiving. WCM software enables organizations to operationalize their own content-delivery strategy with specific user interaction, personalization, and multi-channel delivery.

The implementation process of WCM systems within enterprises however can still be complex and time consuming. There are several reasons to substantiate this. First, WCM-supported web applications often involve customizations and integration with back-office systems. Secondly, WCM has a collaborative aspect since multiple users from different departments of the enterprise work simultaneously on the same content and functionality. A third reason is that WCM presents information over multiple channels (web, mobile, e-mail, print) for different purposes (sales, marketing, e-business, services, questionnaires, etc.), often in a personalized context. And fourth, implementing WCM systems is not just about the technology, but also about people and processes and therefore involve change management [Souer et al., 2007a], [Souer et al., 2008a].

There is a need for a web engineering method for implementing WCM systems in particular to improve the implementation process (quality and user satisfaction) and speed up the development process [Souer et al., 2007b]. Our resulting leading research question is: ‘What is an appropriate method for designing web applications based on a WCM system’. In [Souer et al., 2007b] and [Souer et al., 2008a] we introduced the Web Engineering Method (WEM) as a Web Engineering approach for the implementation of Web Content Management Systems in particular to obtain high maintainability and give business owners the ability to manage and control web applications. We call these type of web applications CMS-based web applications. WEM is integrated in a traditional implementation method consisting of Orientation, Definition, Design, Realization and Implementation. This paper continues this research and elaborates on the Design phase. The contribution of this research consists of the development of the design phase to implement WCM systems. Secondly, we identified fourteen key WCM concepts which are based on a literature and market analysis. And finally the feasibility of integrating a CASE-application with a WCMS system is demonstrated.

To date, the research field of Web Engineering has resulted in several methods to support the complex task of designing and creating web applications [Kappel et al., 2006], the research on the implementation of WCM however is scarce. There are several research groups working on related work. We briefly elaborate on four relevant research groups within the Web Engineering research field. Ceri et al. describe in [Ceri et al., 2003] their Web Modeling Language (WebML), a notation for specifying complex websites at a conceptual level. WebML and WEM are both focused on Web Engineering and the automated generation of web applications. Koch et al. describe the UML-based Web Engineering (UWE) approach in [Koch and Kraus, 2002]. UWE is an object-oriented, iterative and incremental approach for the development of web applications. Pastor et al. describe different methods with the Object-Oriented Web-Solutions Modeling approach (OOWS). OOWS provides mechanisms to deal with the development of hypermedia information systems and e-commerce applications in web environments [Pastor et al., 2001]. Similar to our framework: the OOWS approach is supported by a commercial software application called OlivaNova. Vdovjak and Houben address in their paper on the Hera project the integration of external content providers explicitly [Vdovjak and Houben, 2005]. Hera is a methodology that supports the design and engineering of web information systems. It is a model-driven methodology that distinguishes three parts in the design: integration, data retrieval, and presentation generation [Vdovjak et al., 2003].

We address the research question by constructing a design method from related methods using assembly-based situational method engineering, we validate it with a case study and make a first prototype of a CASE application. The rest of the paper is organized as follows. Section 2 provides an overview of the construction of the design phase of WEM. Section 3 elaborates on the design phase of WEM itself based on four main activities. Section 4 summarizes the validation of the method based on an expert validation and a case study. In section 5 we present an application of the design phase. Section 6 contains concluding remarks.

3.2 Creating the Design Phase of WEM

We apply concepts of assembly-based situational method engineering to find an appropriate method for implementing WCM systems. Situational method engineering is a useful way of approaching this type of research [Brinkkemper, 1996], [Ralyté et al., 2007]. We start with a description of assembly-based situational method engineering which we used to create an approach consisting of six steps. We elaborate on each of the six steps consecutively.

3.2.1 Assembly-based situational method engineering

Several approaches have been proposed in the field of situational method engineering. We adopted the assembly-based situational method engineering approach as proposed by Ralyté [Ralyté et al., 2003] since it was applied successfully in several cases for WCM systems [van de Weerd et al., 2006]. We added two extra steps to refine the selection method for WCM systems: the first additional step is about decomposing the notion of WCM systems in order to get its key concepts. These key concepts are reflected in the developed model. The second step consists of a comparison matrix wherein selected method fragments and key concepts are compared to extract the relevant method fragments. The approach is summarized in the following steps:

1. Analyze implementation situations and identify needs
2. Select candidate methods that meet one or more aspects of the identified WCM implementation needs.
3. Store relevant method fragments in a method base.
4. Identify key concepts of WCM
5. Compare selected methods and key concepts.
6. Select useful method fragments and assemble them in a new method.

We illustrated this approach in Figure 3.1. We elaborate on these steps in the following sections.

3.2.2 Implementation situations and need identification

To make sure that the resulting method is consistent with business requirements we analyzed different implementation situations. A specification of an implementation situation can be established by means of categorizing unique project characteristics [van Slooten and Hodes, 1996]. In total we gathered seven project characteristics by means of artifact analysis and semi-structured interviews with project managers which are organized within three areas of method configuration: Context, Organization and Technology. The project characteristics are summarized in the list below:

1. **Context**
   (a) Dependency to external activities and conditions
   (b) Level of innovation of the applied technology, methods, tools and techniques
2. **Organization**

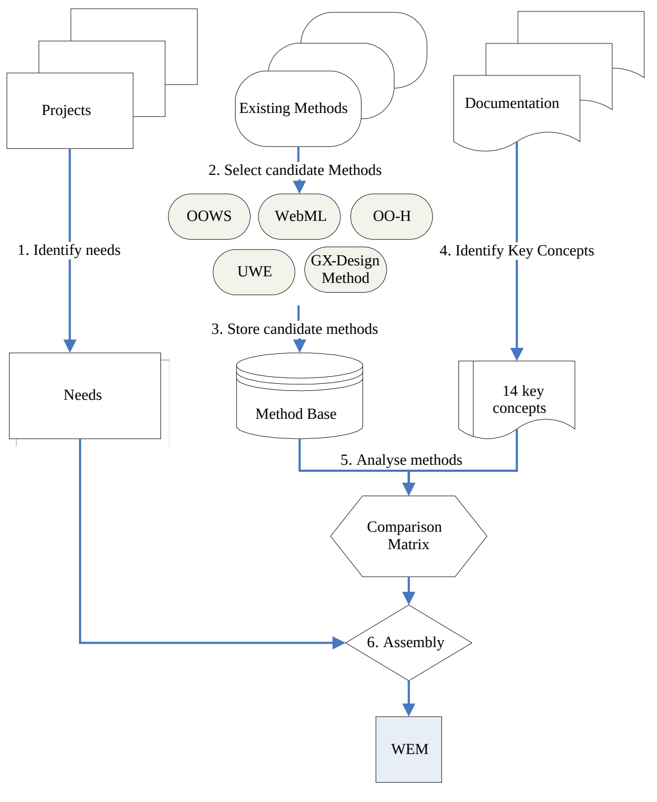

Figure 3.1: Research approach

   (a) Number and diversity of stakeholders
   (b) Uncertainty of customer's expectations by management team
   (c) Uncertainty of development activities by customer

3. **Technology**
   (a) Complexity of functional components
   (b) Number of relationships to existing systems

Based on the project characteristics, we distinguished three types of projects: standard projects; complex projects and migration projects. In this research we focus on standard and complex projects since migration has a unique set of issues by itself and this research is focused on developing new web applications and not software migrations.

Constraints were gathered by analyzing architecture and requirement documents. Moreover, conducting semi-structured interviews with two software architects and two consultants provided additional needs. We abstracted the needs and organized them into overall needs, standard implementation needs and complex implementation needs. In total, eight needs were obtained.

1. **Stakeholder documentation**: a set of understandable and easy-to-create conceptual models and design documents for stakeholders
2. **Requirements starting point**: the requirements document should be the starting point within all design activities
3. **Integration with WCM system**: integration of project lifecycle information with WCM in order to speed up the delivery process
4. **Support by CASE-application**: the design of complex and configurable user interaction components should be supported by means of a graphical application
5. **Structure navigation**: structuring the navigation should be improved to show the hierarchical structure of the website including relevant meta data
6. **Efficiency**: the design method should take the least effort as possible while maintaining high quality
7. **Reuse**: a minimum effort should be put in the creation of design deliverables by means of reusing knowledge of preceding projects where possible

8. **Use case alignment**: conceptual design models should be aligned with the use case descriptions

These eight needs are the defined constraints for the design phase.

3.2.3 Selecting candidate Methods

The second step is the selection of candidate methods who will become the foundation of our framework. As a result of a performed literature study, we selected the following approaches: Hypermedia Design Methodology [Garzotto et al., 1991], the Object Oriented Hypermedia Design Model [Schwabe and Rossi, 1995], Website Design Method [Troyer and Leune, 1998], Web Modeling Language [Ceri et al., 2003], UML-based Web Engineering [Koch, 1999], Object Oriented Hypermedia [Gómez and Cachero, 2003] and Object Oriented Web Solutions [Pastor et al., 2006].

We narrowed down the list of web application modeling approaches to a small comprehensible set of useful approaches based on the following criteria:

1. Acceptance by the community as a web application modeling approach;
2. Matured evolution of the approach; and
3. Advanced software application support.

The first criterion refers to the influence of the approach as well to the utilization of the approach as a web application modeling approach. This criterion also implies that the web modeling approach should be described in literature sufficiently. The second criterion is about the evolution and whether the approach matured over time. The last criterion is about advanced software application support since the development of web applications can be supported very well by software applications with the aim to speed up the design and implementation processes which is in line with one of our research goals.

Based on these criteria, we selected the following four approaches: WebML (WebRatio toolsupport), OO-H (VisualWADE), UWE (ArguUWE) and OOWS (Olivanova).

3.2.4 Method Base

The third step consists of the extraction and storage of method fragments in the method base. We used a meta-modeling technique called Process Deliverable Diagrams (PDDs) to analyze, select, store, and assemble the method fragments. The PDD technique is based on a UML activity diagram, reflecting the activity

Method Fragments →

Key Concepts ↓		WebML																	
		E-R Schema								Hypertext Model									
		Entity	Relationship	User Entity	Core Sub-schema	Interconnection sub-schema	Access sub-schema	Personalization sub-schema	Process sub-schema	Site view	Page	Content unit	Operation unit	Display unit	Entry unit	Selector	Link	Link parameter	Process Management Unit
E-Form	Step	x							x				x		x				x
	Handler	x											x					x	
	Validation												x				x		
	Router												x						
	Field	x																	
Personalization	User Profile	x										x	x			x		x	
	User Profile	x		x				x					x						
	User Access									x			x						

Figure 3.2: Excerpt of Comparison Matrix

side, and a UML class diagram which reflects the deliverables side. The UML semantics are strictly applied, however, some primitives have been added in order to deal with composition issues. For detailed information about the meta-modeling technique, we refer to [van de Weerd et al., 2006].

Examples of method fragments are the Hypertext Model and E-R Schema from WebML. A total of 32 method fragments have been extracted and stored in a method based. Each method contains multiple concepts. In total, 92 concepts of the four web modeling methods are used for association and positioned against the key feature groupings and WCMS functionalities.

3.2.5 Identification of Key WCM Concepts

Step four is the identification of key WCM concepts. The purpose of identifying the key concepts of WCM is to enable the comparison, selection and analysis of method fragments with respect to WCM concepts. To identify key concepts of Web Content Management, we analyzed the available scientific papers [McKeever, 2003], [Vidgen et al., 2001] and added market research reports.

All concepts were assessed by means of two expert validation interviews. One expert held a position as senior solution consultant and the other expert was a product manager. Based on the discussion and the comments given, the list of concepts has been refined and narrowed down to a list of 14 concepts. We distinguished the following 14 key concepts for Web Content Management:

Authoring Authoring deals with the production of web content (also known as the 'content life cycle'). According to the authors in [Vidgen et al., 2001] the content lifecycle covers creation to archiving and destruction of content components.

Authorization Management Authorization is used to infer which privileges authenticated users are granted [Wimmer et al., 2007]. Additionally, the authorizations have to be checked when a particular users accesses data which is relevant because of the collaborative aspects of WCM.

Connectivity Management Connectivity management covers the notions of application integration, interaction, database pages and portal integration.

Content Repository Content repositories provide services which allow for storage of content objects within the database associated with meta-data.

Deployment and Replication The deployment process potentially contains the delivery, assembly, and maintenance of a particular installed software system at a website [van der Hoek et al., 1997]. Note that the deployment aspect could be outsourced when for instance using a WCM based on 'Software as a Service' (known as SAAS).

Digital Asset Management Digital Asset Management (DAM) can be defined as the management of digital content so that it can be cataloged, searched and customized [Natu and Mendonca, 2003]. Enterprises can achieve strategic advantage by implementing a DAM because of the need to incorporate a growing quantity of graphics and rich media into corporate websites.

E-Forms / Transaction Management E-forms are the means which allows a user to interact with a web application and so its services and data. Moreover, e-forms enable companies to provide self-services or other services in an easy way.

Layout and Presentation Management WCM systems separate content, structure and presentation. Based on templates and style sheets the layout and presentation are generated. The main goal of templates is to provide a web application the same look and feel, but it also makes web applications easy to update.

Multi-channel delivery and syndication Customers make use of several channels (e.g. internet, SMS, print) in order to gather information or to interact with companies. However, it is for companies a challenge to synchronize these channels to gain advantage from a multi-channel strategy. One of the best known syndication standards is RSS (Really Simple Syndication).

Personalization Personalization is about tailoring of content, presentation or navigation based on user preferences or user behavior ([Ceri et al., 2003], [Eirinaki and Vazirgiannis, 2003], [Kappel et al., 2006]).

Website Management Website management can be defined as the organization of the structure of a WCM system in terms of effective navigation and content.

Community Technologies Community technologies or web 2.0 concepts are probably still too intangible for a solid classification, however it can be said that the Web 2.0 approach emphasizes interaction, community and openness [Millard and Ross, 2006]. Two community technologies concepts namely blogs and wikis. Both concepts have in common that it is mainly about users who can generate and publish content, while another can edit or reply on it.

Web Analytics Web analytics, similar to the notion of web usage mining can be defined as the application of data mining techniques to discover usage patterns from web data, in order to understand and better serve the needs of Web-based applications. Also McKeever identified reporting tools as in integral part of the Web Content Management [McKeever, 2003].

Workflow The concept workflow can be defined as a collection of tasks organized to accomplish some business process ([Georgakopoulos et al., 1995]). A more complete definition of the concept workflow is 'the automation of a business process, in whole or part, during which documents, information or tasks are passed form one participant to another for action, according to a set of procedural rules' [Coalition, 1999].

In the next step we make a comparison based on these key WCM concepts.

3.2.6 Comparison and selection of method fragments

The fifth step consists of analyzes of the method fragments by comparing functionalities and concepts with meta-model concepts. The purpose is to select relevant method fragments based on the functional criteria and results in a comparison matrix including the design rationale. Figure 3.2 gives an excerpt of the comparison matrix that in total consists of 31 rows and 92 columns

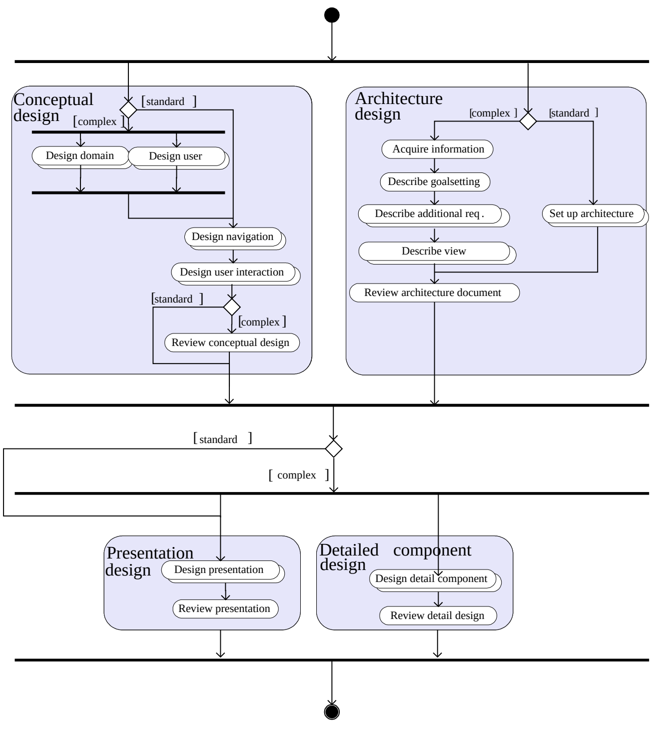

Figure 3.3: WEM Design Phase Overview

By analyzing the comparison table we determined the applicability of each of the models. Additionally, the comparison table provides a functional overview for the selection of models and is the basis for the assembly of the situational method. Furthermore it shows the coverage of WCM functionalities by several web application modeling approaches. Based on the comparison table, we selected the relevant method fragments and constructed the Design phase of WEM.

3.3 The Design phase of WEM

Similar to our earlier research in [Souer et al., 2007b] we made two routes: one for standard projects and one for complex projects. The design phase of WEM consists of the following four activities:

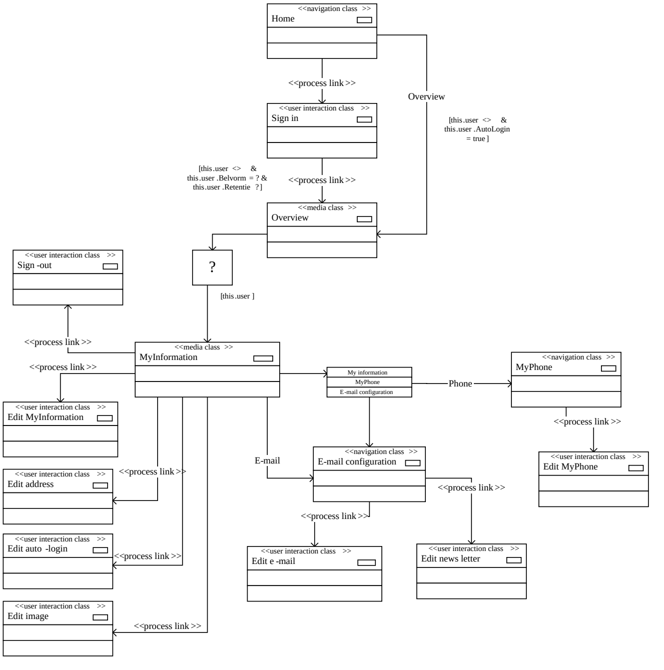

Figure 3.4: Navigational Model

1. **Conceptual Design**: providing a general design of the domain, the users and the interaction of the user with the system.

2. **Architectural Design**: The architecture design describes the system in a 4+1 view, providing the process and deployment of the system.

3. **Presentation Design**: detailing the visual presentation of the website.

4. **Detailed Component Design**: providing the technical details of customized components.

An integrated overview of the design phase of WEM is presented in Figure 3.3. The following sections elaborate on each of the four main activities.

3.3.1 Conceptual Design

As visualized in Figure 3.3, the conceptual design starts with domain modeling within the complex route. In this sub-activity, a domain model is created which represents the main concepts and relationships of the problem domain derived from the requirements document. Concurrently to this step, users of the WCM system can be modeled similar to a domain model.

After that, a navigation model is created (Figure 3.4). The navigation model originates from the navigation model (UWE) but is extended with several WCM specific navigation nodes such as: media class, database class and user interaction class. The first step in navigation design is identifying the navigation view. The navigation nodes are then constructed based on the navigation. An example of a navigation node is a navigation class which is a stereotyped class which can be visited by webusers. Access primitives are then added which are nodes that specify the way a navigation node is accessed (e.g. by query). Last, conditions can be added to links wherein personalization rules are specified. An example of a personalization rule is for example: ”this page can only be accessed by registered users”.

Following, the user interaction design is created by using a business process model (BPM) (Figure 3.5). The business process model is developed by a set of activities performed by a web visitor interacting with the WCM system in separate swimming lanes. We adopt the Business Process Management Notation (BPMN) similar to [Brambilla, 2006]. Complex implementation often involve new business process models in contrast to the standard route where reusable subsets are available. The conceptual design activity is finalized by a review on all conceptual design models with the objective to raise the quality of the conceptual design deliverables.

3.3.2 Architecture Design

The software architecture of the Web Content Management System is described by means of the 4+1 view model developed by Kruchten [Kruchten, 1995]. The 4+1 model describes a software architecture using five concurrent views: logical view, process view, implementation view, deployment view and requirements view.

In the architecture design activity an architecture document is delivered based on the specified requirements. The architecture consists of the foundation of the WCM system itself and the customizations for the project at hand.

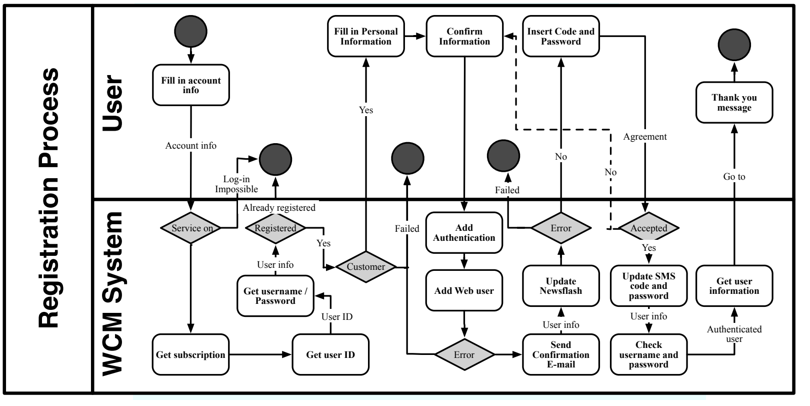

Figure 3.5: Business Process Model

Moreover, project specific non-functional requirements are taken into account (e.g. performance, security, availability). The non-functional requirements are described and categorized.

3.3.3 Presentation Design

Part of a WCM implementation is the construction of the visual presentation of the website which usually is developed by an external graphical designer consisting of static (X)HTML prototype. The implementation of the presentation consists of making the graphical design dynamic that the content can be managed from within the WCMS. The XHTML prototype typically consists of XHTML pages including a CSS, javascript and other frontend file files (e.g. Flash, Shockwave, Silverlight), possibly supported by AJAX libraries (such as BackBase, Echo2, Rico). The XHTML prototype will be assessed for suitability and the content management part is determined. Based on presentation types described in the document the static XHTML files will be implemented to dynamically generated XHTML.

3.3.4 Detailed Component Design

The detailed component design activity covers the process of designing customizations of the WCM system to meet the functional requirements. The customization could be adjustments on the existing components or the development of new components within a WCM implementation project. The main result of this activity is the detailed design document.

The detailed component design activity starts with setting a goal of the component. Then requirements document and architecture document are analyzed and a conceptual description of the component is written. Such a description contains a textual description, a presentation (e.g. table or diagram) and rationale. The description is related to functionality part of one or more (future) components which are described within the implementation view, part of the architecture document. During the final step, the detail design document is reviewed by an architect. Customizations can have a large impact on the system and its stability. A proper way to cope with customizations as described in [Souer and van Mierloo, 2008].

3.4 Validation

According to [Dul and Hak, 2007] and [Yin, 2002] we validated the Design phase of WEM through an expert review and a case study. An expert review can be conducted for confirming the relevance of a theory [Dul and Hak, 2007]. With the case study we experimented with the method in a real life context.

3.4.1 Expert validation

WEM has been evaluated in collaboration with four experts: two functional consultants and two technical architects. The validation has been performed to evaluate the correctness and completeness of the situational method, and to improve the design.

The research and the validation was performed within GX, a Web Content Management vendor. GX develops and implements a proprietary Web Content Management System called GX WebManager based on open standards, which is described in [van Berkum et al., 2004] and [Souer and van Mierloo, 2008].

The selected experts held the positions consultant or software architect within GX who are responsible for the functional and the technical solution respectively. During the interviews both route maps (standard and complex) were validated. Before validation we gave an introduction to the meta-modeling technique. Following to that, we walked through all relevant method fragments, including the high-level overviews of the different route maps. Next to that, we tried to reveal their perception concerning the usage of PDDs for the employment of route maps.

Three experts agreed upon the decision to use a PDD instead of a decision table for decision-making within project situations.One expert stated that ”the visual aspect of the PDD provides more overview and is easier to use than a decision table.” However, one expert stated that a distinction between standard and complex projects is debatable. The expert prefers to reason from a complex implementation situation. When some deliverables are not needed, he leaves them out. When discussing the PDD's, all fragments were positively validated in terms of completeness and correctness: the process steps were perceived as logical precedents and the relationships between the concepts and process steps were seen as correct and useful. Last, the concepts were perceived as nicely related to the concepts of the requirements phase.

In total, approximately 30 changes have been proposed by the experts differing from a cosmetic level to a serious request with a high impact on the initial design. These results have led to adjustments of the situational design method which we used in the case study. An example of a change request which was given when evaluating the 'presentation design' method fragment is: "A presentation variant [concept] is not the same as a presentation, but rather a sub-type; some presentational parameters are changed within the presentation, which accomplishes a new presentation variant."

3.4.2 Case Study

After refinement of the models we applied the design phase of WEM in two real life cases. We describe one of them: the case organization is a large Dutch telecommunications provider which provides telephone, internet and television services to individuals and organizations within the Netherlands. In this case study the complex route map of the 'conceptual design activity' was put to practice based on a recently completed project. The case project provides customers a personal environment with special services: online phone bill, composing Short Message Services (SMS) and Multimedia Messaging Services (MMS) and the ability to change personal information. In total, 16 selected use cases have been designed.

The project had an estimated effort of 1500 man hours. During this phase, the functionalities were restyled and the design took approximately 40 man hours. The case study started with a thorough analysis of the use cases within the requirements document and the implemented WCM system. In total, 29 diagrams have been created, whereof one domain model, one user model, three navigation models and 24 business process diagrams. After the first design, in total two involved software developers and two consultant validated the produced models and filled-out a survey. In this survey, for each deliverable, five questions were asked with regard to the readability, abstraction level, correctness, software application supportability and applicability of the conceptual design deliverables. After filling-out the survey, informal interview were held in order to discuss the usefulness of the deliverables and the situational web design method.

Results

We observed that the respondents were positive concerning the method by itself, although the results diverged depending on the role and the diagrams discussed. First, software developers were more positive about using domain models than consultants, since the latter considered the models too complex for customers. Both architect and consultant agreed that the domain model could be useful to define customizations. This corresponds to our method where we positioned the domain model in the complex route. Some remarks made by the architect concerns the separation of domain and user model which was perceived as an unnecessary separation.The navigational model was found useful by the software developers since it could provide a good alternative for the current description of navigation (usually a spreadsheet). The consultants had contradicting opinions about this navigational model especially concerning the ability to communicate it with all stakeholders.

The Business Process Diagram (BPD) was found the most useful by both software developers and consultants as it visualizes complex user interaction flows and it improves the communication with customers. Moreover, it provides an overview of the interaction as well as the integration aspects. ”When requirements for complex flows are written down textually by a consultant, it is hard to interpret the requirements and transform them into configured functionalities without any ambiguity, but this diagram could filter out misinterpretations” concluded the architect. One software developer stated that the BPD is the clearest model of all conceptual models. The architecture design and the detailed component design was not changed in this method.

3.5 CASE application support

Based on the results of the validation and to further support the Design Phase of WEM we developed a prototype of a Computer-aided Software Engineering (CASE) application. The purpose of the CASE application is to improve the efficiency of the application development. We focus with the CASE-application on the Conceptual Design of WEM and more specific on the Business Process Diagram for the following reasons: (1) the Conceptual Design defines standard components in a WCMS and is suitable for automation. Moreover, interviews with experts showed that defining the user interaction and configuring the WCMS is a time consuming task. (2) the Architecture Design results in a description of the physical server setup and the application architecture which is very hard to automate. (3) the Presentation Design is an interpretation of static XHTML files

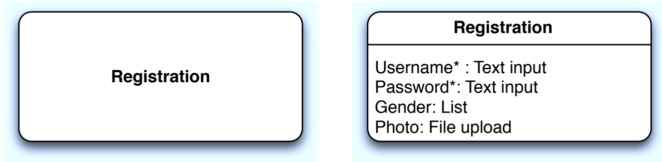

Figure 3.6: Notation of a Step and Formelements

which will then be made dynamic. There are more then one (commercial and free) visual software applications available to create dynamic web pages. 4) the Detailed Component Design is about detailing the customizations of the web application. This customization could be just about anything which makes it hard to formalize it in a CASE-application.

For the development of the CASE-application we need to formalize the models we want to support and develop a graphical notation which corresponds with the formalized models. We use concrete and abstract syntax similar to [Cooper, 2007a] to define our modeling language. In this paper we elaborate on the graphical notation.

Based on interviews, we gathered input how the different elements of a form should be represented graphically. The interviewees had consensus on how a step should look like: a step was drawn as a square. Form fields, which are related to a single step, are properties of a step. We represented the form fields similar to UML Class-Attributes. The user should be able to choose wether he would like to view the form fields or not (different abstraction layers). An example of a step and a step with form fields is shown in image Figure 3.6

For each form element, we made a representation in the CASE application. We modeled the Business Process Diagram as presented in 3.5 with our CASE-application.

All elements in the XML have a unique identifiers. The elements have internal references to other elements using these identifiers. A single step for instance will normally have multiple formfields. The XML is then transformed to an XML format which can be imported into the WCMS GX WebManager. We successfully imported the example in GX WebManager and proved thereby that the application could be used to define Business Process Models during requirements phase and implement it into the WCMS.

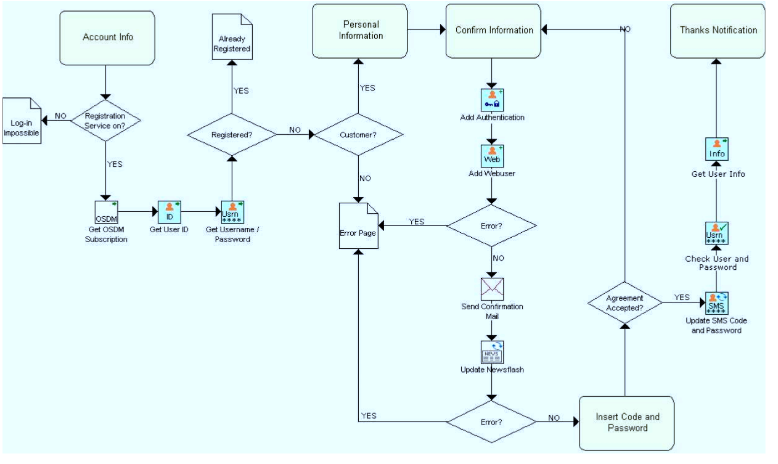

Figure 3.7: BPD within the WEM application

3.6 Conclusions

In this paper we elaborated on the research question 'What is an appropriate method for designing web applications based on a WCM system'. To answer this question we constructed the design phase of WEM: an implementation method specifically addressing WCM systems. The design phase has a standard and a complex track and contains of four main activities: Conceptual Design, Architecture Design, Presentation Design and Detailed Component Design. We validated the designs through an expert validation and a case study. The overall response on the validation was positive although only the Business Process Diagrams was unanimously supported. Based on the outcome of the validation, we made an application which supports the Design phase of WEM. For the application, we formalized the conceptual design diagrams and developed a graphical notation. The application generates XML which can be imported into the WCMS.

We believe that we have made a step forwards in developing a method for the implementation of WCM systems although the results show that not all diagrams are as useful. Furthermore, we think that the approach through the identification of key concepts and the application of situational method engineering is useful beyond the scope of Web Engineering. Future research consists of completion of the WEM framework: construction of the implementation phase. Part of the

WEM framework is the sophistication of the described CASE-application in order to support the implementation processes and speed up development.

Chapter 4

Realizing and Implementing CMS-based Web Applications

Web applications are evolving into large enterprise software applications accessible through the web. Web applications are often constructed based on Web Content Management software products. Web Engineering, the research field concerning high quality and cost effective development of web applications has little research on CMS-based Web Engineering. In this paper, we propose a method for the development and deployment of CMS-based web Applications. We designed the two methods based on the challenges specifically related to CMS-based development and deployment. Based on project evaluations and interviews within a case study company, the designed method is validated by expert reviews. [1]

4.1 Introduction

A unique characteristic of a web information system compared to traditional (non-web) systems is the large amount of content organized in a web structure that is realized via hyperlinks that are available to a large number of potentially divers end-users [Barna et al., 2003]. This lead to the emergence of Web

[1] A short version of this has been accepted at the 13th International Conference on Information Integration and Web-based Applications & Services, entitled 'Engineering Web Information Systems: A Content Management System-based Approach' in 2011 [Souer et al., 2011b]. The work is co-authored with Thijs Urlings, Remko Helms, and Sjaak Brinkkemper.

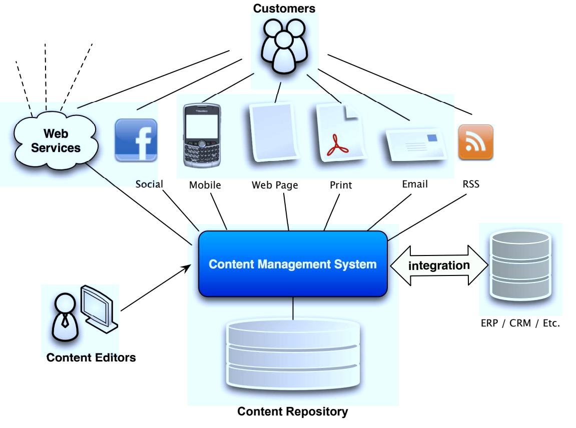

Figure 4.1: Overview of a CMS

Content Management Systems (CMS) to allow non-technical users (usually the online marketers within enterprises) to manage and control web content. A CMS, is a software product for the management and control of web content[Souer et al., 2007b]. Most modern websites are based on a CMS since it provides a set of functionalities which otherwise would have to be build from scratch. However, there is little known about how software products affect existing system development processes or what new processes are needed [Brownsword et al., 2000]. IT Research and advisory firms predict the software market for (commercial) Web Content Management exceeding 1 billion dollar in 2010 [Gartner, 2010]. It is not uncommon to have at least the same amount of realization and implementation costs, up to three or four times the cost in complex projects. And as the function of the web application became more strategic to optimize the communication within the supply chain to do business with consumers or other businesses [Lee and Shirani, 2004b] the number of integrations with other systems such as Enterprise Resources Planning, Customer Relationship Management systems and Workflow Systems increased and the more challenging it became for organizations to develop their web application [Vidgen et al., 2001]. Optimizing this realization and implementation process has a huge potential economical impact.

One of the challenges for examples is that developers need to have a deep understanding of the software product used for the development (the CMS) to match functional requirements with the existing functionalities in the product and have to decide if they have to adjust the requirements to meet the standard functionalities or customize the product to meet the exact requirements. We argue that the development process of CMS-based web application requires a different approach which we call the Web Engineering Method (WEM). WEM is based on the Web Engineering process as defined by Kappel et al [Kappel et al., 2006]. WEM consists of five phases: Orientation, Definition, Design, Realization and Implementation. The first three phases are described in [Souer et al., 2007b], and [Souer et al., 2008b]. This paper focuses on the Realization and Implementation phase. The leading research question in this paper is therefore: 'What does the realization and implementation phase look like in a process of developing web applications based on a CMS'.

The rest of this paper is organized as follows: This article continues with the research approach in section 2, we provide a brief overview of the case study organization as well. In section 3 we elaborate on the challenges in Web Engineering. First we explain why CMS cope with most of the Web Engineering challenges, but define a few CMS-specific challenges as well. In section 4 we present the main artifact of this article, the Realization phase and the Implementation phase. We explicitly show how we cope with the CMS-specific challenges. In section 5 we discuss this research, and the article is concluded in section 6 with a summary of the contributions.

4.2 Research Approach

The main result from this research is the process description of the Realization and Implementation phase of a CMS-based web application. The processes have been developed using design science, applying the model as specified by the authors in [Hevner et al., 2004]. Design-Science research has a set of guidelines which help the researcher execute, evaluate and present design science research: Design as an Artifact, Problem Relevance, Design Evaluation, Research Contribution, Research Rigor, Design as a Search Process, and Communication of Research.

The *artifact* that is developed in this research is a method for the development and implementation of CMS-based web applications. The method can be used as a guideline for organizations who are developing and implementing a web applications based on a CMS. We elaborated on the *problem relevance* in the introduction where we stated that most web applications are based on a CMS, however there is very little research addressing the specific challenges for coping with a CMS. If not addressed properly, these challenges can lead to expensive software projects that will not meet the expected outcome. We did a case study as an observational *evaluation* with expert interviews to gather in-depth knowledge. The main *research contribution* is the artifact itself, the development and implementation method, and an insight in how an product software company copes with the specific challenges of CMS implementations. In regards to *research rigor*, both the methods for the realization phase and the implementation phase are constructed trough a process of questioning different employees at the case study company. For the first iteration of the diagrams two engineers and an architect were interviewed. Based on these results, the interview was adjusted to get even more precise information. After this adjustment, another two architects the case study company were interviewed. To get some specific information about certain points in the then current diagrams, a questionnaire was made and an engineer and an architect were asked to answer them. Based on these results the diagrams were again adjusted. A final check was made with a project manager, after which the final diagrams were created. This iterative approach illustrates the *Design as a Search Process* as well. Moreover, we compared the produced solutions with those constructed by experts human designers for the same problem situation in the form of an expert review. To conclude, we communicated the research in this paper as well as several presentations to stakeholders in the case study company which will incorporate the method in its methodology.

4.3 Challenges in Web Engineering

Several researchers have investigated the differences between 'traditional software engineering' and 'web software engineering', e.g. [Hoick, 2003], [Kappel et al., 2006], [Overmyer, 2000], [Lowe and Henderson-Sellers, 2001]. Similar to Pressman [Pressman, 1996], we believe that Web Engineering is a combination of both traditional and web specific since web applications have evolved from small and networked web applications into full blown web applications, enabling complete business processes. Thereby the strategic importance of web applications moved from 'just another channel' into the primary business process of organizations. The result was that the list with traditional software requirements for large enterprise software (concerning security, stability, scalability, performance, etc) became just as important in Web Engineering as in any other strategic software.

Kappel *et al.* defined a list challenges concerning web projects [Kappel et al., 2006]. However, a CMS can in general cope with most of these challenges. In the next section we will give an overview of the challenges, and how CMSs are addressing these specific topics.

1. **Apparent Simplicity**: Modern Web applications are fully fledged software

systems connected to database and have dynamically just-in-time generated content.
a CMS is a standardized fully fledged software system. A core functionality is generating just-in-time content which is stored in a centralized repository.

2. **Aesthetics**: Keeping a website updated and relevant. For Aesthetic reasons, websites are updated regularly – most of the time, the update consists of content-updates.
Allowing non-technical users to change content is one of the main reasons why a CMS exists in the first place.

3. **Spontaneity**: Users have to be able to use the web applications without instructions to stay loyal.
The usability of the web application is part of the design. A CMS provides additional usability with personalization, recommendations, search

4. **Ubiquity**: It is difficult for organizations to define the requirements for the web visitor.
This is not solved by a CMS. However, a CMS helps with testing frameworks, allowing organizations to perform A/B testing or Multi Variate Testing to optimize the website for the web visitors

5. **Compatibility**: It is difficult for developers to support all browsers.
a CMS helps with a standardized generation of XHTML and templates which provides a high probability that it will work cross-browser

6. **Stability and Security**: Web visitors expect web applications to be available *24*x*7* and secure.
A CMS provides a platform with a secure and reliable software tested and used in production in many 'live'-situation. Security patches are embedded in the CMS platform that allows you to develop a stable and secure web application

7. **Scalability**: Web visitors expect good performing web applications and a website should be able to process many web visitors.
Scalability is part of the architecture of a CMS, as we presented in [Souer and van Mierloo, 2008] a platform itself can be scalable as defined in the deployment view

A CMS does not solve all Web Engineering problems, but what we emphasize here is that one reason a CMS has emerged of the last ten years is that it is able to cope with the most known challenges of Web Engineering. However, implementing CMS-based Web Engineering has lead to a new set of challenges on which we elaborate in the next section.

1. *Standard Platform*: The main difference between traditional Web Engineering and CMS-based Web Engineering lies in the fact that you choose a specific platform. Each CMS has its blueprint and its own advantages and disadvantages depending on an organizations requirements from a technical (e.g. Java / .NET / PHP), infrastructure (e.g. MS IIS, Apache, IBM Websphere, SAP Netweaver), functional (e.g. content reuse, workflow, personalization, integration) point of view.

2. *Template development*: A challenge is translating design or working prototype into a dynamic template. The authors in [Vigder and Dean, 1997] have identified a similar problems: Sets of components may be mismatched when implementing product software. In Web Engineering, the functionalities are partly determined by the prototype and design.

3. *Separation of customization* One of the biggest challenges is managing customizations in web applications. A known problem with CMS' is that if the CMS is customized itself, upgrading to a newer release is going to be a problem. The authors in [Repenning et al., 2001] propose a component based approach as customization, as it combines quality products with a fast time-to-market and effectively development of large, complex, and distributed Web-based applications.

4. *Dual lifecycle: platform and customized implementation*: as described in [Souer and Joor, 2010], the dual lifecycle consists of the product line itself (the CMS), and the implementation (CMS-based web application). A challenge in this process is how to ensure that both customizations and implementations can be upgraded without interference or issues concerning stability or functionality.

With these specific challenges in mind, we defined the realization and implementation phase on which we elaborate in the next section. Note that we do not address the first challenge, since the selection of a standard platform is in the 'Acquisition phase' as described in [Souer et al., 2007b].

4.4 The Realization and Implementation Phase

The main artifact of this research is the realization and implementation method. We elaborate on both in the sections below. This research was conducted at

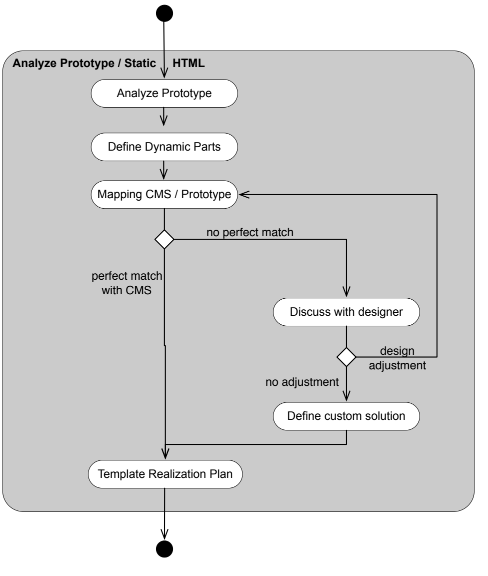

Figure 4.2: meta-model of template analasys

GX Software,a software company with approximately 120 employees, active in both the Netherlands and the United States. GX Software has approximately 150 customers, with several product deployments each. GX Software has two product lines: (1) a CMS called GX WebManager, and (2) a customer engagement system called BlueConic. This research is focused on its CMS GX WebManager – we will refer to this product as the CMS.

4.4.1 The Realization phase

The main goal of the Realization phase is the development of the web application which has been defined in the the requirements phase [Souer et al., 2007b] and technically designed in the design phase [Souer et al., 2008b]. It starts with a technical validation of the requirements document from an engineering perspective. Both the architect and the engineer see if there are specific components defined which require special attention such as dependencies and risks and leads to a list of priorities. Then the development environment are configured, similar to the production environment and based on the platform specific requirements (e.g. Oracle database, Jboss application server, and Microsoft IIS Webserver). Also

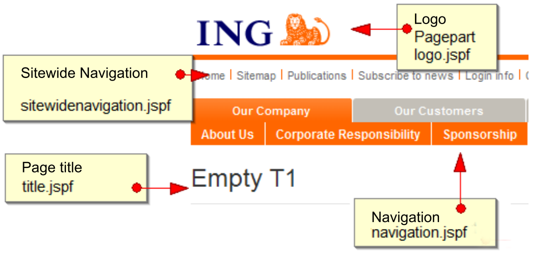

Figure 4.3: Example (fragment) of template analysis

both a revision control system and an issue tracking system are set up to support the development team.

Simultaneously, one of the CMS-specific challenges is addressed: the static HTML or prototype is analyzed to see (a) what parts have to be made dynamic, and (b) what part of the CMS corresponds to it. In [Souer et al., 2008b] we mentioned the 'Presentation Design' which is the functional analysis of the design. In this phase, the design is translated into code snippets. Every CMS has their own set of concepts and structure to make templates dynamic. Not all aspects of the design will fit into the CMS. There are basically two options in such a case, either the design is adjusted and reexamined by the engineer, or the design stays what it is which forces the engineer to find a custom solution to make the template fit the CMS. The number (and impact) of discrepancies between the original prototype and the analysis of the engineer defines for a significant part the complexity (and success) of a project. If there is a poor fit between the prototype and the concepts of the CMS, you should rethink if this design (or CMS) is the right decision. A meta-model of this process is illustrated in 4.2. In Figure 4.3 we illustrated a fragment of a template that has been analyzed for development purposes. In this figure for instance the CMS comes with some functionalities out of the box: Navigation: generates an overview of a specific content type (usually defining the website structure); *SidewideNavigation*: generates the secondary navigation (list of secondary links to specific content); *PagePart*: generated a piece of reusable content that can be added to a multiple content types (such as logos, search box, login form); *PageTitle*: generates the main title of a content rendered in the content type 'page'.

After the templates have been developed, the functional implementation starts as defined in the requirements 'Custom Components', 'CMS Adaptations' [Souer et al., 2007b] and design phases 'Component Design' [Souer et al., 2008b].

This is the part where we address another CMS-specific challenge: for each functional or technical component, the software engineer has to decide if it is already *in the product*, if the functional component *has been developed in an earlier project*, or if it is a *new component*. This way, an optimal form of code reuse is ensured which makes the development process more efficient. The engineering first configures the off-the-shelf components as illustrated in 4.4. Typical examples are workflow, authorization, generic content types, multi-lingual, and navigation. Some of the components are available off-the-shelf but require a small customization. An example is a form module which requires some configuration to make a registration form (e.g. integrated with a custom CRM system). Those components are addressed in the complex route. After the off-the-shelf components, the engineer has to decide if the other components are custom made, or can be based on existing components. Some component based CMSs have a marketplace where you can find and share components as described in [Pereira dos Santos et al., 2009]. Based on what he has found, the engineer decides to use one of the existing components (Standard route) or have to develop a custom component (Complex route).

The complex route starts with determination of the risks, for example the dependancies. The detailed design as described in the 'Component Design' [Souer et al., 2008b] is analyzed and adjusted if necessary. The actual development consists of the engineering, unit test and registration of the code in the version tracking system. After all components have been developed, the integration test takes place to see if all the components work together as *designed*. The process ends with the functional testing of the system as well, in which the consultant checks if the system works as defined in the *requirements*. The Consultant and Architect approve the realization from a functional and technical view respectively. The realization phase is concluded and the resulting web application is ready for the Implementation phase: the deployment of the CMS on the production environment. A high-level meta-model of the realization phase is illustrated in 4.5.

4.4.2 The Implementation Phase

The goal of the implementation phase is a running web application on the production environment. The implementation phase starts with the transfer of the tested components to the production environment. The end-users of the CMS are then trained in how to use the system and how to add content. A final upgrade to the latest version of the CMS is followed by the acceptance test: the evaluation of the system in cooperation with the client in an environment that comes closest to the production environment [Kappel et al., 2006]. Most organization have a separate Development, Test and Acceptance environment next

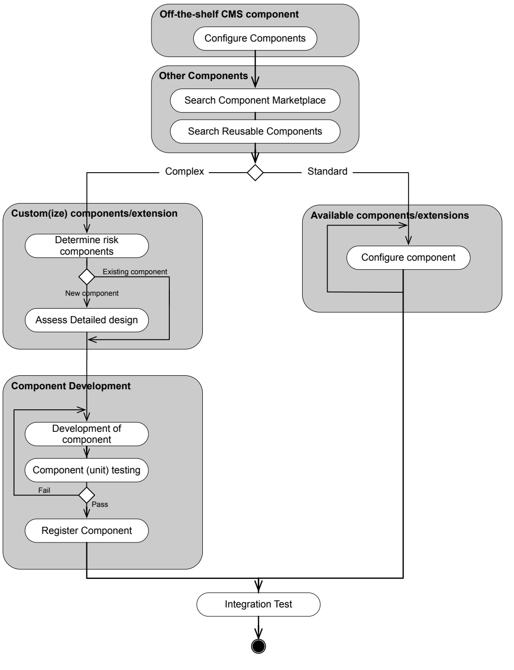

Figure 4.4: Development process of Components

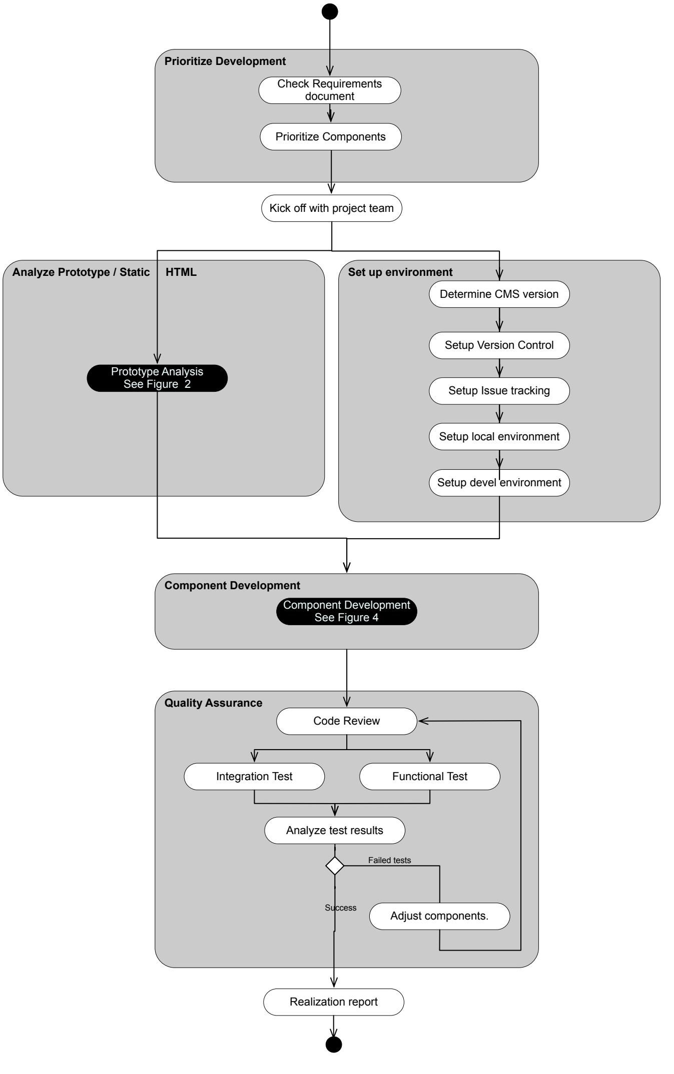

Figure 4.5: Meta-model of realization phase

to their Production environment. However, it is not uncommon to 'promote' the acceptance environment after acceptance to the actual production environment (and duplicate the newly formed production environment as the new acceptance environment). This is done since after all the effort that has been made to optimize the acceptance environment for production purposes, it would take some valuable time to repeat that process for the production environment. The acceptance test starts with the client starting to test according to the test plan. The time for the acceptance test is about a week. The issues that the client finds are examined and analyzed and categorized: *issue/bug* or *Change request*. When the acceptance test is over, and the customer agrees with all made repairs (and open issues), an acceptance contract is signed.

After the acceptance test comes a final quality assurance: the deployment assessment. The list assesses the CMS on a number of parameters defining the overall security, configuration. Parallel to the acceptance test and the deployment assessment, the client starts adding and managing the content for the web application – moreover, a lot of organizations find that the best way to test a system: actually working with it. Once these steps are completed, the website is made live, after which a warranty period of the website begins. During this period the original project team is still directly responsible for solving any problems of the system. Moreover the live website is monitored to see how it reacts on actual website visitors. The architect monitoring the CMS. To take into account the dual lifecycle of CMSs (one of the CMS-specific challenges), we introduce a certification process: the components which are made for this web application are certified. Certified components are guaranteed to work if the platform is upgraded to a newer version. The certification program for its components consists of a list of criteria with 3 levels of certification:

1. **Project specific**. The component is fully functional for a specific implementation of the CMS but may lack documentation, localization and may require specific hardware or software (e.g. only runs on MS SQL). The component may not be suitable for running on other instance of the CMS.

2. **Reusable**. The component is fully functional, provided with proper documentation but may require specific hardware or software as indicated in the packaged readme file. It may lack support for localization. The component is suitable to run on any installation of the CMS of the version the certificate is valid for.

3. **Product ready**. The component is fully functional, provided with proper documentation and supports at least the hardware and software required by the CMS platform itself. It supports localization and all documentation is

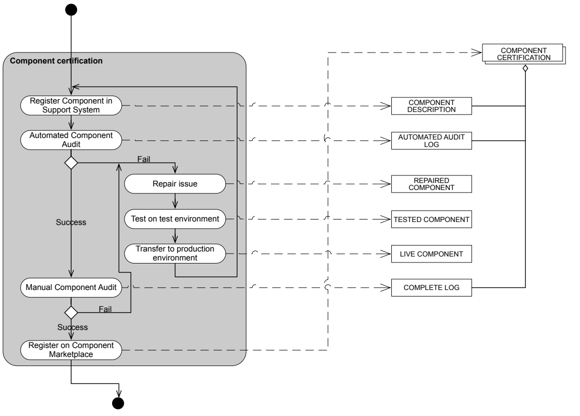

Figure 4.6: Certification Process

available. The component is of such high quality that it could be incorporated in the CMS platform as-is

Some example criteria are: 'The controller is a separate class and implements all controller logic.' (level 1), 'The component supports the software and hardware as described in the CMS Hard- and Software requirements for the appropriate version.' (level 2), and 'The HTML rendering the UI uses only the defined CSS classes.' (level 3). The certification process is illustrated in 4.6. The certification process is supported with an automated audit tool which assesses most criteria. After the automated audit, a manual audit is performed on the component. Once the component has passed the audit is registered on the component marketplace.

After the certification the project team evaluates the evaluation and the implementation with the customer. And that concludes the implementation phase. A meta-model of the implementation phase is illustrated in 4.7.

4.5 Discussion

We described the Realization and Implementation phase for the development of a CMS-based web application. Some reservations should be made with regards to the level of generalization. Moreover, the validation of the model should be made explicit. When trying to define the specific problems faced when implementing a CMS, the authors had one single CMS vendor as a reference case. Therefore the

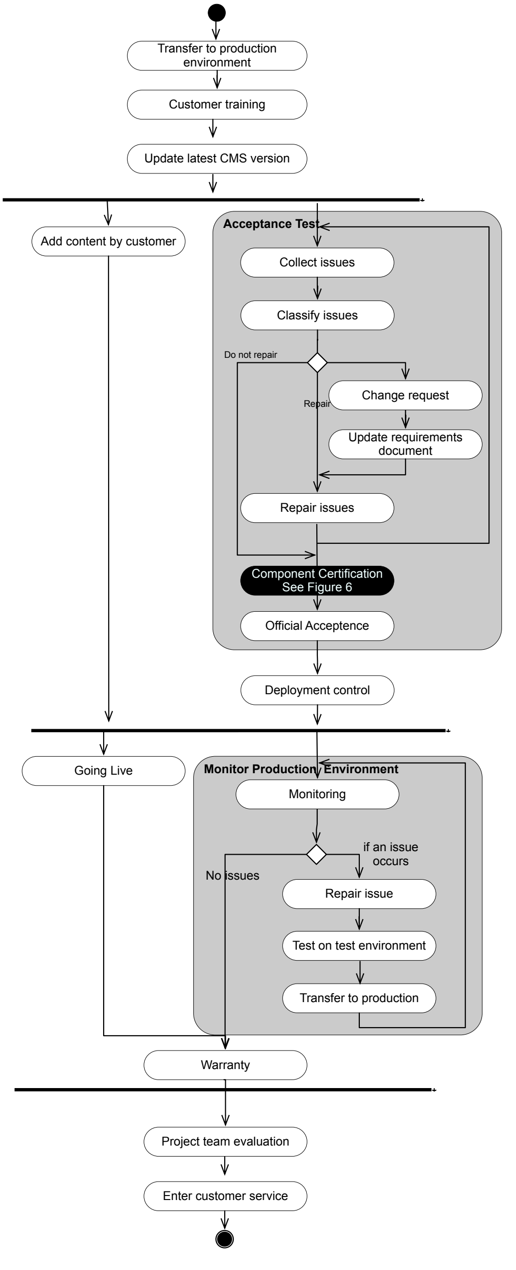

Figure 4.7: meta-model of implementation phase

problems defined here as CMS-specific are based on what the case study company has learned over the past 15 years. There is a possibility that the generalization of defined problems are limited. Similar, the proposed realization and implementation phase are validated in this one case study. Therefore, the defined solutions might be CMS-specific. A good validation over multiple CMS products and organizations is necessary to support this statement.

The research method that has been followed is design research. The artifacts that has resulted from the design cycles are the Product-Data Diagram of the Realization phase and the Implementation phase of the Web Engineering Method. We evaluated the diagrams and process description based on expert reviews: a number of experts working at GX were selected to study and review both the realization and implementation model. Project Managers, Architects, Consultants and Engineers were interviewed. The evaluation consisted of an open interview, combined with a questionnaire that the expert had to fill out after the interview. An important result was that the interviewees considered the realization and implementation models valid and applicable. All experts agreed that the model presented how it should go in the ideal situation. When discussing the certification of the components, experts acknowledged that there was some room for improvements. The idea to certify a component directly after it is build was received well. However performing both the automatic audit check and the manual check directly after building the component would probably take up too much time that the project team doesn't have at that moment. An expert also mentioned that in the future they are only going to work with the automated audit tool to certify a component. This means that the model can be used, where only the manual audit check has to be skipped.

4.6 Conclusion and Future Research

The realization and implementation method described in this research provides a complete overview of steps organizations can take while developing a CMS-based web application. The method has been evaluated with experts in a software company of a CMS. The method copes with three identified challenges which organizations will face when developing a CMS-based web application: Template analysis, Separation of customizations, ensuring platform upgrades. The methods propose a component based solution for the separation of customizations and a certification process to ensure future upgrades. Analyzing templates and mapping those prototypes to the CMS concepts provides an important indicator to measure the success of the project. Although we believe we provided a complete overview of the process of developing and implementing a CMS-based web application, further validation of the case in other organizations and other CMS' is necessary to make more generalized conclusions.

The work in this paper is part of our research toward supporting the development process of CMS-based web applications. As the next step, we are going to advance the current approach to address the maintenance and support of CMS-based web applications. Moreover, we are going to identify potential improvements of WEM on a 1) design level (with a model driven approach); 2) product level (based on product verticalization); and 3) process level (a process optimization). The goal is to provide researchers and practitioners with the right methods to improve the Web Engineering process.

4.7 Related Work

Schwabe and Rossi [Schwabe and Rossi, 1995], [Schwabe et al., 1996] introduced the Hypermedia Design Method (OOHDM). OOHDM comprises of four steps: conceptual modeling, navigational design, abstract interface design and implementation. These activities are performed in a mix of incremental, iterative and prototype-like way. In OOHDM a clear separation is made between on the one hand navigation and the other hand presentation. Koch et al. describe the UML-based Web Engineering (UWE) approach in [Koch and Kraus, 2002]. UWE is an object-oriented, iterative and incremental approach for the development of web applications. The development process of UWE consists of five phases: inception, elaboration, construction, transition and maintenance. Pastor et al. describe different methods with the Object-Oriented Web-Solutions Modeling approach. OOWS provides mechanisms to deal with the development of hypermedia information systems and e-commerce applications in web environments [Pastor et al., 2001]. OOWS strongly focuses on the generation of the required web application and less about managing the content and the application afterwards. OOWS comprises of two main activities: system specification and solution development. Ceri et al. describe in [Ceri et al., 2003] their Web Modeling Language (WebML), a notation for specifying complex websites at a conceptual level. In line with the definition of Web Engineering, the WebML approach consists of seven phases: requirements specification, data design, hypertext design, architecture design, implementation, testing and evaluation and maintenance and evolution.

CHAPTER 5

Operation and Maintenance Processes

Organizations increasingly utilize Web Content Management Systems (WCMS) to improve development speed, online flexibility and cost effectiveness for web applications. However, organizations lack the organizational processes and structure to effectively maintain WCMS. We propose a WCMS Process Framework for the operation and maintenance phase of Web Engineering. In this paper we elaborate on the construction of the framework. It encompasses the description of a generic IT Management framework and the inclusion of Web Content Management processes into a strategic, tactical, and operational level. The framework is validated through an expert validation consisting of three industry experts and a case study at a large Dutch telecommunications services provider. The case study substantiates our vision that the WCMS Process Framework contains a set of process descriptions that effectively supports the operations and maintenance of web applications. [1]

5.1 Introduction

Since the end of the nineties several web modeling methods have been developed, varying from E-R based to UML based, from conceptual to architecture design and

[1] This work was originally published in the Journal of Digital Information Management, entitled ‘A Framework for Web Content Management System Operations and Maintenance’ in 2008 [Souer et al., 2008a]. The work is co-authored with Paul Honders, Johan Versendaal, and Sjaak Brinkkemper.

from web to hypermedia application orientation. However, these methods focus on the design and creation of web applications from scratch rather than building web applications based on a platform with a focus on managing the information or content - of the web application. These web applications are known as Web Content Management Systems (WCMS) and are specifically designed to anticipate on the ever changing demand of Internet visitors [Souer et al., 2007b]. A WCMS can be defined as a group of business rules and editorial processes applied to content by people and organizations to align efforts of online publication with the business goals. With experiences of more than five hundred industrial implementations of WCMS software, we found that although WCMS help organizations from a technological point of view, a lot of organizations are struggling with the business processes surrounding the WCMS.

The research area we are addressing in this paper is Web Engineering which is defined as 'the application of systematic and quantifiable approaches (concepts, methods, techniques, tools) to cost-effective requirements analyses, design, implementation, testing, operation, and maintenance of high quality web applications' [Kappel et al., 2006], p.3. The existing research provides useful insight into Web Engineering in general but the operations and maintenance of web applications are underexposed. We present a WCMS Process Framework for organizing web content management which can be used in conjunction with existing frameworks such as ITIL, ASL and BiSL ([Barafort et al., 2002], [Larsen et al., 2006], [Looijen, 1998], [van der Pols, 2004], and [van der Pols, 2007]).

The WCMS Process Framework which we presented in [Souer et al., 2007a] detailed the processes of the operations and maintenance phase of web applications. In this paper we will re-address the foundation of this framework in more detail. We will provide more insight into the integration of our framework with the complementary frameworks as well as another case description. Furthermore, we added an expert validation based on three industry experts and compare our research to related research.

The paper is structured as follows. The next section discusses current issues in operation and maintenance within the field of web applications. We then give an overview of a generic IT Management model which is the foundation of our WCMS Process Framework and describe how our WMCS Process Frameworks is integrated in the generic model. In section 3 we describe the validation of the framework through an expert validation and new case study. Section 4 elaborates on related work. We end this paper with some conclusions and point out future research.

5.2 A framework for Web Engineering operation and maintenance

Our WCMS Process Framework is based on the assumption that web engineering is different than traditional IS development. The authors in [Kautz and Nørbjerg, 2003] and [Holck, 2003] critically examined these differences and concluded that web development methods are not radically new but merely extensions or variations of fundamental dynamics which have characterized information systems since the inception of the discipline. There may not be one single characteristic which is unique to web development; the collection of characteristics definitely is [Baskerville and Pries-Heje, 2002]. Methods proven for traditional information systems should however not be disregarded. Our WCMS Process Framework therefore uses existing models when possible and only differentiates where specific web related issues arise. In [Souer et al., 2007a] we detailed these web specific issues: 1) identifying the user groups and the fit with their needs and requirements; 2) definition and implementation of a content maintenance strategy; 3) keeping the content valid, accurate, current and complete; 4) coordination of operational content management activities; and 5) management of external content providers.

To cope with the described issues, we developed our WCMS Process Framework, based on Looijens existing model for organizational IT Management [Looijen, 1998]. In Looijens threefold model of management, control and maintenance of Information Systems, three management units are distinguished: Functional Management, Application Management and Technical Management. Each managerial unit is composed of three managerial levels; strategic management, tactical management and operational management levels.

Although the Functional, Application and Technical management of WCMS software can be covered by Looijens IT Management Model the management of the content of the website itself is different. This is acknowledged by the authors in [Kautz and Nørbjerg, 2003] where they argue that managing Web Information Systems (WIS) is not comparable to managing non WIS. There seems to be a gap in the existing model. To bridge the gap we add a Content Management Unit focusing specific on the content of the web application. Consequently with the addition of the content management unit, we should be able to give consideration to the overall management of WCMS applications.

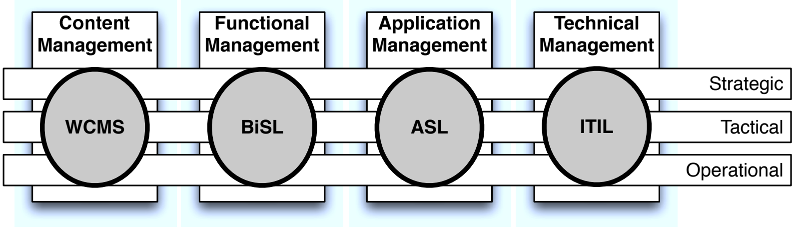

Figure 5.1: Overview of Frameworks

5.2.1 Integration WCMS in the IT Management Model

Over the last 20 years, process frameworks based on best practices have been developed, each addressing a specific part of Looijens IT management model. The most comprehensive structured approach available is probably the Information Technology Infrastructure Library (ITIL) [Barafort et al., 2002]. Based on ITIL, more specific descendants like ASL (Application Service Library) and BiSL (the Business Information Services Library) evolved, covering the Application Management and Functional Management respectively as described by the authors in [van der Pols, 2004] and [van der Pols, 2007]. However, no framework addressing Content Management processes exists. We therefore created our own WCMS Process Framework.

Figure 1 provides an integrated overview of the complete IT Management Model with ITIL, ASL and BiSL covering the technical, application and functional management respectively. Each of these models explicitly addresses the strategic, tactical and operational level. Our WCMS Process Framework should therefore cover the specific content management related issues and should encompass strategic, tactical and operational level. With a certain framework, we create an extended IT Management model for Web Engineering.

Basically the three IT management domains can be viewed upon as respectively the demand and supply side of IT within the business. Viewing the three managerial units in perspective, functional management manages the demands of the user organization. Application and Technical management are the service providers: they deliver the services (application-based and technology/hardware-based respectively) as issued by the functional management unit. With the added Content Management there is a new chain of demand and supplies. In the new situation the end user (or website visitor) creates the demand which should be supplied by the content management unit. The content management unit utilizes the WCMS as an information system to support the process of managing content

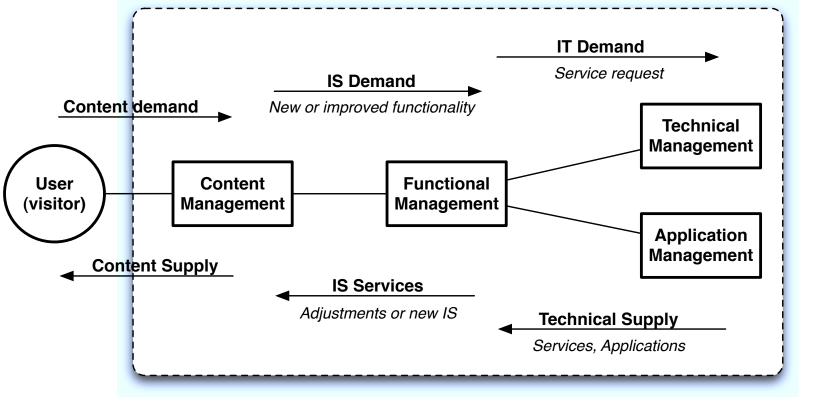

Figure 5.2: Integrating Content, Functional, Application and Technical Management

and is now serviced by the functional management unit whenever improved functionality is needed. Functional management translates their requirements into demands. An example of a functional requirement is that the website should at least be able to cope with 5 million page views a day. That would be translated into demands to application management (e.g. customizations, CMS-architecture and database-clustering), and technical management (i.e. infrastructure, number of servers, failover servers, cpu power, RAM). The system architect translates the requirements into application management and technical management demands.

5.2.2 The WCMS Process Framework

With the differences and specific nature of WCMS we developed the WCMS Process Framework. An overview of the WCMS Process Framework is visualized in Figure 3. We elaborated on each of the processes in [Souer et al., 2007a]. The framework copes with the earlier raised issues and can be used next to the ITIL, ASL and BiSL models. The WCMS Process Framework conformed to the subdivision of three managerial levels; the strategic, tactical and operational level. This also complies with the five identified issues mentioned earlier where strategy and policy issues indicate a strategic management level, coordination issues point to a tactical management level and the production and maintenance of content designate an operational level. The WCMS Process Framework is developed, based on existing web and content related literature and the

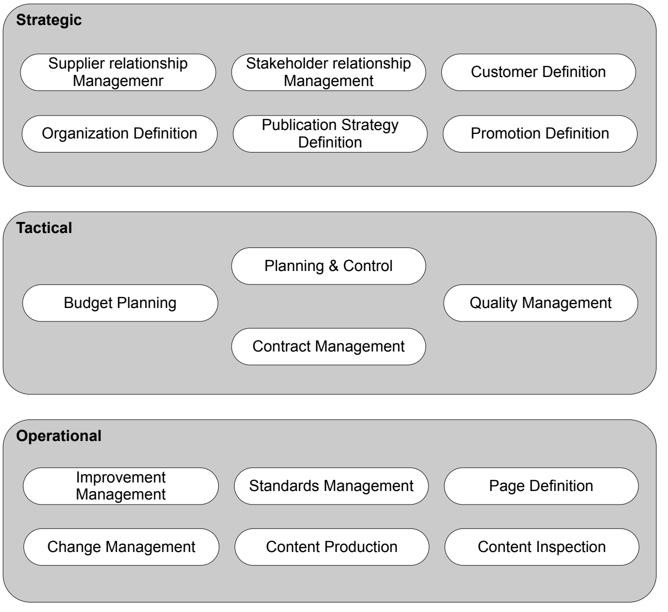

Figure 5.3: WCMS Framework Overview

results of a web management study. Additional content management specific processes were designed for issues not covered by ASL and BiSL to complement the WCMS Process Framework, using other references [Fuller et al., 2004], [Kappel et al., 2006], [Lee, 1999], [van der Pols, 2007], and [Schwickert, 2004].

Within each level, processes are designed consisting of relevant activities. In order to reveal the relations between activities (the process) and artifacts (the data produced in the process) meta process delivery diagrams (PDD) are used. The PDD technique is based on a UML activity diagram, reflecting the activity side, and a UML class diagram which reflects the deliverables side. The UML semantics are strictly applied, however, some primitives have been added in order to deal with composition issues. For detailed information about the technique, we refer to as proposed in [van de Weerd et al., 2006]. We detail one process to illustrate how a rationale of a process is built up and how a process is presented and explained: the Content Inspection process, which is part of the operational processes.

5.2.3 Detailing the Content Inspection Process

While the structure and presentation of a web application are subject to change request, they are generally kept stable over a lengthy period of time [Kappel et al., 2006]. For the content itself, the opposite is true. Content of most web applications are subject to permanent changes which impacts the contents volume, currency, consistency and reliability [Ginige and Murugesan, 2001a]. According to Ebner et al. [Kappel et al., 2006], this means that various update cycles have to be considered, depending on the web application type. By nature, update cycles of news oriented web applications are short. These update cycles are executed as soon as information changes or hourly. However, a web application concerning a company presentation will have longer update cycles. In order to ensure the content remains up-to-date it should not only be reviewed before publication but should be reviewed regularly [Fuller et al., 2004]. A process monitoring the content currency throughout the web application and deciding when and what content should be updated could ensure the currency of the available content of the web application. According to Schwickert [Schwickert, 2004] the initial review should be done by a hierarchically super ordinate instance. For the periodic review, Fuller more concretely mentions a supervisor, an executive, a web council or a Public Affairs officer as examples [Fuller et al., 2004]. Fuller further notes that retired content should be retained in an archive for reference purposes, or as Schwickert notes, to serve as a source for content re-use. Also with legislation concerning archiving, it is a necessity to keep track of the content. However, in his content lifecycle model, Schwickert also added a Destroy process to dispose of content [Schwickert, 2004]. The next illustration provides an overview of the activities and the deliverables of the content inspection process. The three main activities are Content Inspection, Request Update and Completion. As can be seen, two activities contain multiple sub-activities. The deliverables content and archive guidelines provide the necessary input for the activities. One possible outcome is the ‘request for operations’ deliverable. With this schema, we can model and communicate the activities and deliverables.

Each activity and sub-activity is then detailed in a table, providing a textual description per sub-activity. A breakdown of the activities and sub activities of the Content Inspection Process is provided below.

The content inspection is usually done by a (chief) editor supported by the WCMS: the system notifies the editor when a page needs to be checked. The inspection itself - judging the actual content - is a manual process. Completion is again supported by the WCMS: with the archiving or delete option within the WCMS, the chief editor or his delegate can remove the content from the website.

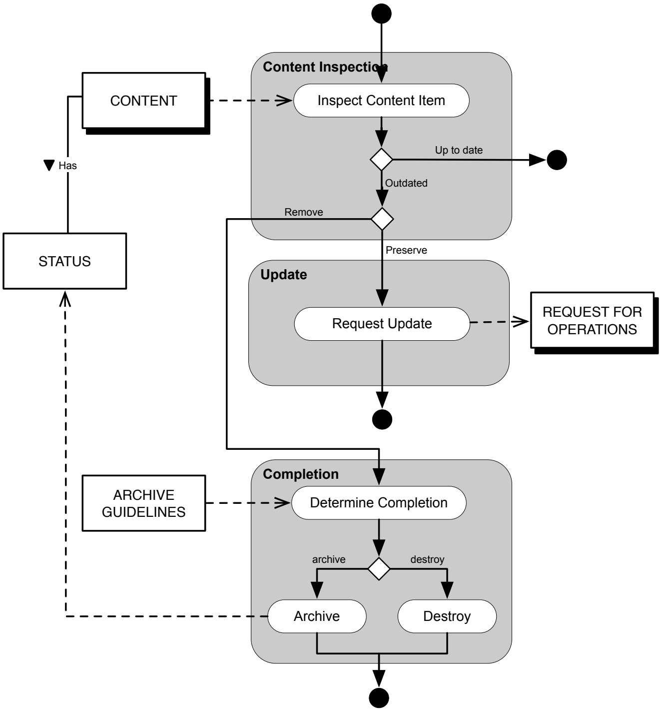

Figure 5.4: Content Inspection Process

Activity	Sub-activity	Description
Content Inspection	Inspect content item	Inspect the published CONTENT ITEM(s). By CONTENT ITEM is meant the page, article or other content (banners etc.) placed on the website.
	Outdated branch	Determine whether the CONTENT is outdated or up-to-date. When the CONTENT or parts of the content are outdated the next branch is triggered.
	Preserve branch	Determine whether to preserve or remove the outdated CONTENT.
Update	Request update	When the content is outdated but should be preserved an update is necessary. This results in a REQUEST FOR UPDATE. A practical implementation of this activity could be to depict the content item and to attach comments in order for the assigned or responsible author/editor to know what the purpose of updating this CONTENT ITEM is.
Completion	Determine completion	When content which is outdated and is not desired to be preserved, a decision needs to be made about the way it is completed. In general two options are available; either retaining the content in an archive or destroying it by removing the actual content. Guidelines for determining the completion are recorded in the ARCHIVE GUIDELINES.
	Completion branch	*Here the actual choice is made whether to archive or destroy a certain CONTENT ITEM.*
	Archive	Here the activities necessary to archive the CONTENT ITEM are located. Within a CMS-based Web application most of this process is done automatically. However, the STATUS of the CONTENT ITEM has to be set to an archive status in order to trigger this automatic process. Archiving in a CMS-based Web application is mostly done by versioning.
	Destroy	Another activity which can be triggered by the completion branch is the destroy activity. Here the activities necessary to perform a physical removal of the CONTENT ITEM are located. The actual destruction of removal itself is done automatically by the CMS-based Web application.

Figure 5.5: Activities and sub activities of Content Inspection Process

5.3 Empirical Validation

Following [Robinson, 1997] we carried out an expert validation and a case study in order to determine the usefulness and correctness of the WCMS Process Framework.

5.3.1 Expert Validation

Within the expert review a number of experts of GX, a medium-sized WCMS vendor, were selected to study and review the WCMS Process Framework. GX implements web-applications based on its WCMS GX WebManager as described in [van Berkum et al., 2004]. As the framework consists of strategic, tactical and operational process levels three experts were selected based on their knowledge on managing web content and their experience with WCM on these particular levels. The resulting selection consisted of a senior consultant, a project manager and an online marketer (strategic, tactical and operational level respectively). To evaluate the validity of the WCMS Process Framework, test-criteria were defined using the validation methodology presented by Beecham, Hall, Britton, Cottee and Rainer in [Beecham et al., 2005]. Resulting test-criteria are scope, comprehension, ease of use and tailorability. Validating against these criteria also addresses our research question, in which we indicated to search for an effective approach for organizing WCM: we define effectiveness as dependent on the four test-criteria. A questionnaire was designed to measure these criteria. The experts were provided with a full explanation of the WCMS Process Framework and its processes and activities separately. After this explanation individual interview sessions were scheduled. Similar to the method of Beecham et al. [Beecham et al., 2005] we provided a 5-point scale for the possible answers on the statements and questions (ranging from strongly disagree to strongly agree). Additionally, we requested the respondents to comment critically where possible. The results of the expert validation are grouped according to each test criteria and discussed briefly.

Scope.The respondents agreed that the framework could be used as an initial guideline for WCM in organizations (2 agreed and 1 strongly agreed). One respondent remarked that the process description included in the framework incorporates quite a level of detail compared to what Ive read on this subject so far. The respondents also agreed that the level of detail within the processes should be sufficient to guide users/practitioners of the model to recognize these processes in practice (1 agreed and 2 strongly agreed). Concerning the completeness of the framework they all agreed that the WCMS Process Framework should cover all relevant processes. Subsequently the respondents were asked to select the key processes of WCM. Unanimously they identified the content production process as the key process of WCM. Other key processes mentioned were the publication strategy definition and the customer definition process. For the scope of the model we can conclude that the WCMS Process Framework can offer a solution for the WCM issues. Users or practitioners of this framework however should be aware that the applicability of each process should be assessed and evaluated for each organization.

Comprehension. On the comprehension the framework the respondents were all positive (all ranging from agree to strongly agree). They indicated that each process was easy to understand (1 agreed and 2 strongly agreed), clearly defined (all noted agree), clearly presented (all noted agree) and the division between strategic, tactical and operational processes was found easy to understand (all noted agree). According to one respondent the differentiation between strategic, tactical and operational processes are logical but he mentioned that this would depend on the size of an organization. For example, in smaller organizations these processes are executed by just a couple of people which therefore make it difficult to clearly distinguish between strategic, tactical and operational levels.

Ease of use. The framework provided enough information to make it easy to interpret it (2 agreed and 1 strongly agreed). However, one respondent mentioned that some previous theoretical knowledge or experience in practice is needed to be able to use all processes. According to this respondent the explanations are clear but will raise some questions at less knowledgeable readers when they try to create a direct link with the processes described. As another respondent mentioned more examples would benefit each process.

Tailorability. The respondents were asked whether the processes and their activities are general and most likely to apply to most companies. On the applicability of the processes two respondents agreed and one respondent strongly agreed. One respondent noted that 'only in large organizations people work according to procedures which are defined top-down. To adapt these procedures in the WCMS Process Framework, even the smallest change could be very hard'. Another respondent stated: 'the approach might be to overwhelming for smaller organizations'. As mentioned the framework can be applied to most organizations. Smaller organizations could also benefit from the framework but should evaluate each process on its usefulness and applicability just like the BiSL and ASL frameworks are used [van der Pols, 2007]. Implementing a process framework for WCM could require organizational changes.

5.3.2 Case study

Yin [Yin, 2002] describes four validity types that should be taken into account when establishing case studies: 1) construct validity; 2) internal validity; 3) external validity; and 4) reliability. With regard to the construct validity we interviewed two respondents and introduced the WCMS Process Framework beforehand to avoid discrepancies. The addressed the internal validity by explaining the 'how' and 'why' of each process and its activities. Further, explanation-building occurs for each statement through a thorough clarification. We externally validated the WCMS Process Framework by means of an expert validation [Carson, 1986]. Future case studies will provide us with more insight into the frameworks external validity. The reliability is provided through the case study with the use of the RASCI method to assess each process and its main activities as defined in the Control Objects for Information & related Technology (COBIT) [Institute, 2007]. The RASCI method identifies the roles and responsibilities in an organization by identifying who is (R)esponsible, (A)ccountable, (S)upportive, (C)onsulted and (I)nformed. Since the COBIT framework can be regarded as an auditing instrument, we adopted this method in order to identify the roles and responsibilities used in practice for each of the processes and sub processes of the WCMS Process Framework.

In [Souer et al., 2007a] we described a case study of a Dutch inter-municipal cooperative union in which municipalities of a city region cooperate in order to promote common interests. We performed a second case study at a large Dutch telecommunications services provider. For their business customers they provide services ranging from voice, Internet and data services to full-managed outsourced IT solutions in The Netherlands, Germany, and Belgium. The organization serves 2.2 million Internet Customers in the Netherlands. Their WCMS was initially designed to provide self-service functionality to their customers via the web (cost reduction). Currently sales incentives are the main reason for having a web presence and their WCMS is an important distribution channel. The case validation included a business consultant of the web department. Presented below are the validation results of the case validation at the organization.

It became clear that all strategic, tactical and operational processes were present at the organization. A sample of the RASCI table is printed in table 2. The observation of the data reveals that strategic WCM activities are somewhat scattered throughout the organization. For instance the Organization definition process, the board of directors is both responsible and accountable and the employees council is consulted. This indicates that WCM activities are interwoven into the corporate structure. On the tactical level the Manager operations is responsible for the Budget planning, the marketing manager is accountable. This indicates that budgeting for WCM activities is a marketing expense due to the main sales incentives of the WCMS. For the management of suppliers the organization has appointed a supply chain manager. This role executes the tasks of the contract management process. This role was also supportive in the strategic process of Supplier relationship management. On the operational level, the following observations were made: for producing content and creating pages, two different roles are distinguished; the content manager and the content specialist. This is a similar hierarchy between an author and an editor. The content manager creates articles (pieces of content) to be used on the different pages. The content specialist creates and maintains the pages, reviews the content produced by the content manager and takes care of the inspection of the content later on. Marketers support the production of content and ultimately the respective product manager (i.e. phone, internet, etc.) is consulted and informed about the results. Standards are developed and tested by respectively the developers and testers. Business consultants however are accountable for these developments. Solution and technical architects are responsible defining, organizing and specifying improvements. The manager of operations is ultimately accountable and informed concerning these improvements.

Compared to the case study at the non-profit organization (presented in [Souer et al., 2007a]), this case supports the WCMS Process Framework. Due to the strategic importance of their WCMS, managing web content professionally is also regarded of importance. This could be the reason why the framework was considered complete, as the case study organization did not execute activities that

		Board of Directors	Employee council	Corporat Communications	Product Managers	Manager Corporate Web site	Manager Marketing	Formula Manager	Marketing Manager consumer/business	Traffic Manager	Manager operations
Organization Definition	*Define organization*	RA	C			S					
	Manage organization	RA	C			S					
	Evaluate organization	RA	C			S					
Publication Strategy Definition	*Corporate Strategy Alignment*			S		A		R	C		
	Formulate publication guidelines						A	R	C		
Promotion Definition	*Define promotion strategy*				S			A	I	R	

Figure 5.6: Sample of the Strategic Processes Validation

were not part (or could not be recognized) in the framework. However, the remark was made of secondary processes of the overall web management, like transactions and logistics of the products and services offered through the Web, could benefit a future version of the framework. Moreover, the processes Budget planning, Quality management and Improvement management process were not regarded as useful for the non-profit organization but are identified within this second case study. These processes might not be as useful for non-commercial organizations but could be for commercial organizations.

5.4 Related work

The WCMS Framework is part of the Web Engineering Method (WEM) [Souer et al., 2007b]. In the field of Web Engineering, there are several research groups working on related work. We briefly elaborate on three relevant research groups ([Ceri et al., 2003], [Koch and Kraus, 2002] and [Pastor et al., 2001]). Ceri et al. describe in [Ceri et al., 2003] their Web Modeling Language (WebML), a notation for specifying complex web sites at a conceptual level. The WebML approach consists of seven phases: from requirements specification to maintenance and evolution. They acknowledge the need for maintenance, but do not elaborate on it on a process level. WebML is a model-drives approach where change requests are translated to the data-model or the hypertext model. An overview of the processes from an information management perspective is underexposed. Koch et al. describe the UML-based Web Engineering (UWE) approach in [Koch and Kraus, 2002]. UWE is an object-oriented, iterative and incremental approach for the development of web applications. The development process of UWE consists of five phases, including maintenance. The development process of UWE is based on the Unified Process (UP) and tailored towards web application development. The development process of UWE consists of five phases: inception, elaboration, construction, transition and maintenance. Within these phases a variable number of iteration workflows take place: risk management, iteration management, iteration evaluation, requirements capture, analysis and design, implementation, validation, verification and testing. We concluded that UWE is more similar to our framework since it acknowledges the importance of maintenance of web applications. It also has certain workflows which correspond with our tactical and operational processes. However, Koch and Kraus neglect the strategic processes and indicate that the main UWE modeling activities are: requirements specification, conceptual, navigation and presentation design. Pastor et al. describe different methods with the Object-Oriented Web-Solutions Modeling approach (OOWS) being the closest one to our method. OOWS provides mechanisms to deal with the development of hypermedia information systems and e-commerce applications in web environments [Pastor et al., 2001]. OOWS strongly focuses on the generation of the required web application. OOWS comprises of two main activities: system specification and solution development. Within the system specification activity the functional requirements are specified by the creation of five models which form the basis for the second activity, solution development, wherein the actual web application components are generated. Within solution development the transformation from the conceptual schema to a software product is realized. OOWS provide some useful insight into the modeling and development of web applications however the organizational processes to maintain the generated web application is not provided. The main contributions of WEM consist of a set of tools, process descriptions and guidelines to create, organize and maintain content driven web applications. The contribution of this research is the WCMS Framework as proposed in this paper about the processes supporting the maintenance of Web Content Management activities.

5.5 Conclusion and Future Research

In this section we presented a process framework for Web Content Management Systems. We used fragments from existing models and combined them into our model. The framework consists of a set of guidelines and process descriptions separated over three managerial levels: strategic, tactical and operation. Each process is described by means of Process Delivery Diagrams and descriptive tables. We validated the model through an expert review and a second case study. Based on the two case studies we are even more convinced that the WCMS Process Framework could act as a useful set of guidelines for organizations to organize WCM processes. Some remarks were made however of the absence of secondary processes and three processes in non-profit organizations. We trust that we have made a step forwards in developing a method for the implementation and maintenance of WCM systems. Furthermore, we think that the framework of WCM processes is useful beyond the scope of Web Content Management and could be applicable in other content driven areas (e.g. publishers). We are currently researching the influence of user generated content and community technologies on existing organizational processes, especially the implications on information control and security. We also continue our research to extend WEM with a modeling language to model the processes in the web application itself

Part III

Process Innovation

Chapter 6

A Model Driven Development Approach to Configure CMS-based Web Applications

With the growing use of Web Content Management Systems for the support of complex online business processes, traditional implementation solutions proofed to be inefficient. Specifically the gap between business requirements and the realized web application should be closed. This paper presents the development of a modeling tool for the automated configuration of Web Content Management Systems (WCM) which aims to reduce the complexity and increase the transparency of implementations. It allows business users to configure the business processes without the need of technical support. We combine fragments of existing Web Engineering methods and specify an abstract and concrete syntax based on a domain model and end user analysis. The resulting WebForm Diagram has been implemented in a prototype and has been validated by subject matter experts. This research is part of a bigger project to develop the Web Engineering Method (WEM) which provides an overall method towards a full coverage of the specification, design, realization, implementation and maintenance of WCM-based web applications.[1]

[1]This work was originally published in the Proceedings of the 9th International Conference on Web Engineering (ICWE), entitled "Model Driven Web Engineering for the Automated Configuration of Web Content Management Systems" in 2009 [Souer et al., 2009]. The work is co-authored with Thijs Kupers, Remko Helms, and Sjaak Brinkkemper.

6.1 Introduction

The World Wide Web has evolved towards a platform for sophisticated enterprise applications and complex business processes. In result, the effort of time specifying and developing these applications reflects the complexity of these applications and business processes. An industry solution to improve development time and stability is a Web Content Management (WCM) system which is product software with out-of-the-box functionalities and allow for specific customizations [Souer et al., 2007b].

Customizing WCM systems to implement business processes is a difficult task. In this context we developed the Web Engineering Method (WEM) as a method to manage and control web applications and websites based on WCM systems [van de Weerd et al., 2006]. WEM describes a complete development and implementation process of Web Engineering based on WCM systems as defined by Kappel et al [Kappel et al., 2006]: requirements analyses, design, implementation, testing, operation, and maintenance of high-quality web applications [Souer et al., 2008a].

The central problem we are addressing in this paper is that there is a gap between the requirements analysis in WCM implementations (usually defined by business users or online marketers) and the realization of those requirements (usually done by technical developers or software engineers). Therefore our leading research question is: 'how to automatically configure a WCM System based on requirements'. Note that we do not try to create a WCM system, but configure an existing WCM for the development of Web Applications. We focus on the configuration of business processes and defined the following goals:(1) Develop an unambiguous form definition to model a business process; (2) Automatically configure a WCM system based on the business process model; and (3) Generate a business process model based on a configured WCM system. This paper presents a new approach for an automated configuration of WCM systems using a modeling tool. The contribution of this research consists of a Model Driven Engineering (MDE)approach for the configuration of WCM software and a unique concrete and abstract syntax to model WCM systems. Secondly, we implement the MDE specification as a prototype and used it to model an actual project situation.

The rest of this paper is organized as follows: In Section 2 we introduce the WebForm Diagram which is the result of this research. Section 3 elaborates on the problem are analysis which resulted in the WebForm Diagram including a domain model, a user analysis and a comparison of existing web engineering methods. In section 4 we formally specify the WebForm Diagram and describe the development of a prototype which is evaluated in section 5. Finally conclusion and future work are discussed in Section 6.

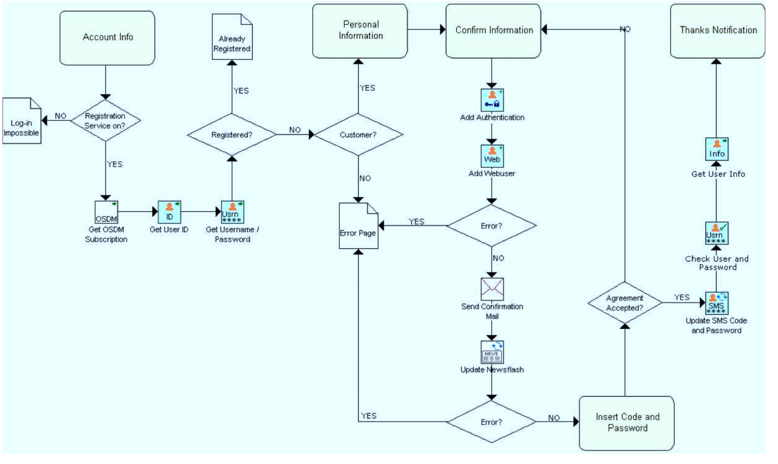

Figure 6.1: Example of a business process modeled with the WebForm Diagram

6.2 The WebForm Diagram

The WebForm Diagram is a visual language for the specification of online business processes. It is developed to cover all online form variables within a web-based application.

As an example of the realized WebForm Diagram we refer to Figure 6.1 that models a user registration scenario from a real life case: it illustrates how a new user can enter his account information, some personal information and after validation by e-mail and text messaging he is allowed to enter the registered area. However, there are some conditions which the system and user should meet and during the process some system processes and database access are initialized. The purpose of this research is to allow functional users to model such a form and automatically configure a WCMS to support this process. We detail this form in the following sections.

6.2.1 Steps, Routers, Validation, Handlers

In the WebForm Diagram, the screens of the form dialogue which are actually presented on a website are called **steps** and are visualized with rounded rectangles. A step can have multiple *formfields* such as text input, list item, checkbox, etc. A typical first step of a user registration process has two formfields: desired username and password. The form displayed in figure6.1, has 5 steps starting with Account Info. Individual formfields are not displayed in this top level view of the WebForm Diagram. However, the diagram allows users to detail steps.

Steps are connected with **routers** which are visualized by arrows between steps. After each step, a form is routed to another step. In the form in figure 6.1, after the user enters his username and password in the 'Account info' step the system will try to route him to the step 'Personal Information'.

The flow of the routers depends on certain conditions which are known as **validations**. These validations are visualized with a diamond. For example, the user can only register when the registration service is set to 'on' which is the first validation after step 'Account Info' in figure 6.1. Otherwise he will be routed to the web page 'Log-in Impossible'.

Between steps, the WebForm Diagram can perform actions which we call **handlers** and are visualized with a small square containing an icon inside representing the functional behavior of the handler. A handler can for example access a database to check the user id, send a confirmation e-mail or check the user credentials with an authorization server by SOAP.

Hence a complete online business process is a logical flow of steps which are routed in the correct way, validated input fields and correct handling of the data.

6.2.2 Formalizing the WebForm Diagram

To develop our WebForm Diagram, we formally specified the syntax to define the logic. The WebForm Diagram relies on the Model Driven Engineering paradigm [Kent, 2002]. MDE is in accordance with the objectives of this research: configuration of a software system based on a model [Schmidt, 2006]. MDE uses models as a primary artifact in the development process and consists of a Domain Specific Modeling Language (DSML) and a transformation mechanism [van Deursen et al., 2000] reflecting the structure, behavior or requirements of a given domain and transforms the model into different implementations respectively.

A DSML is a graphical and executable specification language that is specialized to model concepts within a given domain by means of notations and abstractions. To develop our WebForm Diagram we therefore analyze the problem area with the following three activities: (1) what needs to be modeled, using domain modeling [Kang et al., 1990];(2) what is the expected outcome, with user analysis as defined by [Cooper, 2007b] resulting in a mental model, a representation model and an implementation model; and (3) Comparing concepts of existing models to identify key concepts which we could use in

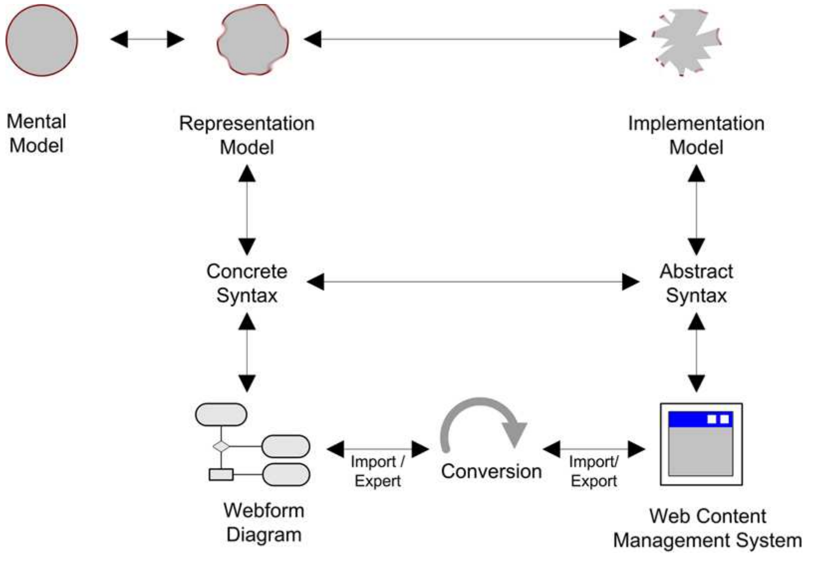

Figure 6.2: Research Approach

our model. Similar to Situational Method-Engineering [Brinkkemper, 1996], [Ralyté et al., 2007], we selected method fragments from existing modeling languages and assembled our new modeling language out of these fragments.

The DSML in this research consists of a comprehensive abstract syntax and an understandable concrete syntax with a graphical notation [Group, 2005]. The concrete syntax resembles the representation model of the end user and the abstract syntax represents the objects which need to be configured in the WCM System. We developed the model as a prototype based on the graphical notation of the concrete syntax which we resulted in the WebForm Diagram. The prototype is developed in MetaEdit+ since it allows us to build our own specific development environment. The definition can be exported into an XML format and is converted into the WCMs compliant XML which by itself can be imported into the WCM system. An overview of the approach is illustrated in Figure 6.2.

The evaluation consists of both a functional assessment in which we used the prototype tool to define a real project situation, and an expert validation in which we interviewed end users.

Figure 6.3: Excerpt of Domain Model

6.3 Problem Area Analysis

This research has been carried out within GX, a software vendor of Web Content Management software. GX develops GX WebManager, a Web Content Management System based on open standards such as OSGi as implemented in Apache Felix, Spring MVC and the Java Content Repository by Apache JackRabbit.

6.3.1 Domain Model

We use a domain model to analyze the elements and their relationships. Domain Modeling helps identify the key concepts which need to be modeled as well as generalizations which relates the entities on a higher abstraction level and is a meta-model of the objects of the modeling language. A relevant functional component within the WCMS – in the context of this research – is the 'Advanced Form Module': a functional component which allows editors to develop business processes based on advanced forms. Using a domain model, we identified all the elements that need to be modeled by the WebForm Diagram and is the foundation of the abstract syntax.

6.3.2 User Analysis

In the next step we identified the expected model by interviewing end-users using a threefold model as proposed by Cooper [Cooper, 2007b]: **Mental model** providing the system, its components and the way they interact from an End-user (or functional) perspective; **Implementation model** reflecting the actual system implementation composing of all components, their relationships and the way they interact; and **Representation Model** representing the implementation to the end-users. One should try to create a representation model as close to the mental model as possible [Fein et al., 1993]. Within a modeling language, the representation model consists of the graphical notation.

We interviewed six users (3 senior technical architects and 3 senior functional consultants). All six users defined a form as a set of *Steps* on the one hand, and a *Flow* defining the order of the steps on the other hand. They visualized it as a set of square blocks (the steps) with arrows in between (representing the order of the steps, or 'flow'). After they defined the steps and the flow on a high level they started to specify the steps in detail. In other words: the users expected at least 2 levels of abstraction: (1) high level defining the steps and the flow between the steps, and (2) a more detailed level specifying the contents of the step. Each step has multiple input fields from a certain type (e.g. text field, password field, radio button, etc), buttons and information. Two architects suggested adopting Business Process Modeling Language (BPMN) as a solution for the modeling language especially since WCM systems are often integrated into other systems and BPMN is a well known standard for defining processes [Smith, 2003] and used within Web Engineering [Brambilla et al., 2008]. When a router was conditional, a diamond was suggested with multiple outbound lines. The users did not have any clear ideas how to cope with database connections and handlers except for the idea of using an object to define that there should be a handler on that specific place. With the user analysis we identified the mental model of the end users.

6.3.3 Comparing existing models

In this section we describe a comparison and selection of web application modeling approaches in order to fill a method base. We continue with our previously filled method base which we gathered using Situational Method-Engineering [Ralyté et al., 2007] , consisting of the following approaches: Object Oriented Hypermedia Design Model [Schwabe and Rossi, 1995], WebML[Ceri et al., 2003], UML-based Web Engineering [Koch, 1999] and Object Oriented Web Solutions [Pastor et al., 2006].

In [Luinenburg et al., 2008b] we compared existing methods with a comparison matrix. In this research additional requirements were gathered which resulted in an adjusted comparison matrix. We compare existing models based on the Cooperative Requirements Engineering With Scenarios Framework (CREWS) [Rolland et al., 1998] as it has been successfully used for classification of software [Rolland and Prakash, 2000], [Soffer et al., 2003]. The adapted CREWS framework classifies modeling methods by four views: Content, Form, Purpose, and Customization process. Each view has a set of associated facets, and each facet is characterized by a set of relevant attributes [Soffer et al., 2003]. We adapted the CREWS framework with similar views, adding a domain view as a separate view. Each view has different aspects describing the properties of the modeling language which we compared. **Content view**: analyzes which existing knowledge is being used within the model; **Form view**: identifies the structure and the graphical notation of the modeling language; and **Domain view**: compares the entities within the domain with the modeling concepts of the modeling language. We detail the Content View. The Content View analyzes to which extend knowledge is being used. It is analyzed on the following three aspects: (1) *Abstraction*, (2) *Context* and (3) *Argumentation*.

Abstraction has one attribute reflecting the possibility of an approach to incorporate different abstraction levels. There are three possible values for this attribute: Flexible (multiple levels possible), Fixed (multiple levels possible, but when a level is chosen it cannot be changed) or Unique (only one level). In our case, the preferred value is 'Flexible' since the user analysis asked for a multiple detailed abstraction.

Context consists of three operators: Internal (model the internal system), Interaction (model the interaction between the system and its environment) and Contextual (model the environment and its context). In our case, the preferred values are 'Internal' and 'Interaction' since the scope of this research is modeling the user interaction (business processes) and the configuration of an internal system.

Argumentation consists of an 'Argument' which defines if the model allows for providing arguments. The preferred value in our case is 'Argument' set to TRUE since it would allow end users to provide arguments for their decisions.

An example of the Content View table is displayed in Figure 6.1. Based on the Content View we conclude that the Business Process Model (OOWS) and the Activity Diagram (UWE).

Aspect	**Attribute**	**OOHDM**		**WebML**	**UWE**	**OOWS**	***Preferred***
		User Interaction Diagram	**Coceptual Class Schema**	**Business Process Diagram**	**Activity Diagram**	**Business Process Model**	
Abstraction	Abstraction	Unique	Unique	Flexible	Flexible	Flexible	Flexible
Context	Intern	–	TRUE	TRUE	TRUE	TRUE	TRUE
	Interaction	TRUE	–	–	TRUE	TRUE	TRUE
	Contextual	–	–	–	TRUE	–	
Argumentation	Argument	–	TRUE	–	TRUE	TRUE	TRUE

Table 6.1: Content View Table

Similar to the CREWS framework we made a Form view and a Domain view. We concluded based on the Form View that – again – both the Activity Diagram (UWE) and the Business Process Model (OOWS) are interesting because of their functional perspective. In the Domain View, we compared the entities from the earlier defined Domain Model with the concepts of the different approaches. Our modeling language will be a construct of multiple modeling languages. Formally defined:

$$MM \subseteq \left(\bigcup_{i=1}^{n} MM_i \right) \cup MM_{new}$$

where MM_1, MM_2,..., MM_n are the meta models of the analyzed modeling languages and MM is the meta model of our new modeling language. MM_{new}

reflects the elements which are not covered by the selected meta models but which are necessary within our new modeling language. The relationship between the entities of the Domain Model and the meta model is defined by function *map*.

$$map : MM \rightarrow D$$

Each entity within the Domain Model should be defined in the meta model. We therefore define the following:

$$\forall d \in D(\exists m \in MM[map(m) = d])$$

Based on the Domain View, the Conceptual Class Schema (OOHDM) seems useful, although it lacks the diversity of entities which makes it hard for business users to use it. The Activity Diagram (UWE) and the Business Process Model (OOWS) provide good possibilities to define the flow of a form. However, they do not provide a way to define the different elements within a form. The Business Process Diagram (WebML) is somewhere in between: it provides the possibilities to define the different elements within the form, but is not that sophisticated for defining the form flow.

Taken all three views in account, we conclude that the Activity Diagram (UWE) and the Business Process Model (OOWS) are the closest to our target modeling language: they provide a way to define the interaction between user and the system, have multiple abstraction levels, a functional perspective and have a strong focus on defining the flow of the form. However, they lack the possibility to define the different elements within the form. The User Interaction Diagram (OOHDM) is more suitable in this particular area because it represents the form elements in a compact and well-organized way. Also the UML Class Diagram (being used by most of the Web Engineering methods) can define different form elements. The Business Process Diagram from WebML has other possibilities to model these forms, but has a less compact notation to define the user interaction with the system.

We therefore selected the Business Process Model, the User Interaction Model and the UML Class Diagram as the base models for our modeling language.

6.4 Specifying the WebForm Diagram

To allow end users to define models which precisely express their expectations of the business process, and automatically translate models into the configuration of a WCM system as defined by its architecture, we need to formally specify our model. Similar to any other modeling language the WebForm Diagram consists of syntax and semantics. The syntax defines the rules, constructs and its decomposition where the semantics defines the meaning and interpretation of the constructs. The syntax and notation should be separated when developing a graphical language [Group, 2005]. We therefore use a concrete and abstract syntax similar to the author in [Cooper, 2007b]. We elaborate on the concrete and abstract syntax in the following sections.

6.4.1 Concrete syntax

The graphical notation of the concrete syntax is the representation model of the WebForm Diagram. It should therefore match the mental model of the user which defined a web form in concepts of steps, routers, formfields and dependencies. More formalized we define form *f* in the following nonuple:

$$f = \{N, S, FE, V, H, C, P, B, E\}$$

Where form f consists of: *N* is the set of nodes, *S* the set of steps, *FE* the set of form elements, *V* the set of validators, *H* the set of handlers, *C* the set of conditions, *P* the set of web pages, *B* the set of blocks, and *E* the set of edges. We elaborate on one element to illustrate how the elements are defined – it goes beyond the scope of this paper to elaborate on each of these elements.

Form Element: A Form element is similar to a Data Entry (User Interaction Diagram) or an Attribute (UML). It is a superclass of elements which are presented to a visitor in a single step of the business process. The set of form elements *FE* can be defined as a union of different formfields where *INPUT* is the set of input types, *BUTTON* the set of buttons and *INFO* the set of information elements. Each step has zero or more form fields in a specific order. The function *fields* provides a set of form elements for a given step:

$$fields : S \rightarrow \acute{S}(FE)$$

where *Ś(FE)* is the set of all possible tuples of form elements. The function *fields* has the following characteristics: each form element is linked to one single step. This can be formalized in the following axiom:

$$\forall f \in FE\, \forall s_1, s_2 \in S\, \forall i, j \in \mathbb{N}$$
$$[occ(f, fields(s_1), i) \wedge occ(f, fields(s_2), j) \Rightarrow s_1 = s_2]$$

The function *occ()* determines if an element f exists on position i within the given sequence and since form element can only exist *once* within a single step we add the following characteristic:

$$\forall f \in FE\, \forall s \in S \forall i, j \in \mathbb{N}$$
$$[occ(f, fields(s), i) \wedge occ(f, fields(s), j) \Rightarrow i = j]$$

Similar to the definition of the formelement, we defined all the other elements of the Form. The interviewees had consensus on how a step should look like: a step was drawn as a square. The form fields are properties of a step. We represented the form fields similar to UML Class-Attributes. The user should be able to choose whether he would like to view the form fields or not (different abstraction layers).

6.4.2 Abstract Syntax

Where the concrete syntax corresponds to the representation model, the abstract syntax is a representation of the implementation model and is a formalization of the Domain Model. There are similarities between the Abstract and the Concrete Syntax, such as *Steps*, *Validation rules*, *Handlers*, *parameters* and *pages*. However, when validation rules and handlers which are connected to a single step are configured within the WCM, they are executed in a consecutive order. Edges are are therefore transformed into two different concepts: a sequential number or to a *router*. The sequential number defines the order of execution from the formvalidations, handlers and routers. An edge which leads to a different *step* or *page* will be transformed into a *router* since it will route to a new step or a page. The actual configuration of the router depends on the conditions of the object. An examples is that a user will only be routed under certain preconditions. Formalized, the WebForm Diagram *f'* is a septupel:

$$f' = \{N', S', FE', V', H', R', P'\}$$

Where form *f'* consists of: *N'* is the set of nodes, *S'* the set of steps, *FE'* the set of form elements, *V'* the set of validators, *H'* the set of handlers, *R'* the set of router, and *P'* the set of edges. Note that Conditions, Blocks and Edges as defined in the concrete syntax are defined within this definition. An Edge for instance is translated into a Router in the abstract syntax. Also in the Abstract Syntax, there are axioms defining the constraints. Example: each router can only be attached to one single step.

$$\forall r \in R' \forall s_1\, s_2 \in S'$$

$$[r \in routersS'(s_1) \wedge r \in routersS'(s_2) \Longrightarrow s_1 = s_2]$$

6.4.3 WebForm Diagram Modeling Tool

We have defined the concrete and abstract syntax. We then developed a modeling tool to implement the WebForm Diagram as a prototype. To build our prototype of the WebForm Diagram we used MetaEdit+ [Kelly et al., 1996]. This application is both proven in building CASE tool as well as also provides computer aided support for method engineering [Tolvanen and Rossi, 2003].

The integration of MetaEdit+ and the WCMS is facilitated by XML: the WebForm is exported from MetaEdit+ as an XML file. This XML file however is MetaEdit+ specific which we transform into an XML resembling the Concrete Syntax (GXML). We then translate the Concrete Syntax (GXML) into the WCMS compliant XML (GXML') using a conversion tool. Since it is a prototype, we have written the conversion in general-purpose languages Java and XSLT. The GXML' can be imported into the WCM system. The transformation process works both ways: WCM form definitions can be exported to XML and can be transformed and imported into back into MetaEdit+.

6.5 Evaluation

We evaluated the WebForm diagram from a functional assessment to see if business users are able to develop a business process in a modeling tool to automatically configure a WCMS, thereby closing the requirements gap and making the implementation more transparent. The reduction in complexity is determined by interviewing experts on the matter.

6.5.1 Case Validation

To test the application in a real life situation, we took a real case situation and implemented a business process from an existing website using the WebForm Diagram. The complete implementation has been tested both ways: i.c. Designing a WebForm Diagram in MetaEdit+, XML transformation and importing it into the WCMS; and the same process in the opposite order to visualize a defined WebForm in MetaEdit+. The organization of the case is a large Dutch telecommunications provider which provides telephone, internet and television services to individuals and organizations within the Netherlands.

A WebForm Diagram is modeled by a business user, exported into the MetaEdit+ specific XML, transformed the exported XML and then imported into the WCMS. The import went without any problems and the resulting form in the WCMS is displayed in Figure 6.4. The form had the correct configuration of

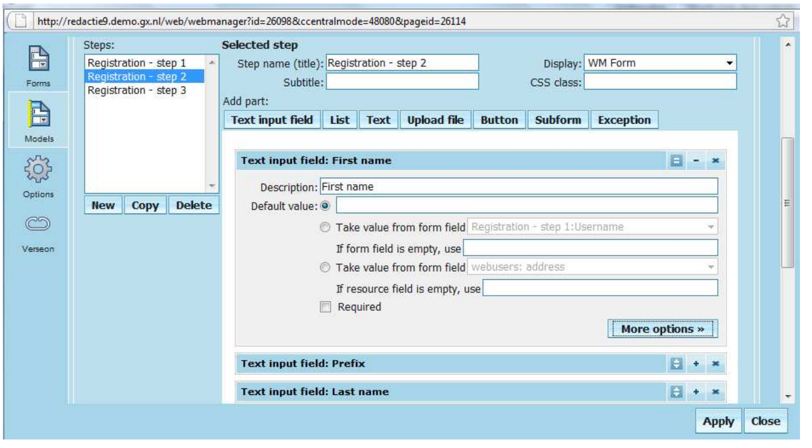

Figure 6.4: WebForm Diagram converted into a WCMS Form Module

handlers and routers which could also be tested by registering a new user using the newly configured form.

To gain transparency of a WCMS implementation, we also validated the process in the opposite direction: creating a WebForm Diagram based on an existing form configuration in the WCMS. This process is slightly more difficult since not all aspects of the existing form can be extracted into an XML. The form we use for this validation is a user sign in form allowing users to access a secure page. This form consists of two steps: Step 1 allows the user to enter his credentials (username/password) and submit them to the web application. Although the form is not that complex, it is an often used business process in online transactions. If the credentials are correct the user will be routed to step 2, the secure page. If the credentials are not correct he will be routed back to the sign in page and will get an error message. We exported this form and transformed it back to the MetaEdit+ specific XML and imported it to present the WebForm Diagram without any problems.

Based on the functional validation, some limitations concerning this implementation where identified preventing it from putting it directly into practice. Within the WCM system, there is an extensive library of existing handlers and routers. Still, in projects new handlers and routers are sometimes developed for project specific purposes (e.g. connect to a specific customer legacy system). These routers and handlers are not available by default in the WebForm Diagram. This resulted in a change in the WebForm Diagram: a new placeholder handler during the modeling phase which needs to be configured or developed within the WCM System afterwards. A second limitation we found when we visualized a web form based on a predefined form definition: some routers were conditional based on volatile information (e.g. user specific session information). This conditional information is not available in the router definition and can it was not taken into consideration when we developed the WebForm Diagram.

6.5.2 Expert Validation

We validate the complexity of the WebForm Diagram by interviewing eight users: four technical architects and four functional consultants. The aim of this validation is to find out if the users find the WebForm Diagram useful and applicable in implementation projects. The users were shown the WebForm Diagram as illustrated in Figure 6.1 with additional abstraction layers to detail the form steps. They were provided with a list of statements addressing usability, the suitability, transparency (for example to use with internal and external communication) and if they would use it in actual projects. The list with statements had a 4-points scale resulting in respectively the following predicates: Strongly disagree, Disagree, Agree, and Strongly Agree. The results were then gathered and summarized.

The users find the Diagram clear, easy to read, and easy to use. They also think that it would improve the transparency within projects. The architect and the consultants are convinced that it would improve the requirements analysis and design with the end customer. Using this model will improve the validity of the requirements since the customer will probably understand such a diagram easier then the written descriptions. The architect can also use the model to check whether all relevant information is available for the implementation of a business process. However, they are not yet convinced that the handlers and validators will help the design phase much. The general opinion is that it depends much on how it is visualized in the modeling tool. They agree however that this visualization should keep a balance of usable icons but not an overload of details. Other icons in the diagram are quite abstract yet useful. They all agree that a WebForm Diagram will improve the visibility of a form and that end users will gain insight. They also state that they want to use the modeling tool if they have the proper tools supporting it. However, a remark is made concerning the fact that the current application as developed in MetaEdit+ is not web based in contrast with the WCM system. It would therefore require a rebuild in a web based environment to get it into production.

6.6 Related Work

In the field of Web Engineering, there are several research groups working on the development of MDE products. We briefly elaborate on four relevant research groups.

One research which started early with MDE is the Object-Oriented Hypermedia Design Method (OOHDM) as proposed by Schwabe and Rossi [Schwabe and Rossi, 1995], [Schwabe et al., 1996]. OOHDM comprises of four steps: conceptual modeling, navigational design, abstract interface design and implementation. These activities are performed in a mix of incremental, iterative and prototype-like way. In OOHDM a clear separation is made between on the one hand navigation and the other hand presentation. OOHDM has introduced powerful navigational concepts for contexts, personalization and user interaction.

Koch et al. describe the UML-based Web Engineering (UWE) approach in [Koch and Kraus, 2002]. UWE is an object-oriented, iterative and incremental approach for the development of web applications. The development process of UWE consists of five phases: inception, elaboration, construction, transition and maintenance. The approach focuses mainly on customized and adaptive systems and also do not take content management and integration of external sources into account.

Pastor et al. describe different methods with the Object-Oriented Web-Solutions Modeling approach. OOWS provides mechanisms to deal with the development of hypermedia information systems and e-commerce applications in web environments [Pastor et al., 2001]. OOWS strongly focuses on the generation of the required web application and less about managing the content and the application afterwards. OOWS extends OO-method by means of a navigation model, a presentation model and a user categorization. OOWS comprises of two main activities: system specification and solution development. Similar to our framework: the OOWS approach is supported by a commercial tool called OlivaNova.

Ceri et al. describe in [Ceri et al., 2003] their Web Modeling Language (WebML), a notation for specifying complex websites at a conceptual level. In line with the definition of Web Engineering, the WebML approach consists of seven phases: requirements specification, data design, hypertext design, architecture design, implementation, testing and evaluation and maintenance and evolution. The WebML method is supported by a commercial Model Driven development environment called WebRatio that allows modeling and automatic generation of web interface applications.

These models are used within our research. We compared these models based on the Cooperative Requirements Engineering With Scenarios Framework

(CREWS) [Rolland et al., 1998].

6.7 Conclusion

The purpose of this research is to reduce complexity and increase the transparency of the development of online business processes supported by Web Content Management Systems. To achieve these goals, we defined modeling tool to automate the configuration of Web Content Management Systems with a strong focus on defining the business processes. Based on user analysis and domain modeling, we presented the WebForm Diagram which utilizes fragments of established Web Engineering methods. The WebForm Diagram consists of an abstract and concrete syntax resembling the implementation model and the mental model respectively. We developed a prototype of the model in MetaEdit+ and were able to automatically configure the WCMs. Several abstraction layers in the WebForm Diagram supports the different process steps in the implementation project. We validated the WebForm Diagram by means of a prototype validation and an expert evaluation. The results from both the prototype and user evaluation were positive and promising. However, the validation presented some technical limitations which need to be addressed with the modeling tool being web enabled is most important. Another aspect to consider is the fact that we use general-purpose languages such as Java and XLST in the prototype to transform the different models while upcoming model-to-model transformations seem promising [Koch et al., 2008]. We believe that we have made an improvement in an approach for WCMS-based Web Engineering and that concepts of this research can be applied beyond the scope of WCMS. Future research includes further development of the WEM Framework and refinement of the modeling tool to support the automated configuration of WCMS.

Chapter 7

Verticalization of CMS-based Web Application: An Approach to Building Industry Templates

The main purpose of this paper is to improve a web content management system (WCMS) product line for future implementations by identifying software commonalities in WCMS-based web applications. WCMS play a central role in modern web application development: most large public and internal websites are based on a WCMS foundation. If we can improve the implementation process the effectiveness and efficiency of web application development will increase significantly. This research identifies reusable solutions from existing WCMS implementations using Problem Diagrams and Structured Goal Modeling. From configurations were matched with atomic e-Business models by linking them to the strategic competencies through bottom-up goal modeling. A designed method was constructed on how requirements can be elicited in WCMS implementations using goal-modeling and problem frames. The resulting method provides insight in relevant e-Business models and their relation to software product lines. Moreover, the approach is applied in a WCMS study which demonstrates its applicability.[1]

[1]This work is accepted for publication in the International Journal of Web Information Systems, entitled 'Identifying Commonalities in Web Content Management System Engineering' in 2011

7.1 Introduction

Since the last decade when most websites were developed to publish static information, they have evolved steadily and ubiquitously to serve dynamic and complex web content and business functions. With the growing popularity of websites and web applications, the web aligned with e-Business models started to emerge, which relies on server side programs executing business logics hosted on application servers to generate the dynamic HTML as per each user request [Ravi et al., 2009]. Moreover, in response to the increasing amount of content to be managed and its scatteredness throughout organizations have resulted in the growing popularity of Content Management products [Grahlmann et al., 2010]. To provide organizations with and their marketing communications department in particular the flexibility to publish dynamic and personalized content on the web, a specific type of Content Management product software evolved called Web Content Management System. A Web Content Management System (WCMS) is product software which can be tailored and customized by means of configuration and will lead to a CMS-based web application [Souer et al., 2007b]. Most dynamic websites utilize some sort of WCMS to support the organization with their online business because it allows them to create web initiatives in a time and resource efficient way based on standardized components.

A WCMS is configured and customized during each implementation project to meet an organizations requirements. This process can be time consuming and if several organizations have similar requirements, the same customizations are repeated. The goal of this research is to optimize the implementations of WCMS-based web applications. We focus on the identification of software components that can be reused in similar situations. This leads to more efficient and effective implementations. The idea is that if organizations within a certain domain have similar requirements the WCMS implementation could be optimized by means of pre-configuration of the software product, or so called ‘Industry Templates’. Our main research question is therefore: ‘How can we identify software commonalities in CMS-based web applications and incorporate that in the WCMS product line to improve future implementations’. We answer this question by comparing implementations of companies in the publishing sector using a reference model for the WCMS implementations and identify commonalities that provide input for the product software (i.e. the WCMS itself). This paper presents a new approach to identify software commonalities in WCMS Product Family Engineering using atomic e-Business models of Weill and Vitale [Weill and Vitale, 2001]. Moreover,

[Souer et al., 2011a]. A shorter version of this paper was published in the Proceedings of the 12th International Conference on Information Integration and Web-based Application & Services (iiWAS), in 2010. The work is co-authored with Dirk-Jan Joor, Remko Helms, and Sjaak Brinkkemper.

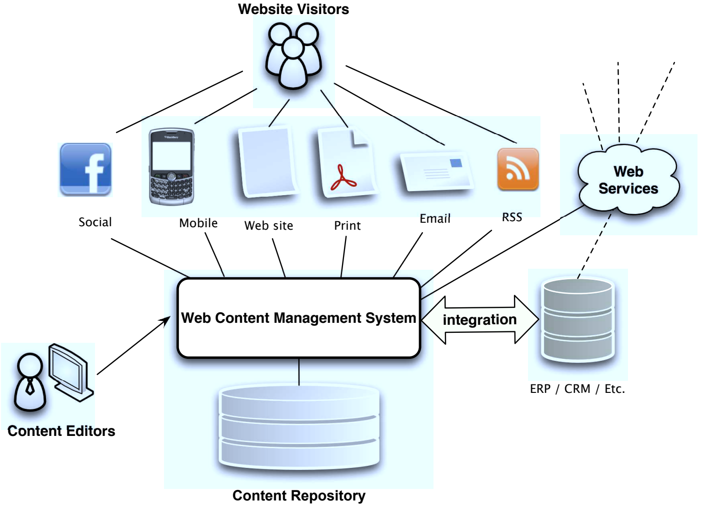

Figure 7.1: Overview of a Web Content Management System

we propose an adjusted approach for the derivation of generic requirements from atomic e-Business models to requirements where we combine goal modeling [Bleistein et al., 2004b] and problem frames [Jackson, 2001].

The research area we are addressing in this paper is web engineering, which is defined as 'the application of systematic and quantifiable approaches (concepts, methods, techniques, tools) to cost-effective requirements analyses design implementation, testing, operation, and maintenance of high quality web applications' [Kappel et al., 2006]. We defined the process of web application engineering using a WCMS as foundation in our Web Engineering Method [Souer et al., 2007b], [Souer et al., 2008b] and [Souer et al., 2008a]. This process is based on the idea that all relevant components are selected and congured within the WCMS. Only if the customer requires a specific functionality that is not available within the WCMS or as an off-the-shelf component, the WCMS must be customized. However, most common requirements for a website are realized using a configuration of available components. That is the main reason why using a WCMS is interesting from a resource and investment point of view: since there are configurable standard components available, there is less development needed while maintaining a high quality. Although we believe that using a WCMS for web applications engineering leads to higher quality web applications in a cost-effective way, we keep searching for improvement of the development process of

CMS-based web applications. In [Souer et al., 2009] for instance, we investigated a model driven approach to automate the gap between requirements analysis and software product configuration.

This research consists of two main parts: the Part I is focused on the identification of the reusable components and matching those components with e-Business models which is described in [Souer and Joor, 2010]. This paper focuses on Part II where we consolidate the identified e-Business Models and show how they can be integrated into a software product line. The remainder of this paper is organized as follows: Section 2 provides our theoretical background on software product lines as a theory for reusing software and e-Business models as a new way of grouping functionalities. Section 3 provides the main artifact of our research, the goal modeling approach with problem frames that provides the basis for standardized product configuration or new business requirements for the next iteration of the product release. In Section 4 we validate the model at a case study company to test its application, which is evaluated in Section 5. Finally conclusion and future work are discussed in Section 6.

7.2 Background

7.2.1 Software Product Lines

To identify commonalities we look at the theory of software product lines since a software product line offers an efficient way for software reuse [Clements and Northrop, 2001] and maximizes the use of common assets by construction of a reusable infrastructure from which family members can be constructed [DeBaud and Schmid, 1999]. Identifying the commonality is also one aim of the product line platform. A *commonality* is a software solution or characteristic, such as expressed in a functional specification, that applies for the family of products within a solution domain [Muthig and Atkinson, 2002]. Relevant product line assets are *business requirements*, *architecture*, *design*, and *implementation*. Muthig defined two dimensions to categorize the reuse strategy of software products. The first dimension is the extend in which a supplier follows a customer orientated approach; higher customer orientation increases the variety of software products from a family. The second dimension is the extent to which the supplier follows a market oriented approach; higher market orientation increase the change rate of the product platform developed to keep up with the changing market demands. These two dimensions are illustrated in Figure 7.2

- *Individual products*: The extreme towards the customer on the spectrum. An organization develops tailor made applications for individual customers.

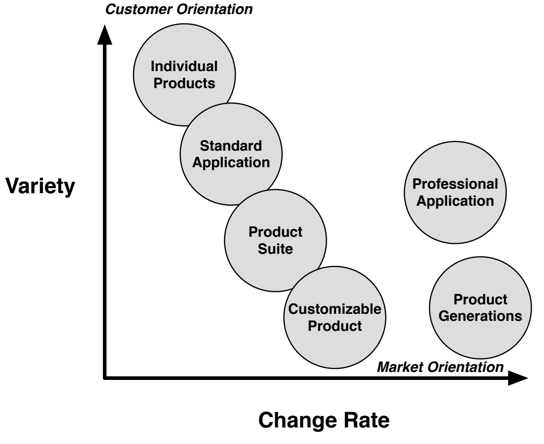

Figure 7.2: Characterization of software product lines

The variety of the product family is the number of delivered systems. These systems are relatively stable so change rate is low.

- *Standard application*: This product platform is sold and initially installed at customers sites, but extended to fully satisfy the customer needs. The latter is also part of the selection process and extensions and adaptations are a large part of the overall development effort driven by individual customers. The variety is lower because of its relative stable application core, but the change rate is increased for the many requirements that cannot be fulfilled by the core alone.

- *Professional application*: This provides all features to satisfy experts and professional users in an application domain. All sorts of applications are derived from this super application on which successors are also created. The variety of the super application is lower because of the many features available, but its change rate for keeping up with the features is high.

- *Product suite*: A set of products that provide complementary functionalities that can be used separately or in conjunction with products from the suite. The variety corresponds to the number of products in the suit. The change rate corresponds to the frequency in which new versions from single members of the whole suite are released.

- *Product generations*: This is related to individual products, but developed for a mass market instead of individual customers. Each generation derived from this original product is replaced by it successor product which adds functionality problem fixes or technology upgrades. A large number of older and newer generations are maintained as individual software products. The variety in individual software products is therefore low. The change corresponds to the frequency in which new generations are released.

- *Customizable Product*: This is a single product that covers a spectrum of applications in such a way that an implementation variant is not explicitly handled as an application on its own. The customization is handled via user interface controls, but may be done earlier during implementation when configuring variability becomes more complicated. The variety is the number of maintained product version; the change rate is lower than product generations because new requirements are being integrated as variability within the existing product

7.2.2 Commonalities in Software Product Lines

To reach economical benefits through software reuse, the level of variety of a software product should be low so most software can be developed prior to implementation to be quickly implemented as a solution for anticipated business requirements. Our research is focused on identification of commonalities in specific product software: a Web Content Management System. To find commonalities in WCMS implementations, we first identified a classification in [Souer et al., 2008b] that allows us to compare the different implementations. Based on that classification, we used *feature modeling* as a form of variability modeling [Bosch et al., 2002] to identify commonalities in the dual lifecycle if product software [Weiss and Lai, 1999]: the distinction between Web Product Family Engineering (or the product development process to create the WCMS) and web application engineering (the implementation of the WCMS which results in a CMS-based web application). Feature modeling is a notation technique for variability modeling which represents common and variable features in a product line [Czarnecki et al., 2005]. A feature links the design requirement and business requirements and is defined as a ‘coherent and identifiable bundle of system functionality that helps characterize the system from the user perspective’. In a WCMS, for example, a ‘Form module’ allows an organization to configure a form on their website. If we could identify characteristics over multiple organizations, we might be able to improve the WCMS with pre-configured Forms for a specific domain. The dual lifecycle of the software product lines is illustrated in Figure 7.3.

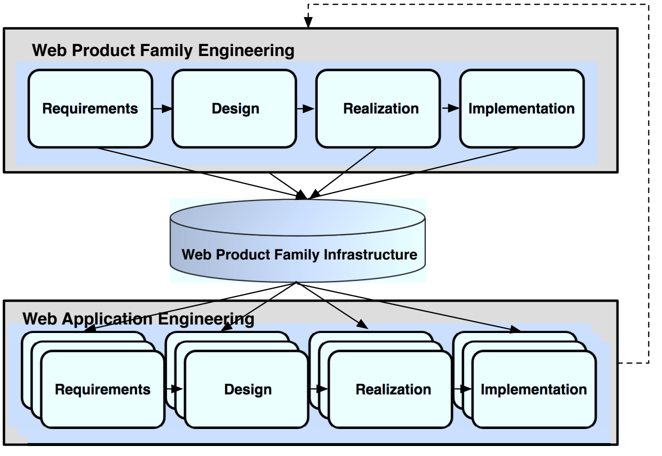

Figure 7.3: Web Product Family Engineering and Web Application Engineering

7.2.3 e-Business Models

In traditional product software it is common practice to identify customer domains based on market sector (or 'verticals') [Xu and Brinkkemper, 2007]. The idea is that within a domain, it is expected that the customers have similar business requirements – and therefore similar implementations. Typical examples are industries like Financial and Insurance, Healthcare, Pharma, Manufacturing, and Public Sector. However identifying new ways of doing e-Business may require a different perspective than this traditional approach [Jeary and Phalp, 2004]. The atomic e-Business model to identify patterns in organizations is one way to do this. The competencies and critical success factors from the atomic models by Weill and Vitale [Weill and Vitale, 2001] are regarded as a best practice strategy for e-Business [Bleistein et al., 2004a]. To illustrate the e-Business models, we detail the 'Intermediary' in Figure 7.4 as it is the most common model among publishers. An intermediary brings together buyers and sellers by concentrating information and does this in a variety of forms. The Intermediary earns money by mediating between suppliers and customers (or sellers and buyers). Although information and money flows through the intermediary, the product itself is delivered from a supplier to a customer without intervention of the intermediary. The most common form was a portal for a targeted audience by narrowing its focus on a specialized topic [Weill and Vitale, 2001]. The website acts as a gathering place for industries or topics of interest and provides the latest research, calendars and discussion groups. Predictions are that there will be as many portals as there are specialized topics, although there is unlikely to be room for more than one portal for each topic given. The link between publishing companies and the portal is explained by the intent of leveraging its core business; the content available from their already specialized magazine. In [Souer and Joor, 2010] we elaborated on the e-Business models and how we used feature modeling as defined in [Kang et al., 1990] to identify common features in our web application engineering process [Schmid et al., 2006] and to identify the underlying e-Business models. This was based on the assumption that each of the eight atomic e-Business models has core competences that are needed to sustain the model [Prahalad and Hamel, 1990].

7.3 Identifying Commonalities: A Goal Modeling Approach

Following the approach by the authors in [Bleistein et al., 2004a], a goal modeling technique is applied to translate the best-practice strategies from the atomic e-Business models to generic requirements. Although goal modeling has the traceability needed for aligning high level strategy with low level business requirements, goal modeling alone provides little guidance for reaching consistent modeling: goal modeling leads to unsatisfactory results as the models differ from one requirements engineer to another [Bleistein et al., 2004b]. To bring structure in the approach, goal modeling is combined with *problem frames*. We detail the problem frames below and how they combine with goal modeling.

7.3.1 Problem Frames

Software problems originate from real world problems, and a software solution must address its real world problem in a satisfactory way [Bleistein et al., 2004b]. A software engineer must therefore understand the real world problem he intends his software to address. To this end, the software engineer must understand the problem context and how it is to be affected by the proposed software, expressed as the requirements.

Jackson proposed the Problem Frames approach to cope with this concern [Jackson, 2001]. Problem frames offers a set of reoccurring problems that tend to occur in the diversity of software development. Problem Frames are a means of u describing the problem context where software provides a solution. These allow structuring of the development process even when the problem area is relatively

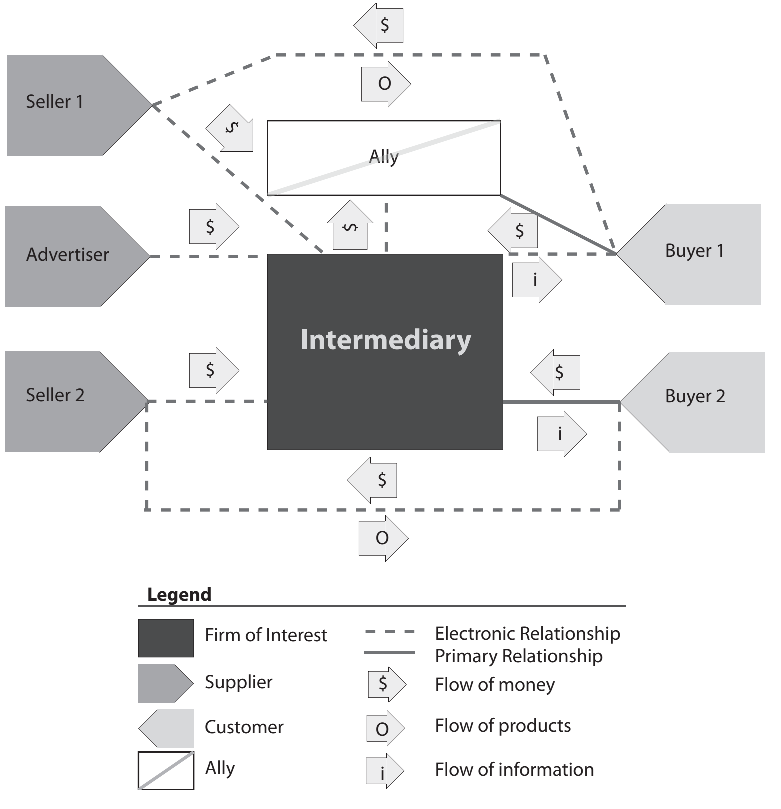

Figure 7.4: E-Business model schematic of Intermediary – by Weill & Vitale

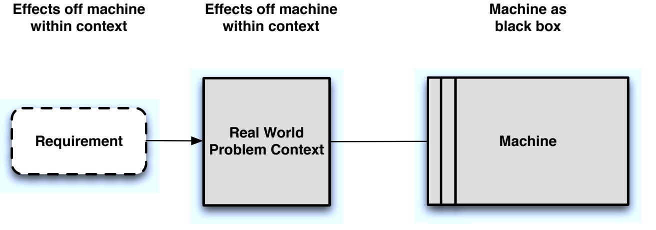

Figure 7.5: Basic structure of a Problem Diagram

unknown. Problem frames also offer a solution to 'shallowness', which can occur when the developer does not fully understand the problem context. The notation used within problem frames is illustrated in Figure 7.5. From right to left, the first part on is the software to be developed, a *machine*. A *context diagram* consists of the identified aspects of the real world that are relevant and which set the problem context. Within this context the Machine should produce effects. These effects are described in terms of the third part of a problem diagram; its requirements.

Since realistic problems are too complex to be captured in a single diagram Jackson introduced the concept of a *progression of problems* [Jackson, 2001] as illustrated in Figure 7.6. In the progression of problems the problem could start in a context as close as an actual problem in 'the real world' and progress with a sub problem 'closer to the machine' until the sub problem reaches the level of the machine, which is the actual configuration of the software. The problems at a level closer to the machine have a different set of concerns to deal with. Whereas a problem in the 'real world' deals with people and unpredictability, a sub problem close to the machine has far more formal problems and deals with transactions and symbolic representations of information.

The concept of progression of problems is effective in its simplicity to drill down problems from e-Business level to operational or machine level. However, the Problem Frames approach provides little to no link between requirements at the different levels. This is important in our research since we use a requirements decomposition to analyze e-Business models based on requirements.

7.3.2 Structured Goal Modeling

Goals are optative statements that express a wish and can be used both to express requirements at a high strategic level or a low technical level [van Lamsweerde, 2001]. When achieved, the lower level goal should satisfy the higher level goal to which it contributes. Next to eliciting requirements, goals

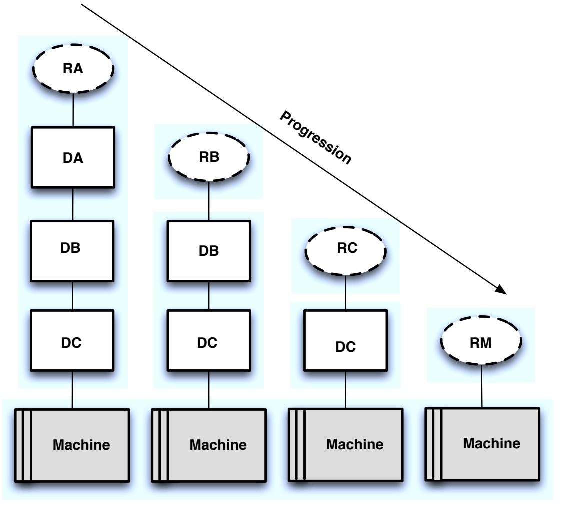

Figure 7.6: Progression of Problems

offer traceability, achieve requirement completeness and prioritization, explain requirements to stakeholder and provide the right level of abstraction to consider alternative solutions. An Example of a high level goals is ‘Optimal Customer Service on the website’ which could have a lower level contributing goals ‘On every webpage should contain a help and a contact button’.

Although goal modeling has the traceability needed for aligning high level strategy with low level business requirements, using goal modeling alone there was little guidance for reaching consistent modeling. It ‘resulted in feeling somewhat ad hoc and difficult to comprehend to what extend the models would have differed between different requirements engineers using the goal modeling approach’ [Bleistein et al., 2004b]. It also left out the requirement scope of where to begin modeling. Bleistein et al. proposes a hybrid approach, combining goal modeling and the problem diagrams with the concept of progression of problems to translate a high level e-Business strategy ‘deep in the real world’ into business requirements ‘close to the machine’. The context diagrams help ensure that the requirements are consistent with business and system context, while the goal model provides a means to link the business requirements with strategic objectives. The starting point of the progression of problems is the e-Business model. When comparing the notation technique from the atomic e-Business models as illustrated in Figure 7.4 with that of a context diagram for problem frames a fit can be

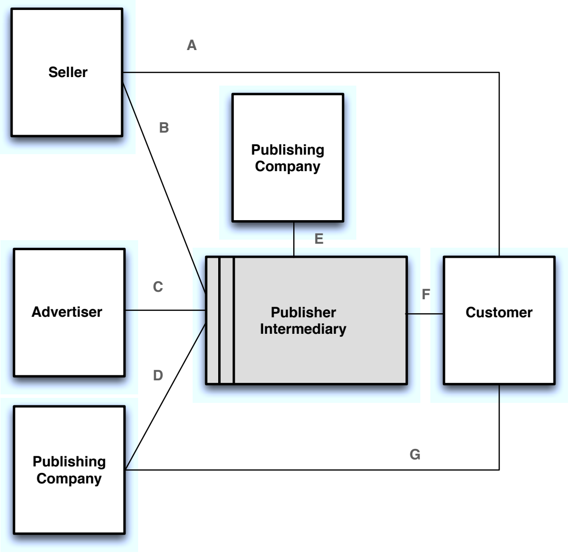

Figure 7.7: Atomic e-Business model for a publishing company in a context diagram

made. Business model participants can be represented by given domains, with their electronic relationships indicated as interfaces on which shared phenomena such as money, products and information flow.

The starting point of the progression of problems is the e-Business model. When comparing the notation technique from Weill and Vitale with that of a context diagram for problem frames a fit can be made. Bleistein et al.propose that business model participants can be represented by given domains, with their electronic relationships indicated as interfaces on which shared phenomena such as money, products and information flow. An example of an atomic e-Business model fitted in the problem frame context diagram is given in Figure 7.7.

The principle of goal modeling, which states that a lower level goal should satisfy a higher-level goal, still applies here and like the progression of problem, a lower sub problem should be derived from a higher problem deeper into the world. When combining these principles, the integrated approach is constructed as illustrated in Figure 7.8. The e-Business model context of Domain A (*DA*) has its best practice strategy that forms the requirements patterns of domain A (*RA*). Domain B (*DB*) is deferred from the e-business model context of *DA* and its requirements should satisfy the strategy of *RA*. Domain C (*DC*) is again deferred

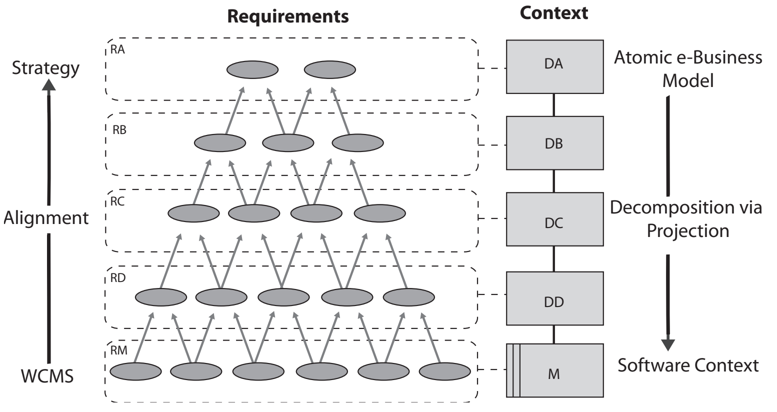

Figure 7.8: Goal model integrated with progression of problems

from its deeper in the world context; in this case *DB*. *DC* requirements (*RC*) should satisfy the requirements in *RB*, until the machine context is reached where the requirements are expressed in software configuration.

Using the goal modeling approach combined with problem frames, we are able to identify general requirements for the atomic e-Business models. That provides the basis for standardized product configuration or new business requirements for the next product release.

7.4 Validation

We applied the structured goal modeling approach and the feature models to identify e-Business domains at a case study company. In the next sections we will introduce the case study company and the results of the analysis.

7.4.1 About the Case Study Company

This research was conducted at GX Software. The software products that GX Software develops are a WCMS called GX WebManager and an Online Customer

Product Line Type	Potential for finding generic requirements from the problem domain
Product Platform	No: The WCM authoring, which is implemented as commonality, is expected to be an often occurring, maybe even implicit requirement.It is difficult to relate this to a strategic objective from a particular atomic e-business model. Yes: variability configurations could be determines by customer requirements patterns from the e-business model.
Product Suite	Yes: requirement patterns from the problem area could determine which components are installed and configured.
Standard Application	Yes: patterns of requirements could determine which solutions are tailor made and which are future candidates for variability in either the product suite or product platform. The tailor made solutions may be treated as features that are not yet available in the product suite or product platform for future product line scoping.
Individual Application	No: this software is uniquely developed for a single customer and generic requirements are not expected. No: the website design is made by the editor in the customer organization. Although some design requirements may be too complex for editors unfamiliar with the particular WCMS and be initially developed by its supplier.

Figure 7.9: Overview of requirement reuse potential in WCMS development

Engagement product called BlueConic. GX WebManager is delivered as a standard platform that is configured during the implementation for each customer. GX Software has three locations: Nijmegen, Stamford and Redwood City. Its product GX WebManager is developed on a component based architecture that allows organizations to build, share and reuse functional components.

Using the product line characterization dimensions as illustrated in Figure 7.2, the product line from GX Software is described to investigate which parts of the software solution could be determined from generic requirements in the problem area. The product line setting in which the WCMS is developed forms a mix between the product line types on the dimensions and can be described as a combination of a customizable product, product suite, and standard application. The product line from the case study company is described to investigate which parts of the software solution could be determined from generic requirements in the problem area.

7.4.2 Document Analysis

To identify the e-Business models we analyzed WCMS implementations which is known as *archival analysis* [Yin, 2002]. We use feature modeling as defined in [Kang et al., 1990] to identify common features in our Web Application

Engineering [Schmid et al., 2006] and to identify the underlying e-Business models.

The translation is specifically done for publishing companies and their identified atomic e-Business models of Intermediary, Full-service Provider and Virtual Community. The most encountered atomic e-Business model at the initial investigation of publishing companies is the intermediary. Similar to what Weill and Vitale claim, most organizations have more than one single e-Business model. The atomic e-Business models at implementations were identified according to their core functionality and that the commonalities among functional concepts simply represent the functionality that identified the atomic model in the first place. This applies to some extend for the full-service provider as these were identified by their high ownership of customer data. It should also be noted however that two companies also had the full-service provider objective of obtaining detailed customer profiles stated in their requirement documents and may be identified on evidence other than their functionality. The intermediary was identified by their use of bannering to earn revenue through advertisement.

7.4.3 Reusable Components in Web Content Management Systems

Based on the document analysis and the bottom-up goal modeling, we placed all requirements in a large table categorized by e-Business model. For each requirement, the *relevancy* and the *fulfillment* of that by the WCMS was determined. The relevancy is defined by the percentage of publishing companies that requested that particular requirement. And the fulfillment is defined as a number on the scale of 1 to 10 where 1 is not available in the WCMS in any way and requires a custom component, and 10 is a full match which means that the requirements is met by the WCMS and doesn't require any customization at all. A part of the table is presented in Table X . Note that the requirements with a high relevancy and low fulfillment are those requirements that are the most interesting requirements since they are requested by many organizations but not part of the standard offering in the WCMS.

Based on the table, a few conclusions can be drawn: a WCMS fulfills most requirements that are in line with our statement that a WCMS provides a foundation for web application development. However there are a few requirements that are not (or at least not good enough) covered by the WCMS. Some of them are 'Calculation tools' which are small modules allowing a website visitor to calculate a specific number such as a subscription fee. Also the 'Customers can access needed resources from the database' is a requirement that is not fully covered.

	Relevance (%)	Fulfillment (1 - 10)	Session Logging	User Registration	User Content Access	Directory Access	E-Forms	Advanced Forms	E-Commerce	Calculation Tool	Captcha	E-Form Handler	Quiz Module	Social Bookmarks	Weblog	Discussion	Content Rating	Poll
E-Forms																		
Intermediary																		
Contact form	73%	10					x											
Apply for subscription	67%	7					x											
Change subscription	13%	5						x										
Buy products	13%	8						x	x									
Virtual Community																		
Contact form	73%	10					x											
Calculation Tools	13%	5						x		x								
Buy products	13%	8						x	x									
Full-Service Provided																		
Contact form	73%	10					x											
Apply for subscription	67%	7					x											
Change subscription	13%	5						x										
Buy products	13%	8							x									
Access external resources	13%	4						x										
Calculation tools	13%	4						x										

Figure 7.10: Analyzing goals with available product platform solutions

The WCMS has a generic database resource connector, however it requires an implementation specific configuration to show the right content. This way, we created a list of potential reusable components and evaluated them with the product management team in the case study company.

7.5 Evaluation

We evaluated both the process of bottom-up goal modeling and the outcome of the product requirements in two workshops, one held for application engineering and the other for product family engineering. The sources used to collect data are interviews, observation and participation. Interviews are done both structured using a questionnaire and using open-ended questions during workshops. Documents as data source indicates that the information has been retrieved prior to conducting the case study. Participation is done by constructing a proposed deliverable for analysis for the case study participants to evaluate. This evaluation is focused on the product family engineering aspect.

7.5.1 Product Family Engineering Workshop

The evaluation was conducted in the form of a workshop in which the product management team participated consisting of one senior product managers, one junior product manager and one senior product consultant. The workshop itself gave the participant the opportunity to place business requirements from a large publisher company in the existing goal model by bottom up modeling. This publisher company is one of the largest publishers in the Netherlands and a real life customer of the case study company. The requirements of the implementation were collected in advance as a sample of convenience. The participants were asked to increase the available product assets by adding the requirements to an existing goal model from the requirements document of 15 publishing companies.

The goal model from the 15 publishing companies was prepared prior to the case study and ordered in sections for an intermediary, virtual community and full-service provider domain for e-Business. The diagrams used were the same from bottom-up goal modeling for identification of atomic e-Business models as illustrated in Figure 7.11. In total the product line asset consisted of 12 diagrams per e-Business model that were ordered according to their lexical context of symbolic information, e.g. ‘digital assets’, ‘user profile’ or ‘user content’. Each diagram had an empty template to add new goals into the model. The participants were to find out in which e-Business model the new requirements were most appropriate and at which computer-based information they fitted (lexical context)

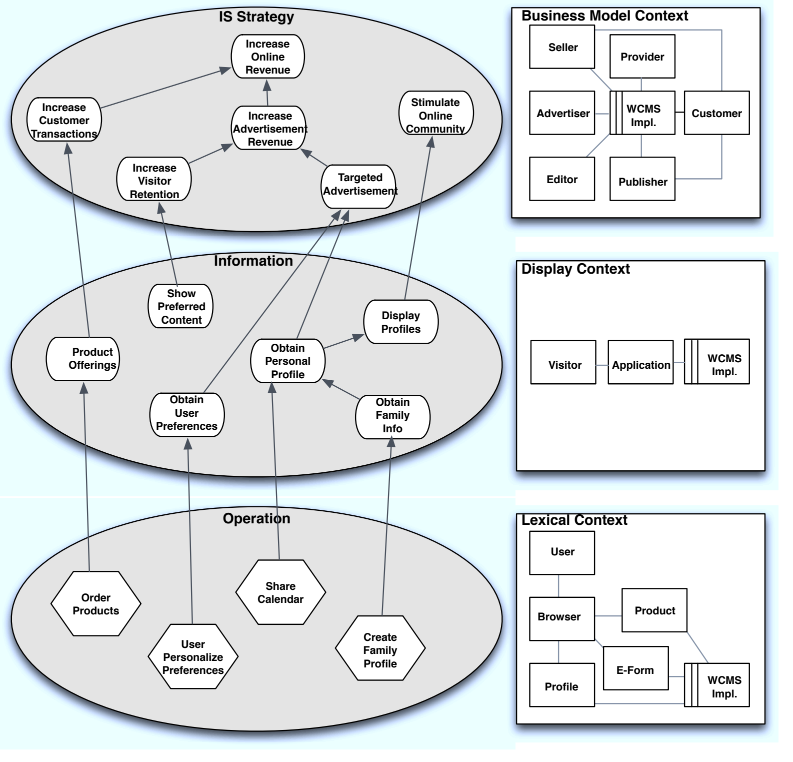

Figure 7.11: Example model of validation

for ordering requirements. The new requirement increases the completeness of the model which then could be used as a product line asset on which new product solutions decisions are made.

To establish software platform completeness and find new opportunities for reuse, the resulting business requirements from the goal models were analyzed in a table that compared business requirements from the bottom-up goal model of the 15 publishing companies with existing WCMS solutions. In this analysis with existing components of the WCMS product were used to validate if the original component descriptions from the case study company without feature model could be used. Although their application depended on customer inputs for need and solution fulfillment, while this application used correlations from the authors judgment. Finally a questionnaire was handed out for asking the direct questions about the proposed method and techniques.

7.5.2 Expert Evaluation

The first topic case study question specifically related to product family engineering is ‘are business requirements from the structured goal model useful for finding new product development opportunities?’ The benefits for finding new development opportunities for the development process were appreciated as mildly positive . A discussion during the workshop revealed that efficient reuse by finding new requirements patterns is interesting but the main product development focus is on finding unique strategy product functions rather than reusing best practice strategy. One participant remarked that the commercial benefits of the technique did became clear when the business requirement of the large publication company was indeed done at prior publishing companies and a basis for reuse was found.

Whether development opportunities can be identified by the product managers through bottom up goal modeling is address by the case study question ‘is bottom up goal modeling useful to apply in the product family engineering process?’ It was also remarked that the development of the model should be part of the application engineering process rather than the product family engineering process. The product managers considered structured goal modeling less useful for product family engineering. A perceived benefit for application engineering was the enablement of knowledge sharing between consultants for feature identification. A remark was also made about the difficulty in identifying the goals for the requirement such as ‘integration with subscription administration’ and that this could have been asked the customer directly during application engineering.

Another question reported concerns the product line identification with the question ‘Are the atomic e- business models appropriate for identifying product lines?’ The atomic e-Business models were considered as useful for recognizing domains on the web. The application of mapping business requirements with the atomic e-Business models was considered mildly positive and the model from the 15 publications companies was more or less understandable. This might have been caused by the difficulty in understanding the match between a particular e-Business model and business requirements.

During observation, the identification of which diagram a business requirement should belong according to its lexical context was performed with consistent results. The business requirement concerning ‘personalized tools’ was associated by all participants with the lexical context for ‘personal profile’. With regard to the atomic e-Business models there was a larger discrepancy. The family profile page was placed under the virtual community model by one participant, while another placed it under intermediary. After the application the participants discussed the technique and their findings in an open ended interview. It was remarked that although the related context was clear, there was ambiguity under which atomic e-Business model the requirements should have been put.

Two participants were observed to start their requirement modeling by dividing the business requirement in smaller operations first and then putting these in the model. This indicates that for discrimination between businesses requirements from the atomic e-Business models would be further enhanced by more detailed specifications. As a consequence, using the structured goal model for its full product line identification potential would mean having fully specified requirements. To continue specifying more detailed requirements, a different technique such as use cases or textual description would have been more appropriate as the author noticed that goal model diagram space is limited for specifying these amounts of requirements.

7.6 Conclusion

In this paper we elaborated in Web Content Management System based Web Engineering with a strong focus on the reusability of components in software product lines. From structuring goal elicitation with the problem frames approach it can be concluded that eliciting WCMS requirements in e-Business starts from identifying the e-Business model of the web initiative and goals are set for the required user operations in the WCMS. Having the knowledge of both the available software solutions in the product platform of the WCMS and the objectives from a e-Business strategy, the requirement engineer could use this as the starting point to prioritize requirements by applying reusable software for best-practice strategy. A remark should be made that many organizations do not explicitly apply their e-Business model in requirements engineering. We do believe that we have found a useful solution to improve WCMS-based Web Engineering, but further validation and real life cases should provide more data. Further research includes applying these models in other vertical markets to identify software commonalities. Also it might be interesting to investigate how the atomic e-Business models can help understand the reuse of product software components in other software domains.

Part IV

Conclusion and Discussion

CHAPTER 8

Conclusion

This research has led to two main conclusions. The first conclusion is that there is now a Web Engineering Method for developing Content Management System-based web applications where one was lacking. The second conclusion is that innovations can be applied to the CMS-based Web Engineering approach using a model-driven paradigm and product software family approach. The main contributions of this work are the Web Engineering Method (WEM) and a prototype to support the model-driven approach. The Web Engineering Method enables organizations and product software vendors to reduce the time it takes to implement CMS-based web applications.

8.1 Research Questions

We introduced in chapter 1 the main research question of this research concerning the web engineering processes based on Web Content Management Systems. Our aim was to construct CMS-based web engineering processes on the one hand and to innovate the process as well. We elaborated on those two main topics in part 2 (Developing Content Management System based Web Applications) and 3 (Process Innovation) of this dissertation. We defined the CMS-based web engineering processes using design science research and method engineering. The process innovation part extended the constructed Web Engineering Method by enriching the concepts and deliverables with a model-driven web engineering approach and prototype. Furthermore, we developed an approach to improve the Web Content Management System product line for future implementations by identifying software commonalities in CMS-based web applications. In the following sections the findings of the research presented in this dissertation are related to the research questions.

8.1.1 Developing Content Management System-based Web Applications

The main research question concerning the development of CMS-based web engineering:

RQ1: What are the concepts and processes of web engineering using a web content management system as a foundation?

The answer to this question consists of the Web Engineering Method (WEM) that describes all the five phases in detail: requirements, design, realization, implementation, operations and maintenance using Process-Data Diagram (PDD) descriptions. These provide insights into both the process steps and deliverables attached to each process. WEM is assembled using fragments of web engineering methods from literature and best practices found in a vendor of a Content Management System using a situational method engineering approach. The method has been validated in several cases and by expert evaluations. The method can be used by practitioners in organizations that are involved in the development and implementation process of Content Management System-based web applications. *RQ1* was divided into four related sub-questions on which we will elaborate in the next sections.

SQ1.1: What is a web content management system? This dissertation defines a (Web) Content Management System as a specialization of Enterprise Content Management System and provides an explanation for why modern web applications are built based on content management principles. We defined CMS-based web applications in chapter 2 as a web application for the management and control of content. Furthermore, it notes the lack of process descriptions in the literature suitable for the process of developing CMS-based web applications. We stress that existing software and Web Engineering principles are not a good fit with the goal we defined for this research. Nevertheless, the existing software and Web Engineering principles provide valuable input for the Web Engineering Method.

SQ1.2: What is an appropriate method for requirements engineering and designing web applications using a web content management system? The Web Engineering Method follows the definition by Kappel *et al.* to develop requirements engineering and design processes for CMS-based Web Engineering.

To construct a new method, we used fragments of existing methods to evaluate earlier work and combined useful parts into our existing method. The selection of our fragments is based on an extensive study of the literature surrounding existing software and web engineering methods. Combining the Unified Processes, UML-based Web Engineering (UWE), and the existing implementation best practices at a vendor of Content Management Systems, we assembled the requirements engineering processes of WEM. These descriptions provide the answer to the first part of this research question in chapter 2 and consist of three routes (we described two of them: Complex route and the Standard route). Depending on the route, different process steps are defined. For example, the resulting method uses domain modeling and use case modeling to define the complex route while a brief overview of the users and their information needs are defined in the standard route. Both routes have as a last step the reformulating of the customers wish and a requirements validation. The defined processes and deliverables provide researchers and practitioners with guidance and insights on how to organize the requirements engineering processes for CMS-based web applications.

The second part of the research question addresses designing CMS-based web applications in chapter 3. Designing web applications is a complex task and requires a different approach when using a Content Management System as a foundation. There are a several reasons for this. First, Content Management System-based web applications often involve customizations and integrations with back-office systems. Secondly, Content Management Systems have a collaborative aspect because multiple users from different departments in the enterprise can work simultaneously on the same content and functionality. A third reason is that a CMS delivers content and functionalities over multiple channels (web, mobile, e-mail, print) often in a personalized context. We apply concepts of assembly-based situational method engineering to find suitable methods for the design phase of WEM. To select the fragments we identified fourteen key concepts of Content Management Systems and selected fragments using a comparison matrix. We validated the designs through an expert validation and a case study. The overall response of the validation was positive although only the Business Process Diagrams was unanimously supported.

The answer to this research question contributes to the research described in this dissertation and consists of the Design Phase of WEM as presented in chapter 3. The design phase of WEM consists of four main activities: Conceptual Design, Architectural Design, Presentation Design, and Detailed Component Design. The architecture consists of the foundation of the Content Management System and any customizations. The resulting method and activities has led to best practices that can be used by the engineers and architects responsible for designing CMS-based web applications, in particular the Business Process Diagram.

SQ1.3: What does the realization and implementation phase look like in content management system based web engineering? We demonstrated that a Content Management System copes with most of the known challenges of Web Engineering which might explain the reason why Content Management Systems are often used as a foundation for web applications. However, Content Management Systems come with a set of specific challenges: platform specific characteristics; template analysis to match requirements and the graphical design with the CMS; separation of customizations to cope with custom-specific requirements and ensuring platform upgrades. The realization and implementation phases of WEM propose a component-based solution for the separation of customizations and a certification process to ensure future upgrades. Analyzing templates and mapping those prototypes to the CMS concepts provides an important indicator to measure the success of the project.

This research question has been answered by the definition of the Realization and Implementation phase. Both phases are modeled using the Product-Data diagrams and provide product software vendors of Content Management Systems and developers of CMS-based web applications knowledge, guidance, and best practices on how to cope with CMS-specific challenges.

SQ1.4: How to organize operations and maintenance in content management system based web engineering? As discovered in the literature, organizations should not neglect the organizational processes and structure required to effectively maintain web applications. We provide a framework for the operation and maintenance phase of web engineering consisting of guidelines for managing a Web Content Management System. The WCMS Process Framework can be used together with existing frameworks such as ITIL, ASL and BiSL. There are several reasons why operations and maintenance in a Content Management System based web applications have specific challenges compared to other information systems, those being identifying the user groups and the fit with their needs and requirements, the definition and implementation of a content maintenance strategy; keeping the content valid, accurate, current and complete; the coordination of operational content management activities and the management of external content providers. Given the differences and specific nature of CMS-based web applications, we describe the CMS Process Framework divided over three levels: the strategic level, the tactical level, and the operational level. Organizations that have a CMS-based web application can use these processes for guidance when implementing processes for operation and maintenance.

The strategic, tactical and operational processes provide an answer to this research question as elaborated on in chapter 5. These processes were evaluated by assessing two real-life case study companies to see what processes they had implemented and to assess their usefulness. The answer to this question is fundamental to the conclusion that organizations need to define processes for operation and maintenance once the CMS-based web applications have been deployed.

8.1.2 Process Innovation

The main research question concerning the process innovation of CMS-based web engineering:

RQ2: How can process innovations be included in the web engineering method?

The second research question is based on the motivation to find ways to innovate the web engineering method in terms of efficiency (less time or fewer steps to implement the system), reusability (reusing solutions or parts the solutions in similar situations), and ease of implementation (less configuration or coding). To answer the second research question, we defined two sub questions. The main conclusion based on the second research question is that we can innovate the Web Engineering Method in terms of efficiency, reusability and ease of implementation by applying model-driven paradigm and a product family line approach. The model-driven engineering method allows business users to define the requirements that will lead to a configured CMS. Thereby we skipped the ‘design phase’ for that particular component. Also this adds to the ease of implementation in terms of less configuration and coding. The product family line approach is focused on identifying reusable components and configurations in a particular market to speed up the development process which adds to the efficiency as well.

We elaborate on both questions in the following paragraphs.

SQ2.1: Can a model-driven approach innovate the web engineering method? Yes, we can improve the development processes of designing and realizing the CMS-based web application development with a model-driven approach. The purpose of this research questions was to reduce complexity and increase the transparency of the development of CMS-based web applications. Although the model-driven approach has been applied to only a part of the Content Management System (that is, the webforms), the resulting models and configuration were perceived as useful and transparent. Based on user-analysis and domain modeling, we defined the WebForm Diagram and a prototype supporting the WebForm Diagram in MetaEdit+. With this model-driven approach, requirements engineers and end-users could use the Webform diagram that leads to a configured CMS without the need for manually configuring the web form component in the CMS. It seems like this process is more efficient, more transparent and definitely easier to use than a written requirements document that needs to be interpreted by a software engineer.

The answer to this research question is yes, with the provided Webform Diagram to support the automated configuration of CMS-based web applications on which we elaborated in chapter 6. With this model, practitioners are able to model the requested Webform during the requirements phase, which leads to a configuration for the CMS. Other developers of Content Management Systems could use the described method and description of the prototype to apply the model-driven approach in their CMS.

SQ2.2: Can a product family line approach innovate the web engineering method? Yes. When vendors of Content Management Systems or services firms implementing Content Management Systems analyze their implementations and identify commonalities, they could improve the development process. We identified reusable solutions from existing WCMS implementations using Problem Frames and Structured Goal Modeling. One can conclude that the designed approach of identifying WCMS requirements in e-Business starts with identifying the e-Business model of the web initiative and then setting goals for the required user operations in the WCMS. Having the knowledge of both the available software solutions in the product platform of the WCMS and the objectives from an e-Business strategy, the requirements engineer could use this as the starting point to prioritize requirements by applying reusable software for best-practice strategy

We therefore conclude that the answer to this research questions is Yes, we can improve the development process of CMS-based web applications with a product family line approach as we described in chapter 7. The practical implications of this research is two-fold: WCMS developers now have a method to improve their product line based on e-Business models and Requirements engineers responsible for implementing the WCMS can use this model to apply reusable software and prioritize requirements.

8.2 Implications for Practice

The Web Engineering Method provides an overview of the processes and deliverables for organizations that are involved in the development and implementation processes of CMS-based web applications. Throughout this dissertation, we provided processes descriptions, examples and evaluations of web engineering based on a Content Management System. Moreover, we applied a model-driven development method and a product family approach in order to improve to the CMS-based web engineering with the goal of improving the overall processes. In doing so, this research *extends the existing knowledge on web engineering considerably with content management practices*.

Furthermore, we provided practical process descriptions that allow organizations to *systematically develop and implement CMS-based web applications*. In particular, vendors of Content Management Systems, organizations implementing Content Management Systems, and clients that are looking for a Content Management System-based web application can formulate strategies and roadmaps. Applying the concepts and processes of the Web Engineering Methods is valuable to any organization that is reassessing their Content Management System implementation practices. Although every organization involved in developing and implementing CMS-based web applications has its development methodology and best practices, we are convinced that the principles, process description and models in this dissertation provide new insights that can lead to better web engineering practices.

Vendors or developers of Content Management Systems can use the knowledge of this dissertation to apply key principles to their content management product. The identified CMS-specific challenges and the solutions we propose might be applicable in other Content Management Systems as well. Furthermore, the second part of this dissertation demonstrates how model-driven development principles and a product family approach can extend the Content Management System. To support the model-driven development principles, *we developed a prototype and elaborated on how Content Management Systems can be configured using a computer aided software engineering (CASE) tool*. The product family approach *provides developers of CMS-based web applications with industry templates that can improve the adoption of Content Management Systems in specific verticals*.

We adopted *concepts of assembly-based situational method engineering* to assemble our Web Engineering Method. Situational method engineering is a subset of method engineering that allows for the creation of methods that are tuned to the unique situation of the development project. Assembly-based situational method engineering is an approach in which method components are extracted and stored in a method base driven by situational method requirements. These theories allowed us to construct WEM combining fragments from several existing web engineering methods into our situational Web Engineering Method based on Content Management Systems.

8.3 Reflection on the Research

In this section we will reflect on our research and discuss decisions made throughout this dissertation. We will elaborate on the decisions and steps we made during our research to provide readers insight and means to value the findings of our research since the research presented in this dissertation is subject to a number of limitations that could imply that caution is required when interpreting the findings.

First of all, in chapter 2 we defined defined three routes in our Web Engineering Method: standard, complex, and migration projects. We elaborated on the first two routes but not the third, migration project. Migrations have always been a complex task, especially in web engineering given all its diversity and the ever changing requirements of web applications.

In chapter two we defined the requirements engineering processes of WEM. After a literature study, we concluded that known development methods are not capable of coping with the dynamics of CMS-based web applications or of engineering the requirements. However, existing methods do provide useful principles. Therefore, we applied situational method engineering practices to combine fragments from existing web engineering methods into WEM. After conducting a literature research, the choice was made to use the Unified Process (UP) and UML-based Web Engineering (UWE) as candidate methods. Although we do believe we have designed a suitable process for Requirements Engineering of CMS-based web engineering, the choice of candidate methods (UWE and UP) can be questioned. In choosing the candidate methods, the following considerations were taken into account: (a) their suitability of being divided into fragments and store in a method base; (b) UWE combines the strengths of the Unified Process with several Web-specific characteristics; (c) the Unified Process is a popular de facto, standard, modern software development process and it is known by the consultants who are going to use the method and (d) both methods use UML as modeling language, which is the standard notation for modeling object-oriented systems and widely accepted by the software engineering community. As can be seen in the considerations, we took several considerations into account, some of them pragmatic, when selecting the candidate methods. We looked for methods with a high possibility for adoption by the end users (hence the Unified Process) and their standardization. In our situation, for instance, the consultants that were part of the requirements engineering team were familiar (and trained) in UML as well.

For these reasons, the other web engineering methods we use in this research (e.g. WebML, OOHDM) have not been selected as candidate methods. Furthermore, the definition and selection of the fragments from the candidate methods (UWE, UP) can be considered arbitrary. Literature on situational method engineering states that the simplest way to construct a new method is to first put meaningful method fragments in a method base, to then select useful method fragments from this method base, and finally adapt and integrate them in a new method. The complexity lies in the definitions of meaningful and useful. To structure this process and to provide a transparent method we used a meta-modeling technique for the analysis of the candidate methods, the storage of relevant method fragments in a method base, the assembly of a new method from useful method fragments and the use of route map configuration to obtain situational methods. This meta-modeling technique, developed for method engineering purposes, resulted in the Process-Data Diagrams (PDD) you found throughout this dissertation. The PDDs are used in the process of analyzing, storing, selecting and assembling the method fragments. However, it is the researchers interpretation of what the meaningful and useful fragments are also from the granularity perspective (scope). Still, as a method, assembly-based situational method engineering has proven valuable throughout this dissertation and other research as can be concluded based on our findings and validation.

To further structure the selection of fragments, we proposed an adjusted method for structuring the selection of fragments in chapter three where we added two additional steps to refine the selection method for Content Management Systems. The first additional step is about decomposing the notion of Content Management Systems in order to define its key concepts. These key concepts are reflected in the developed model. The second step consists of a comparison matrix wherein selected method fragments and key concepts are compared in order to extract the relevant method fragments. Based on literature we defined fourteen key web content management concepts that are used to analyze the fragments of the selected candidate methods. This part of the analysis was operationalized with a comparison matrix. We listed feature groups in one column of the association table. This list has been made situational by means of mapping key feature groups of the WCM domain to existing functionalities of a CMS in particular. By making this list situational, it is possible for any organization to create a situational web design method that fits with the software products functionalities in relation to key feature groups of a given web domain. The ‘*x*’ in the association table implies that a method fragment or concept is needed for the design of a feature group or CMS functionality. When a cell is left blank, a feature group or CMS functionality is not addressed by any method fragment. The notation represents two extremes and does not allow for nuances or more rich information. This by itself suggests that the conclusions drawn from the comparison table might be arbitrary. Furthermore, most of the analysis of the web engineering methods and fragments within our research are based on literature study and tool analysis (e.g. ArgoUWE [Koch et al., 2003] and WebRatio [Brambilla et al., 2007]).

In chapter four, we reviewed the list of challenges present in web projects. Moreover, we stated that Content Management Systems address those challenges and that might be one of the reasons why Content Management Systems are used as a platform for web application development. Obviously, Content Management Systems do not solve all web engineering problems and we articulated a set of four challenges that are specific to Content Management Systems. However, these challenges are not yet supported by the literature and are for the most part based on insights the researcher gained from the seven years of participating at the case study company and the many unstructured discussions he had on this particular topic with product managers, project managers, marketing, and consultants. The decision to incorporate those challenges in our study might seem arbitrary and, although we lack support in the literature that would support our statement, we argue that those challenges are real, as acknowledged by the expert evaluation.

Also, because there was only a very limited number of relevant literature available on web engineering in relation to Content Management Systems at the start of our research, we defined a broad scope for this research on purpose. The goal was to cover all relevant aspects of Content Management System-based web engineering. With WEM, we believe we did provide a method that covers most of the relevant aspects. However, the trade-off is that in our effort to cover such a broad area, a number of aspects are barely touched upon and lack in-depth attention in our study.For example, given the definition of web engineering we used in this dissertation by Kappel et al, we did not provide an in-depth analysis on the role of *testing* in our method. We mentioned testing in chapter 4, but unlike the other aspects of the definition, we did not devote a great deal of energy to explicitly investigating this subject.

The operations and maintenance processes for CMS-based web applications are defined in chapter five. They consist of guidelines for managing Web Content Management Systems including processes for web content management on a strategic, tactical, and operational level. The framework is validated through a case study. Based on the case study, we conclude that the WCMS Process Framework acted as an extensive checklist for the case study organization. The framework was considered complete because the case study organization could not define the primary activities that were not described in the framework. Those activities and processes in the framework that were not dealt with in the case study organization served as a trigger to further consider and investigate the applicability.

However, we designed the processes apparently with for-profit companies in mind: the usefulness of some processes was questioned in the case study at a not-for-profit organization. The case study at a large for-profit organization showed that these processes could be useful in a larger organization and for CMS-based web applications that have commercial incentives. The applicability of each process of this framework should be assessed and evaluated for each organization, especially smaller organizations or non-profit organizations that may not require all processes, as indicated by two of the experts. Also, a remark was made that secondary processes of the overall web management, like transactions and logistics of the products and services offered through the Web, could benefit a future version of the framework.

We introduced a Model-driven Engineering (MDE) approach to configure Content Management Systems in chapter 6. The resulting WebForm Diagram consists of an abstract and concrete syntax resembling the implementation model and the mental model respectively and we developed a prototype of the model in MetaEdit+ and were able to automatically configure the Content Management System. The conversion of the output from the Prototype and the XML that was imported by the CMS was written in the general purpose languages Java and XSLT. However, MDE discipline has evolved during last few years and the hard-coded generation techniques based on general-purpose languages such as Java and XLST have been replaced by model-to-model transformation languages. Nevertheless, since we were evaluating the MDE principle to see if that is applicable in a context of configuring a Content Management System, we chose a pragmatic implementation in the preferred programming language of the resources we had at that moment. We believe that the main contribution here lies in the application of MDE principles as reflected by the goal we defined: to develop a MDE approach that allows business users to configure a CMS in order to create an interactive web application based on a user requirement model. Based on user analysis we got an idea of how the user interface would ideally look like and formalized that into a syntax that could lead to a configured CMS. Therefore, the fact that we used general purpose language, as well as the tool we used to build the prototype, MetaEdit+ was less relevant.

A second point to be made here is that the scope of this MDE application was focused to develop the Business Process or web Forms. The definition of this scope was a direct result of the validation of the Design Phase that made clear that the Business Process Diagram was useful and the visualization of the Business Process Diagram made it a suitable candidate for Model-driven Engineering. It would be interesting to see whether several other components in a Content Management System are suitable for configuration by a model-driven method.

A second approach for innovating the Web Engineering Method was presented in chapter 7, where we analyzed CMS-based web applications in a domain and identified the commonalities. The idea is that if organizations within a certain domain have similar requirements the WCMS implementation could be optimized by means of pre-configuration of the software product. Using the atomic e-business models, we identified among fifteen publishing companies the three main e-business models: intermediary, virtual community and full-service provider. A benchmark of their functionality through feature modeling revealed that these models have tendencies toward certain functional concepts. The atomic e-business models may therefore be seen as domains in which reusable solutions for these concepts can be applied. To identify a specific model, we defined objectives per atomic e-business model. Examples are ‘Intermediary: Must in some way receive payment from advertisers’. Although that statement might be true, it also could simplify the parameters that we might have taken into account to analyze the different cases. Moreover, the cases tend to have aspects of multiple e-business models. It is therefore questionable if a better understanding of the e-business model helps organization in implementing CMS-based web applications. However, the identified business requirements that were generalized into the e-business models were very useful in finding commonalities in CMS-based web applications. Furthermore, we use several theories to support the modeling and analysis of commonalities in CMS-based web applications. To name a few: the atomic e-business models, feature modeling, software product lines, goal modeling, and problem frames. These theories helped us build a strong foundation for the findings and combined provides a unique method for identifying commonalities. A consequence from using several theories was that it took more effort to explain the used methodology to the end-users at the case study company and to demonstrate the value of the structured approach. Furthermore, by applying several theories it is challenging to identify which theory provides the most contribution or value, both from a research and practical perspective.

A last limitation is a result of the fact that the research took place within a single Content Management Software company where the main researcher was employed and actively involved in the execution of the Web Engineering Method during a part of this research. This fact has led to criticisms of the validity of the research process, for example the researchers' bias in data gathering and analysis. Furthermore, researchers that are involved in the execution of models are often accused of producing results that are less generalizable. This certainly is a factor to be taken into account in this research since we designed and validated the models within a single vendor of Content Management Systems. However, we designed the models in a generic way that makes us believe that the Web Engineering Method is applicable at other Content Management Systems, or even other software systems with a focus on web development. Obviously, additional studies need to be conducted to address the defined topics of interests. Comparing results from validation within other Content Management Systems in other markets might well contribute to the generalizability of our findings. This would provide additional input to adjust and improve the Web Engineering Method. Moreover, it would result in more research data that would add more quantitative validation. And since Content Management Systems are in essence, product software for the development of dynamic web applications, we have the conviction that our concepts and approach are suited to be applied to web engineering research and other fields.

It goes without saying that additional studies are needed to further support our findings and conclusions. We elaborate on future research in the next section.

8.4 Future Work

Besides the topics we mentioned in the previous section for further research such as defining a migration route, testing of CMS-based web applications, and validating at other Content Management Systems, our research program has defined several themes that we are considering for future research. We elaborate on a few of them.

One topic we are currently researching has to do with an iterative approach to the Web Engineering Method. The Web Engineering Method, as proposed in this dissertation, might imply that the approach is strictly linear and similar to a classical waterfall model. That is not entirely true nor can we deny it. The Web Engineering Method is usually applied on the request of a client that wants a CMS-based web application within a fixed budget and within a fixed timeframe. We therefore define the (fixed) requirements to make sure that the developers know exactly what they have to deliver and in what timeframe. Although the

Web Engineering Method is a phased approach with a beginning and an end, it is usually applied in a few, larger iterations. The first iteration consists of setting up the CMS-platform with some basic configuration. The next iterations depend on user requirements. We are currently in the process of finalizing our analysis of a more iterative approach that utilizes all the components in the Content Management System in the first iteration and allows organizations to implement CMS-based web engineering in a more flexible way. A result might be that this agile approach becomes more important and that we have to revise our Web Engineering Method to add the iterative approach more explicitly.

Furthermore, the primary focus of our study is on Web Content Management Systems although this is closely related to other web applications that allow organizations to communicate with their customers and partners. As content grows and the number of online customer contact point increases as well, it becomes harder for organizations to create dynamic and relevant web applications. For years, organizations used personalization to personalize dynamic parts of the website and to make content more relevant to the visitor. Using analytical capabilities to target content based on user behavior is often referred to as Web Engagement Management (conveniently abbreviated as 'WEM'). However, with the diversification of the devices the visitors utilize to get in contact with an organization (laptop, mobile device, tablet) and the different channels they use (Facebook, Twitter, Fora, Wiki, website) it becomes a real challenge for organizations how to cope with online personalization over all relevant channels and devices. For instance, a single visitor can start its day on an iphone, at work he might visit your site on his desktop computer to review that product, in the train he might visit your site again on his ipad and at home in the evening on his laptop. An interesting topic for further research is how to manage relevant and personal content over several channels and devices.

Finally, this research is based on the ultimate goal of defining the web engineering processes with a Content Management Systems as a foundation. The outcomes of this dissertation suggest that this perspective provides a valuable contribution to the analysis and understanding of web engineering in general, especially content-driven websites and applications, now and in the near future. However, with upcoming trends such as Cloud Technologies, Social Media (e.g., Facebook, blogs, Twitter), Ubiquitous internet (internet anywhere, anytime, any device), and Semantic web, it is my expectation that Content Management Systems will focus to a larger extent on the actual Content and less on a single development platform. It will be interesting to see what role and in what form content management will be in the future. This offers many opportunities for future research because managing content will be a big challenge.

Bibliography

[Arch-int and Batanov, 2003] Arch-int, S. and Batanov, D. N. (2003). ”Development of industrial information systems on the web using business components”. *Comput. Ind.*, 50, pp. 231–250.

[Arlitt et al., 2000] Arlitt, M.; Cherkasova, L.; Dilley, J.; Friedrich, R.; and Jin, T. (2000). ”Evaluating content management techniques for Web proxy caches”. *SIGMETRICS Perform. Eval. Rev.*, 27, pp. 3–11.

[Barafort et al., 2002] Barafort, B.; Renzo, B. D.; and Merlan, O. (2002). ”Benefits Resulting from the Combined Use of ISO/IEC 15504 with the Information Technology Infrastructure Library (ITIL)”. In *Proceedings of the 4th International Conference on Product Focused Software Process Improvement*, PROFES ’02, pages 314–325, London, UK. Springer-Verlag.

[Baresi et al., 2001] Baresi, L.; Garzotto, F.; and Paolini, P. (2001). ”Extending UML for Modeling Web Applications”. *Hawaii International Conference on System Sciences*, 3, pp. 3055.

[Barna et al., 2003] Barna, P.; Frasincar, F.; Houben, G.-J.; and Vdovjak, R. (2003). ”Methodologies for Web Information System Design”. In *ITCC*, pages 420–424.

[Baskerville and Pries-Heje, 2002] Baskerville, R. and Pries-Heje, J. (2002). ”Information Systems Development at Internet Speed : A New Paradigm In The Making!”. In *Proceedings of the 10th European Conference on Information Systems, Information Systems and the Future of the Digital Economy, ECIS 2002, Gdansk, Poland, June 6-8, 2002*.

[Baumeister et al., 1999] Baumeister, H.; Koch, N.; and Mandel, L. (1999). ”Towards a UML extension for hypermedia design”. In *Proceedings of the 2nd international conference on The unified modeling language: beyond the standard*, UML’99, pages 614–629, Berlin, Heidelberg. Springer-Verlag.

[Beecham et al., 2005] Beecham, S.; Hall, T.; Britton, C.; Cottee, M.; and Rainer, A. (2005). ”Using an expert panel to validate a requirements process improvement model”. *Journal of Systems and Software*, 76(3), pp. 251–275.

[Bleistein et al., 2004a] Bleistein, S. J.; Aurum, A.; Cox, K.; and Ray, P. K. (2004a). ”Strategy-Oriented Alignment in Requirements Engineering: Linking Business Strategy to Requirements of e-Business Systems using the SOARE Approach”. *Journal of Research and Practice in Information Technology*, 36(4).

[Bleistein et al., 2004b] Bleistein, S. J.; Cox, K.; and Verner, J. M. (2004b). ”Requirements Engineering for e-Business Systems: Integrating Jackson Problem Diagrams with Goal Modeling and BPM”. In *APSEC*, pages 410–417.

[Boiko, 2001] Boiko, B. (2001). *Content Management Bible*. John Wiley & Sons, Inc., New York, NY, USA.

[Bolchini and Paolini,] Bolchini, D. and Paolini, P. ”Capturing web application requirements through goal-oriented analysis”. In *Proceedings of the Workshop on Requirements Engineering (WER 02)*, pages 16–28. Citeseer.

[Bosch et al., 2002] Bosch, J.; Florijn, G.; Greefhorst, D.; Kuusela, J.; Obbink, J. H.; and Pohl, K. (2002). ”Variability Issues in Software Product Lines”. In *PFE ’01: Revised Papers from the 4th International Workshop on Software Product-Family Engineering*, pages 13–21, London, UK. Springer-Verlag.

[Brambilla, 2006] Brambilla, M. (2006). ”Generation of WebML web application models from business process specifications”. In *ICWE ’06: Proceedings of the 6th international conference on Web engineering*, pages 85–86, New York, NY, USA. ACM.

[Brambilla et al., 2007] Brambilla, M.; Comai, S.; Fraternali, P.; and Matera, M. (2007). *Designing Web Applications with Webml and Webratio*.

[Brambilla et al., 2008] Brambilla, M.; Preciado, J. C.; Linaje, M.; and Sanchez-Figueroa, F. (2008). ”Business Process-Based Conceptual Design of Rich Internet Applications”. In *ICWE ’08: Proceedings of the 2008 Eighth International Conference on Web Engineering*, pages 155–161, Washington, DC, USA. IEEE Computer Society.

[Brinkkemper, 1996] Brinkkemper, S. (1996). ”Method Engineering: Engineering of Information Systems Development Methods and Tools”. *Journal of Information and Software Technology*, 38(4), pp. 275–280.

[Brinkkemper et al., 1999] Brinkkemper, S.; Saeki, M.; and Harmsen, F. (1999). ”Meta-modelling based assembly techniques for situational method engineering”. *Information Systems*, 24(3), pp. 209–228.

[Brownsword et al., 2000] Brownsword, L.; Oberndorf, T.; and Sledge, C. A. (2000). ”Developing New Processes for COTS-Based Systems”. *IEEE Software*, 17, pp. 48–55.

[Carson, 1986] Carson, J. S. (1986). ”Convincing Users of Model’s Validity is Challenging Aspect of Modeler’s Job”. *Industrial Engineering*, 18(6), pp. 74–85.

[Castro et al., 2004] Castro, V.; Marcos, E.; and Cáceres, P. (2004). ”A user service oriented method to model web information systems”. *Web Information Systems–WISE 2004*, pages 41–52.

[Ceri et al., 2000] Ceri, S.; Fraternali, P.; and Bongio, A. (2000). ”Web Modeling Language (WebML): a modeling language for designing Web sites”. *Comput. Netw.*, 33, pp. 137–157.

[Ceri et al., 2003] Ceri, S.; Fraternali, P.; Bongio, A.; Brambilla, M.; and Matera, M. (2003). *Designing Data Intensive Web Applications*. Morgan Kaufmann.

[Ceri et al., 1999] Ceri, S.; Fraternali, P.; and Paraboschi, S. (1999). ”Data-Driven, One-To-One Web Site Generation for Data-Intensive Applications”. In *VLDB ’99: Proceedings of the 25th International Conference on Very Large Data Bases*, pages 615–626, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Clements and Northrop, 2001] Clements, P. C. and Northrop, L. (2001). *Software Product Lines: Practices and Patterns*. SEI Series in Software Engineering. Addison-Wesley.

[Coalition, 1999] Coalition, W. M. (1999). ”Workflow Management Coalition Terminology Glossary”.

[Cooper, 2007a] Cooper, A. (2007a). *About Face 3: The Essentials of Interaction Design*. Wiley, New York.

[Cooper, 2007b] Cooper, A. (2007b). *About Face 3: The Essentials of Interaction Design*. Wiley, New York.

[Czarnecki et al., 2005] Czarnecki, K.; Helsen, S.; and Eisenecker, U. (2005). ”Staged configuration through specialization and multi-level configuration of feature models”. In *Software Process Improvement and Practice*, page 2005.

[de Troyer, 2008] de Troyer, O. (2008). ”Audience-Driven Design Approach for Web Systems”. *Encyclopedia of Information Science and Technology*, pages 274–278.

[DeBaud and Schmid, 1999] DeBaud, J.-M. and Schmid, K. (1999). ”A Systematic Approach to Derive the Scope of Software Product Lines”. In *ICSE*, pages 34–43.

[Dul and Hak, 2007] Dul, J. and Hak, T. (2007). *Case Study Methodology in Business Research*. Butterworth-Heinemann.

[Eirinaki and Vazirgiannis, 2003] Eirinaki, M. and Vazirgiannis, M. (2003). ”Web mining for web personalization”. *ACM Trans. Inter. Tech.*, 3(1), pp. 1–27.

[Escalona and Koch, 2004] Escalona, M. and Koch, N. (2004). ”Requirements engineering for web applications-a comparative study”. *Journal of Web Engineering*, 2, pp. 193–212.

[Fein et al., 1993] Fein, R. M.; Olson, G. M.; and Olson, J. S. (1993). ”A mental model can help with learning to operate a complex device”. In *CHI ’93: INTERACT ’93 and CHI ’93 conference companion on Human factors in computing systems*, pages 157–158, New York, NY, USA. ACM.

[Fernandez-Iglesias et al., 2005] Fernandez-Iglesias, M.; Alvarez-Sabucedo, L.; Lojo-Silva, C.; and Anido-Rifon, L. (2005). ”Generating high quality printouts from content management systems: a cost-effective approach”. *Software: Practice and Experience*, 35(11), pp. 1007–1026.

[Fuller et al., 2004] Fuller, H.; Institute, P.; and (US), G. S. (2004). *Becoming a Citizen-Centered Government Through Best Practices in Web Management*. Citeseer.

[Gartner, 2010] Gartner (2010). ”Gartner Magic Quadrant for Web Content Management – 2010”.

[Garzotto et al., 1991] Garzotto, F.; Paolini, P.; and Schwabe, D. (1991). ”HDM – a model for the design of hypertext applications”. In *HYPERTEXT ’91: Proceedings of the third annual ACM conference on Hypertext*, pages 313–328, New York, NY, USA. ACM.

[Georgakopoulos et al., 1995] Georgakopoulos, D.; Hornick, M. F.; and Sheth, A. P. (1995). ”An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure”. *Distributed and Parallel Databases*, 3(2), pp. 119–153.

[Gilbane, 2000] Gilbane, F. (2000). ”What is Content Management?”. Gilbane, http://gilbane.com/gilbane$_report.pl/6(last visited 07/03/2011)$.

[Ginige and Murugesan, 2001a] Ginige, A. and Murugesan, S. (2001a). ”The Essence of Web Engineering;Managing the Diversity and Complexity of Web Application Development”. *IEEE Multimedia*, 8, pp. 22–25.

[Ginige and Murugesan, 2001b] Ginige, A. and Murugesan, S. (2001b). ”Guest Editors’ Introduction: Web Engineering - An Introduction.”. *IEEE MultiMedia*, pages 14–18.

[Gnaho, 2001] Gnaho, C. (2001). ”Web-Based Information Systems Development-A User Centered Engineering Approach”. *Web Engineering*, pages 105–118.

[Gómez and Cachero, 2003] Gómez, J. and Cachero, C. (2003). ”OO-H Method: extending UML to model web interfaces”. In *Information modeling for internet applications*, pages 144–173, Hershey, PA, USA. IGI Publishing.

[Grahlmann et al., 2010] Grahlmann, K.; Hilhorst, C.; van Amerongen, S.; Helms, R.; and Brinkkemper, S. (2010). ”Impacts of implementing Enterprise Content Management systems”. In *Proceedings of the 18th European Conference on Information Systems*, Pretoria, South Africa.

[Grossniklaus and Norrie, 2002] Grossniklaus, M. and Norrie, M. C. (2002). ”Information Concepts for Content Management”. In *Proceedings of the Third International Conference on Web Information Systems Engineering (Workshops) - (WISEw’02)*, WISEW ’02, pages 150–, Washington, DC, USA. IEEE Computer Society.

[Group, 2005] Group, O. M. (2005). ”Unified Modeling Language: Infrastructure, Version 2.0”. *http://www.omg.org/docs/formal/05-07-05.pdf*.

[Hevner et al., 2004] Hevner, A. R.; March, S. T.; Park, J.; and Ram, S. (2004). ”Design science in information systems research”. *Management Information Systems Quarterly*, 28(1), pp. 75–106.

[Hoick, 2003] Hoick, J. (2003). ”4 perspectives on Web information systems”. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, page 8. IEEE.

[Holck, 2003] Holck, J. (2003). ”4 Perspectives on Web Information Systems”. *Hawaii International Conference on System Sciences*, 8, pp. 265b.

[Houben et al., 2008] Houben, G.-J.; Sluijs, K. v. d.; Barna, P.; Broekstra, J.; Casteleyn, S.; Fiala, Z.; and Frasincar, F. (2008). ”HERA”. In Rossi, G.; Pastor, O.; Schwabe, D.; and Olsina, L., editors, *Web Engineering: Modelling and Implementing Web Applications*, Human-Computer Interaction Series, pages 263–301. Springer London. 10.1007/978-1-84628-923-1$_1$0.

[Hysell, 1998] Hysell, D. (1998). ”Meeting the needs (and preferences) of a diverse World Wide Web audience”. In *Proceedings of the 16th annual international conference on Computer documentation*, SIGDOC ’98, pages 164–172, New York, NY, USA. ACM.

[Institute, 2007] Institute, I. G. (2007). *Cobit 4.1*. ISA.

[Jablonski and Meiler, 2002] Jablonski, S. and Meiler, C. (2002). ”Web-Content-Managementsysteme”. *Informatik Spektrum*, 25(2), pp. 101–119.

[Jackson, 2001] Jackson, M. (2001). *Problem frames: analyzing and structuring software development problems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Jacobson et al., 1999] Jacobson, I.; Booch, G.; and Rumbaugh, J. (1999). *The unified software development process*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Jeary and Phalp, 2004] Jeary, S. and Phalp, K. (2004). ”On the applicability of problem frames to web-based business applications”. In *Proceedings of the 1st International Workshop on Applications and Advances of Problem Frames*.

[Jeenicke et al., 2003] Jeenicke, M.; Bleek, W.; and Klischewski, R. (2003). ”Revealing web user requirements through e-prototyping”. In *Proceedings of the Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE 03), San Francisco, USA*. Citeseer.

[Kang et al., 1990] Kang, K. C.; Cohen, S. G.; Hess, J. A.; Novak, W. E.; and Peterson, A. S. (1990). ”Feature-Oriented Domain Analysis (FODA) Feasibility Study”. Technical report, Carnegie-Mellon University Software Engineering Institute.

[Kappel et al., 2006] Kappel, G.; Prll, B.; Reich, S.; and Retschitzegger, W. (2006). *Web Engineering: The Discipline of Systematic Development of Web Applications*. Wiley, New York, NY.

[Kautz and Nørbjerg, 2003] Kautz, K. and Nørbjerg, J. (2003). ”Persistent problems in information systems development: the case of the World Wide Web”. In *Proceedings of ECIS*, pages 19–21. Citeseer.

[Kelly et al., 1996] Kelly, S.; Lyytinen, K.; and Rossi, M. (1996). ”MetaEdit+: A Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment”. In *CAiSE ‘96: Proceedings of the 8th International Conference on Advances Information System Engineering*, pages 1–21, London, UK. Springer-Verlag.

[Kent, 2002] Kent, S. (2002). ”Model Driven Engineering”. In *IFM*, pages 286–298.

[Koch, 1999] Koch, N. (1999). ”A Comparative Study of Methods for Hypermedia Development”. Technical Report 9905, Institut fr Informatik der LMU.

[Koch and Kraus, 2002] Koch, N. and Kraus, A. (2002). ”The expressive Power of UML-based Web Engineering”. In *Proceedings of IWWOST02*, pages 105–119.

[Koch et al., 2008] Koch, N.; Meliá, S.; Moreno, N.; Pelechano, V.; Sanchez, F.; and Vara, J. M. (2008). ”Model-Driven Web Engineering”. *Upgrade-Novtica Journal (English and Spanish), Council of European Professional Informatics Societies (CEPIS)*, IX(2), pp. 40–45.

[Koch et al., 2003] Koch, N.; Moser, F.; and Zhang, G. (2003). ”ArgoUWE: A CASE Tool for Web Applications”.

[Kruchten, 1995] Kruchten, P. (1995). ”The 4+1 View Model of Architecture”. *IEEE Software*, 12(6), pp. 42–50.

[Larsen et al., 2006] Larsen, M. H.; Pedersen, M. K.; and Andersen, K. V. (2006). ”IT Governance: Reviewing 17 IT Governance Tools and Analysing the Case of Novozymes A/S”. *Hawaii International Conference on System Sciences*, 8, pp. 195c.

[Lee et al., 2004] Lee, C.; Suh, W.; and Lee, H. (2004). ”Implementing a community web site: a scenario-based methodology”. *Information and Software Technology*, 46(1), pp. 17–33.

[Lee, 1999] Lee, S. (1999). ”Usability testing for developing effective interactive multimedia software: Concepts, dimensions, and procedures”. *Educational Technology & Society*, 2(2), pp. 1–13.

[Lee and Shirani, 2004a] Lee, S. and Shirani, A. (2004a). ”A component based methodology for Web application development”. *Journal of systems and software*, 71(1-2), pp. 177–187.

[Lee and Shirani, 2004b] Lee, S. C. and Shirani, A. I. (2004b). ”A component based methodology for web application development”. *J. Syst. Softw.*, 71(1-2), pp. 177–187.

[Looijen, 1998] Looijen, M. (1998). *Information Systems: Management, Control and Maintenance*. Kluwer, Deventer.

[Lowe, 1999] Lowe, D. (1999). *Hypermedia and the Web: An Engineering Approach*. John Wiley & Sons, Inc., New York, NY, USA.

[Lowe and Henderson-Sellers, 2001] Lowe, D. and Henderson-Sellers, B. (2001). ”Characteristics of Web Development Processes”. In *proceedings of the 2001 International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*.

[McKeever, 2003] McKeever, S. (2003). ”Understanding Web content management systems: evolution, lifecycle and market”. *Industrial Management and Data Systems*, 103(9), pp. 686 – 692.

[Millard and Ross, 2006] Millard, D. E. and Ross, M. (2006). ”Web 2.0: hypertext by any other name?”. In *HYPERTEXT ’06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 27–30, New York, NY, USA. ACM Press.

[Morrison et al., 2002] Morrison, M.; Morrison, J.; and Keys, A. (2002). ”Integrating web sites and databases”. *Commun. ACM*, 45, pp. 81–86.

[Murugesan et al., 2001] Murugesan, S.; Deshpande, Y.; Hansen, S.; and Ginige, A. (2001). ”Web engineering: A new discipline for development of web-based systems”. *Web Engineering*, pages 3–13.

[Muthig and Atkinson, 2002] Muthig, D. and Atkinson, C. (2002). ”Model-Driven Product Line Architectures”. In *SPLC 2: Proceedings of the Second International Conference on Software Product Lines*, pages 110–129, London, UK. Springer-Verlag.

[Nakano, 2002] Nakano, R. (2002). *Web content management: a collaborative approach*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Natu and Mendonca, 2003] Natu, S. and Mendonca, J. (2003). ”Digital asset management using a native XML database implementation”. In *CITC4 ’03: Proceedings of the 4th conference on Information technology curriculum*, pages 237–241, New York, NY, USA. ACM.

[Overmyer, 2000] Overmyer, S. P. (2000). ”What’s Different about Requirements Engineering for Web Sites?”. *Requir. Eng.*, 5(1), pp. 62–65.

[Pastor et al., 2006] Pastor, O.; Fons, J.; Pelechano, V.; and Abrahao, S. (2006). ”Conceptual Modeling of Web Applications: The OOWS Approach”. *E. Mendes and N. Mosley (eds.) Web Engineering: Theory and Practice of Metrics and Measurement for Web Development*.

[Pastor et al., 2001] Pastor, O.; Silvia Mara Abrah a.; and Fons, J. (2001). ”An Object-Oriented Approach to Automate Web Applications Development”. In *EC-Web 2001: Proceedings of the Second International Conference on Electronic Commerce and Web Technologies*, pages 16–28, London, UK. Springer-Verlag.

[Peffers et al., 2007] Peffers, K.; Tuunanen, T.; Rothenberger, M.; and Chatterjee, S. (2007). ”A Design Science Research Methodology for Information Systems Research”. *J. Manage. Inf. Syst.*, 24, pp. 45–77.

[Pereira dos Santos et al., 2009] Pereira dos Santos, R.; Werner, C. M. L.; and da Silva, M. A. (2009). ”Incorporating information of value in a component repository to support a component marketplace infrastructure”. In *Proceedings of the 10th IEEE international conference on Information Reuse & Integration*, IRI’09, pages 266–271, Piscataway, NJ, USA. IEEE Press.

[Prahalad and Hamel, 1990] Prahalad, C. K. and Hamel, G. (1990). ”The Core Competence of the Corporation”. *Harvard Business Review*.

[Pressman, 1996] Pressman, R. S. (1996). *Software Engineering: A Practitioner’s Approach*. McGraw-Hill Higher Education, 4th edition.

[Ralyté et al., 2007] Ralyté, J.; Brinkkemper, S.; and Henderson-Sellers, B. (2007). ”Situational Method Engineering: Fundamentals and Experiences”. *Proceedings of the IFIP WG 8.1 Working Conference*, 38(4), pp. XII + 368.

[Ralyté et al., 2003] Ralyté, J.; Deneckère, R.; and Roland, C. (2003). ”Towards a Generic Model for Situational Method Engineering”. In *Lecture Notes in Computer Science,*, volume 2681. Springer-Verlag.

[Ravi et al., 2009] Ravi, J.; Yu, Z.; and Shi, W. (2009). ”A survey on dynamic Web content generation and delivery techniques”. *J. Netw. Comput. Appl.*, 32(5), pp. 943–960.

[Repenning et al., 2001] Repenning, A.; Ioannidou, A.; Payton, M.; Ye, W.; and Roschelle, J. (2001). ”Using Components for Rapid Distributed Software Development”. *IEEE Software*, 18, pp. 38–45.

[Robinson, 1997] Robinson, S. (1997). ”Simulation model verification and validation: increasing the users’ confidence”. In *Proceedings of the 29th conference on Winter simulation*, WSC ’97, pages 53–59, Washington, DC, USA. IEEE Computer Society.

[Rolland et al., 1998] Rolland, C.; Achour, C. B.; Cauvet, C.; Ralyté, J.; Sutcliffe, A.; Maiden, N.; Jarke, M.; Haumer, P.; Pohl, K.; Dubois, E.; and Heymans, P. (1998). ”A proposal for a scenario classification framework”. *Requir. Eng.*, 3(1), pp. 23–47.

[Rolland and Prakash, 2000] Rolland, C. and Prakash, N. (2000). ”Bridging the Gap Between Organisational Needs and ERP Functionality”. *Requir. Eng.*, 5(3), pp. 180–193.

[Rothfuss and Ried, 2002] Rothfuss, G. and Ried, C. (2002). *Content Management mit XML*. Number 3. Springer.

[Schmid et al., 2006] Schmid, K.; Krennrich, K.; and Eisenbarth, M. (2006). ”Requirements Management for Product Lines: Extending Professional Tools”. In *SPLC ’06: Proceedings of the 10th International on Software Product Line Conference*, pages 113–122, Washington, DC, USA. IEEE Computer Society.

[Schmidt, 2006] Schmidt, D. C. (2006). ”Guest Editor’s Introduction: Model-Driven Engineering”. *Computer*, 39(2), pp. 25–31.

[Schuster and Wilhelm, 2000] Schuster, E. and Wilhelm, S. (2000). ”Content Management - Aktuelles Schlagwort”. *Informatik Spektrum*, 23(6), pp. 373–375.

[Schwabe and Rossi, 1995] Schwabe, D. and Rossi, G. (1995). ”The object-oriented hypermedia design model”. *Commun. ACM*, 38(8), pp. 45–46.

[Schwabe et al., 1996] Schwabe, D.; Rossi, G.; and Barbosa, S. D. J. (1996). ”Systematic hypermedia application design with OOHDM”. In *HYPERTEXT ’96: Proceedings of the the seventh ACM conference on Hypertext*, pages 116–128, New York, NY, USA. ACM.

[Schwickert, 2004] Schwickert, A. (2004). ”Dezentrales Web Content Management”. In *Impulse aus der Wirtschaftsinformatik: 5. Liechtensteinisches Wirtschaftsinformatik-symposium an der Fachhochschule Liechtenstein*, page 247. Physica-Verlag Heidelberg.

[Smith, 2003] Smith, H. (2003). ”Business process management–the third wave: business process modelling language (bpml) and its pi-calculus foundations”. *Information & Software Technology*, 45(15), pp. 1065–1069.

[Soffer et al., 2003] Soffer, P.; Golany, B.; and Dori, D. (2003). ”ERP modeling: a comprehensive approach”. *Inf. Syst.*, 28(6), pp. 673–690.

[Standing, 2002] Standing, C. (2002). ”Methodologies for developing Web applications”. *Information and software technology*, 44(3), pp. 151–159.

[Taylor et al., 2002] Taylor, M.; McWilliam, J.; Forsyth, H.; and Wade, S. (2002). ”Methodologies and website development: a survey of practice”. *Information and Software Technology*, 44(6), pp. 381–391.

[Tolvanen and Rossi, 2003] Tolvanen, J.-P. and Rossi, M. (2003). ”MetaEdit+: defining and using domain-specific modeling languages and code generators”. In *OOPSLA ‘03: Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 92–93, New York, NY, USA. ACM.

[Troyer and Leune, 1998] Troyer, O. D. and Leune, C. (1998). ”WSDM: a user centered design method for Web sites”. *Comput. Netw. ISDN Syst.*, 30(1-7), pp. 85–94.

[Turban et al., 2000] Turban, E.; Wetherbe, J.; and McLean, E. (2000). *Information Technology for Management: Making Connections for Strategic Advantage*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition.

[Vaishnavi and Kuechler, 2005] Vaishnavi, V. and Kuechler, B. (2005). ”Design Research in Information Systems”. Association for Information Systems, http://desrist.org/design-research-in-information-systems/ (last updated 16-08-2009).

[van Berkum et al., 2004] van Berkum, M.; Brinkkemper, S.; and Meyer, A. (2004). ”A Combined Runtime Environment and Web-Based Development Environment for Web Application Engineering”. In *Proceedings of the Advanced Information Systems Engineering. CAiSE04*, pages 307–321. LNCS 3084.

[van de Weerd et al., 2006] 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*, 11(5), pp. 521–538.

[van der Hoek et al., 1997] van der Hoek, A.; Hall, R. S.; Heimbigner, D.; and Wolf, A. L. (1997). ”Software Release Management”. In Jazayeri, M. and Schauer, H., editors, *Proceedings of the Sixth European Software Engineering Conference (ESEC/FSE 97)*, pages 159–175. Springer–Verlag.

[van der Pols, 2004] van der Pols, R. (2004). ”ASL A Framework for Application Management”.

[van der Pols, 2007] van der Pols, R. (2007). ”BiSL A Framework for Functional Management and Information Management”.

[van Deursen et al., 2000] van Deursen, A.; Klint, P.; and Visser, J. (2000). ”Domain-specific languages: an annotated bibliography”. *SIGPLAN Not.*, 35(6), pp. 26–36.

[van Lamsweerde, 2001] van Lamsweerde, A. (2001). ”Goal-Oriented Requirements Engineering: A Guided Tour”. In *RE*, page 249.

[van Slooten and Hodes, 1996] van Slooten, K. and Hodes, B. (1996). ”Characterizing IS development projects”. In *Proceedings of the IFIP TC8, WG8.1/8.2 working conference on method engineering on Method engineering : principles of method construction and tool support*, pages 29–44, London, UK, UK. Chapman & Hall, Ltd.

[Vdovjak et al., 2003] Vdovjak, R.; Frasincar, F.; Houben, G.-J.; and Barna, P. (2003). ”Engineering semantic web information systems in hera”.

[Vdovjak and Houben, 2005] Vdovjak, R. and Houben, G.-J. (2005). ”A Model-Driven Approach for Designing Distributed Web Information Systems”. In *ICWE*, pages 453–464.

[Vidgen et al., 2001] Vidgen, R.; Goodwin, S.; and Barnes, S. (2001). ”Web Content Management”. In *Proceedings of the 14th International Electronic Commerce Conference*, pages 465–480.

[Vigder and Dean, 1997] Vigder, M. R. and Dean, J. (1997). ”An architectural approach to building systems from COTS software components”. In *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research*, CASCON ’97, pages 22–. IBM Press.

[Weill and Vitale, 2001] Weill, P. and Vitale, M. R. (2001). *Place to Space: Migrating to Ebusiness Models*. Harvard Business School Press, Boston, MA, USA.

[Weiss and Lai, 1999] Weiss, D. M. and Lai, C. T. R. (1999). *Software product-line engineering: a family-based software development process*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Wimmer et al., 2007] Wimmer, M.; Schauerhuber, A.; Schwinger, W.; and Kargl, H. (2007). ”On the Integration of Web Modeling Languages: Preliminary Results and Future Challenges”. In *Proceedings of the 3rd International Workshop on Model-Driven Web Engineering (WMDE2007)*.

[Xu and Brinkkemper, 2007] Xu, L. and Brinkkemper, S. (2007). ”Concepts of product software”. *EJIS*, 16(5), pp. 531–541.

[Yin, 2002] Yin, R. K. (2002). *Case Study Research: Design and Methods, Third Edition, Applied Social Research Methods Series, Vol 5*. Sage Publications, Inc, 3rd edition.

Publication List

[Souer et al., 2005] Souer, J.; van de Weerd, I.; Versendaal, J.; and Brinkkemper, S. (2005). ”Situational Requirements Engineering for the Development of Content Management System-based Web Applications”. Technical Report, Utrecht University, UU-CS-2005-050.

[Souer et al., 2007a] Souer, J.; Honders, P.; Versendaal, J.; and Brinkkemper, S. (2007a). ”Defining operations and maintenance in web engineering: A framework for CMS-based Web applications”. In *IEEE ACM Second International Conference on Digital Information Management ICDIM 2007*, pages 430 – 435.

[Souer et al., 2007b] Souer, J.; van de Weerd, I.; Versendaal, J.; and Brinkkemper, S. (2007b). ”Situational requirements engineering for the development of Content Management System-based web applications”. *International Journal of Web Engineering and Technology (IJWET)*, 3(4), pp. 420 – 440.

[Souer and van Mierloo, 2008] Souer, J. and van Mierloo, M. (2008). ”A Component Based Architecture for Web Content Management: Runtime Deployable WebManager Component Bundles”. In *Proceedings of the Eighth International Conference on Web Engineering (ICWE08)*, pages 366 – 369.

[Souer et al., 2008a] Souer, J.; Honders, P.; Versendaal, J.; and Brinkkemper, S. (2008a). ”A Framework for Web Content Management System Operation and Maintenance”. *Journal of Digital Information Management (JDIM)*, pages 324 – 331.

[Souer et al., 2008b] Souer, J.; Luinenburg, L.; Versendaal, J.; van de Weerd, I.; and Brinkkemper, S. (2008b). ”A Design Method and Application for Web Content Management Implementations”. In *Proceedings of the 10th International Conference on Information Integration and Web-based Application & Services (iiWAS08)*, pages 351 – 358.

[Souer and Kupers, 2009] Souer, J. and Kupers, T. (2009). ”Towards a Pragmatic Model Driven Engineering Approach for the Development of CMS-based Web Applications”. In *Proceedings of the 5th Model Driven Web Engineering Workshop (MDWE09)*, pages 31 – 45.

[Souer et al., 2009] Souer, J.; Kupers, T.; Helms, R.; and Brinkkemper, S. (2009). ”Model Driven Web Engineering for the Automated Configuration of Web Content Management Systems”. In *Proceedings of the 9th International Conference on Web Engineering (ICWE09)*, pages 121 – 135.

[Souer and Joor, 2010] Souer, J. and Joor, D.-J. (2010). ”An Approach to Identify Commonalities In Web Application Engineering for a Web Content Management System”. In *Proceedings of the 12th International Conference on Information Integration and Web-based Application & Services (iiWAS10)*, pages 556–563.

[Souer et al., 2011a] Souer, J.; Joor, D.-J.; Helms, R.; and Brinkkemper, S. (2011a). ”Identifying Commonalities in Web Content Management System Engineering”. *International Journal of Web Information Systems*, page Accepted for Publication.

[Souer et al., 2011b] Souer, J.; Urlings, T.; Helms, R.; and Brinkkemper, S. (2011b). ”Engineering Web Information Systems: A Content Management System-based Approach”. *Accepted for publication at the International Conference on Information Integration and Web-based Application & Services (iiWAS11)*.

[van de Weerd et al., 2005] van de Weerd, I.; Souer, J.; Versendaal, J.; and Brinkkemper, S. (2005). ”Situational requirements engineering of web content management implementations”. In *Proceedings of the 1st International Workshop on Situational Requirements Engineering Processes (SREP05)*. University of Limerick.

[Janssen et al., 2006] Janssen, S.; Souer, J.; and Brinkkemper, S. (2006). ”A Case Study of a product software vendor’s Customer Configuration Updating Process: GX Creative Online Development”. Technical Report, Utrecht University, UU-CS-2006-044.

[van de Weerd et al., 2006] van de Weerd, I.; Souer, J.; Versendaal, J.; and Brinkkemper, S. (2006). ”Situational requirements engineering of web content management implementations”. *Software Process: Improvement and Practice*, 3(11), pp. 521 – 538.

[Luinenburg et al., 2008a] Luinenburg, L.; Jansen, S.; Souer, J.; Brinkkemper, S.; and Weerd, I. (2008a). ”An Approach to Creating Design Methods for the Implementation of Product Software: The Case of Web Information Systems”. In *ER ’08: Proceedings of the ER 2008 Workshops (CMLSA, ECDM, FP-UML, M2AS, RIGiM, SeCoGIS, WISM) on Advances in Conceptual Modeling*, pages 426 – 436.

[Luinenburg et al., 2008b] Luinenburg, L.; Jansen, S.; Souer, J.; van de Weerd, I.; and Brinkkemper, S. (2008b). ”Designing Web Content Management Systems Using the Method Association Approach”. In *Proceedings of the 4th International Workshop on Model-Driven Web Engineering (MDWE 2008)*, pages 106 – 120.

[Jansen et al., 2012] Jansen, S.; Brinkkemper, S.; Souer, J.; and Luinenburg, L. (2012). ”Shades of Gray: Opening up a Software Producing Organization with the Open Software Enterprise Model”. In *Journal of Systems and Software*, Accepted for publication.

Summary

Web engineering is the application of systematic and quantifiable approaches (concepts, methods, techniques, tools) to cost-effective requirements analysis, design, implementation, testing, operation, and maintenance of high quality web applications [Kappel et al., 2006]. Over the past years, Content Management Systems (CMS) have emerged as an important foundation for the web engineering process. CMS can be defined as a tool for the creation, editing and management of web information in an integral way. A CMS appears to be of value since it provides a standardized platform for web development with a set of out-off-the-box functionalities which allow the users (or business owners of the website, often the marketing/communications department) to manage the website without the need for technical knowledge or support from the IT department. However, developing CMS-based web applications can be complex to implement because of the dual lifecycle of CMS-implementations, matching requirements with software product capabilities, customizations to meet end-user requirements and maintenance processes once the application has been implemented. To overcome the numerous implementation failures, our research is focused on the web engineering process for the development of CMS-based web applications.

The hypothesis on which my research is based is that this can be solved by providing methodical support to organizations in the form of an integrated development and an implementation model that provides the necessary activities and deliverables to guide the creation of CMS-based web applications. We propose the web Engineering Method (WEM) as a situational development method for the definition, design, realization, implementation and maintenance of high quality CMS-based web applications. We use the situational method-engineering approach to gather relevant method fragments coming from several web engineering methods and combined them into a new approach.

WEM starts with the definition phase in which the CMS-based web application is specified. To get a clear understanding of the required solution, we propose that an organization first needs to have a clear understanding of the business goals and

make a solid plan in line with high level requirements in an acquisition phase to select the CMS and the definition phase to define the requirements. The process consists of two routes, a standard route and a complex route, with their own set activities and deliverables such as user modeling, domain modeling and application modeling.

A CMS is product software and therefore provides a set of components and a platform as a foundation for the architecture of the CMS-based web application. For that reason, the application architecture of the CMS-based web application is based on the CMS itself. The activities in the design phase however are more focused on customizations of the CMS and specific configurations of it. The design phase of WEM addresses fourteen identified key CMS concepts and consists of the main activities: Conceptual Design, Architecture Design, Presentation Design and Detailed Component Design.

A CMS provides solutions to common identified web engineering challenges, but comes with some CMS-specific challenges on its own. The Realization phase consists of Prioritization, Development of templates, Selection, Adjustment and Customization of components, and Quality Assurance. The Implementation phase consists of transferring the development environment to the production environment, End-user training and Content Management of the site, Acceptance test, Go-Live and monitoring of the CMS-based web application.

We conclude the development process with the inclusion of maintenance processes in the organization and described an operations and maintenance framework. We defined specific CMS-processes on a strategic, tactical and operational level and present how it can be used in conjunction with existing frameworks such as ITIL, ASL and BiSL.

We explored two specific methods to further develop the process of WEM. The concepts of Model Driven web Engineering are explored with the purpose of improving the realization process by automating the configuration of the CMS. We describe how we created a model driven approach in the design phase and the automated configuration of CMS-based web engineering. Finally, we explore the concepts of product verticalization. We identified commonalities in implementations of CMS-based web applications in a specific market sector and provide a method for the improvement of WEM for given market sectors.

The applicability of WEM was validated in several case studies, questionnaires and expert reviews. To validate the model-driven approach, we developed a prototype and .

The main contribution is a detailed overview of WEM: the implementation process of CMS-based web applications in chapter 2 through chapter 5. There are many publications in the fields of web engineering, but a complete description on how to cope with a CMS, as a foundation for web applications does not exist. The process descriptions tell the scientific community how we combined existing methods and created a new method for the development of CMS-based web applications using a situational method engineering approach. We provide an overview of key concepts of Content Management systems in chapter 5 that we take into account during the process description. Based on the key concepts we compared, analysed and selected method fragments with respect to WCM concepts. We present a number of improvements for developing CMS-based web applications based on a model driven approach where we specified and developed a prototype for the configuration of CMS-based web applications. Moreover, we present an improvement in the dual lifecycle of content management systems where we identify commonalities in CMS-based web applications and incorporate it into the product platform.

These contributions provide organizations that implement Content Management Systems with guidelines and best practices, which allow them to improve the overall quality of the CMS-based web application. Simultaneously, software developers creating Content Management Systems can improve the software product and the way it is implemented using a model-driven approach or product-verticalization. Organizations looking for a CMS-based web application have a detailed process overview of all the steps involved, best practices, potential pitfalls and suggestions for improvement.

Nederlandse Samenvatting

Web Engineering is het proces van het systematisch toepassen van concepten, methoden en technieken om op een efficiënte manier kwalitatief hoogstaande web applicaties te beschrijven, ontwerpen, implementeren, testen en onderhouden [Kappel et al., 2006]. De laatste jaren wordt het proces van Web Engineering steeds meer ondersteund door specifieke product software genaamd Content Management Systeem (CMS). Een CMS is een software systeem voor het ontwikkelen, beheren en onderhouden van web gerelateerde content. Een van de redenen waarom een CMS steeds meer gebruikt wordt, is dat het voorziet in een gestandaardiseerd platform voor het efficiënt ontwikkelen van web applicaties. Het CMS bevat bovendien functionaliteiten die een gebruiker in staat stelt om de websites te beheren zonder dat daar technische kennis of een traditionele IT afdeling voor noodzakelijk is. Hoe mooi dit ook klinkt, ook de ontwikkeling van op CMS-gebaseerde web applicaties heeft zo zijn valkuilen en kan complex zijn. Een belangrijke oorzaak is te vinden in het duale karakter van CMS-implementaties. Dit duale karakter is te omschrijven als de ontwikkeling van het CMS product zelf enerzijds (bijvoorbeeld door een ontwikkelteam bij een software bedrijf) en de toepassing van het product in de praktijk bij een organisatie voor de ontwikkeling van een website anderzijds. De wensen die klanten hebben, moeten zo veel mogelijk aansluiten bij de functionaliteiten die het CMS te bieden heeft; aanpassingen aan het product moeten geminimaliseerd worden om upgrade problemen te voorkomen en het onderhoud op het CMS overzichtelijk te houden. Om deze problemen inzichtelijk te maken en mogelijke oplossingen aan te dragen, is dit onderzoek gericht op het ontwerpen van een Web Enigeering proces voor de ontwikkeling van op CMS-gebaseerde web applicaties. Het resultaat van dit onderzoek is de Web Engineering Method (WEM) als een situationele methode voor het definiëren, ontwerpen, realiseren, implementeren en onderhouden van op CMS-gebaseerde web applicaties. We hebben situationele method-engineering als proces gebruikt om relevante fragmenten te combineren uit bestaande Web Engineering methoden en hebben deze gecombineerd in WEM.

WEM begint bij de eerste fase van het ontwikkel proces: de definitiefase. In de definitie fase worden alle wensen en eisen vastgelegd om een goed begrip te krijgen van de uiteindelijk oplossing. Het proces bevat twee routes, een ‘standaard-route’ en een ‘complexe-route’, die elk hun eigen verzameling aan activiteiten en resultaten hebben zoals het modelleren van type gebruikers, een domein model en een applicatie model. De ontwerpfase is interessant omdat het CMS als productsoftware een groot deel van de basisarchitectuur levert. De activiteiten in de ontwerpfase zijn daarom meer gericht op het configureren van componenten en het ontwikkelen van productaanpassingen om aan specifieke klantwensen te kunnen voldoen. De ontwerpfase heeft op basis van veertien CMS-concepten de volgende hoofdactiviteiten: conceptueel ontwerp, architectuur, presentatie ontwerp en gedetailleerd component ontwerp. Het gestandaardiseerde platform van een CMS biedt oplossingen voor veel voorkomende ontwikkelproblemen. Echter, het CMS heeft een aantal CMS-specifieke uitdagingen die aandacht vereisen. De realisatiefase omvat prioritering, ontwikkeling van templates, selectie, aanpassing en ontwikkeling van componenten en het testen van het product. De implementatiefase begint met het overzetten van de ontwikkelde omgeving naar de product omgeving, het trainen van de eindgebruikers in het CMS en het voorzien van de website met de content. Dan zal de acceptatie test plaatsvinden en is de site gereed voor het zogeheten ‘go-live’-moment. Een CMS kan grootschalige websites ondersteunen waarbij een nauwkeurige monitoring van de productieomgeving noodzakelijk is om de kwaliteit te kunnen waarborgen. Tot slot ronden we het ontwikkelproces af met onderhoudsprocessen die een organisatie in acht kunnen nemen zodra ze een op CMS-gebaseerde web applicatie in gebruik nemen. We definiëren zowel strategische, tactische als operationele processen en integreren deze in een bekend service level model.

Naast de ontwikkeling van WEM hebben we twee specifieke methodes onderzocht om te zien of WEM verbeterd kan worden: een Model Driven Web Engineering (MDWE) benadering en een Product Verticalisatie benadering. De MDWE methode is toegepast om het proces van ontwerpen naar ontwikkeling van websites deels te automatiseren. In de situatie van dit onderzoek gaan we uit van geconfigureerd CMS als doel-applicatie die de basis vormt voor een Web applicatie. We beschrijven een methode hoe we bepaalde functionaliteiten kunnen modelleren en deze modellen kunnen omzetten naar een configuratie voor het CMS. De Product Verticalisatie gaat uit van het principe dat organisaties met vergelijkbare eigenschappen (zoals markt, business model, of doelgroep) vergelijkbare wensen en eisen aan het product hebben. Het idee is dat als we een CMS al kunnen pre-configureren voor een specifieke markt, het zal resulteren in efficiëntere implementatie.

De toepassing van WEM is gevalideerd in verscheidene case studies, vragenlijsten en expert reviews. De Model Driven Web Engineering methode is bovendien gevalideerd in de vorm van een prototype.

De belangrijkste bijdragen van dit onderzoek zijn de volgende:

- We geven een gedetailleerd overzicht van WEM met het ontwikkelproces van op CMS-gebaseerde web applicaties. In het onderzoeksgebied van Web Engineering zijn reeds vele publicaties, maar een gedetailleerde beschrijving hoe omgegaan kan worden met een CMS als ontwikkelplatform voor web applicaties ontbreekt. De procesbeschrijvingen in dit proefschrift geven mede-onderzoekers inzicht in hoe wij bestaande methoden gecombineerd hebben tot een nieuwe methode voor de ontwikkeling van op CMS-gebaseerde web applicaties waarbij we gebruik hebben gemaakt van situationele method-engineering.

- We bieden een overzicht van belangrijkste concepten die gerelateerd zijn aan Content Management Systemen en nemen in de procesbeschrijving mee hoe daar mee omgegaan kan worden. Deze concepten worden gebruikt bij de vergelijking, selectie en analyse van methode fragmenten.

- We presenteren twee verbeteringen voor het ontwikkelproces gebaseerd op Model Driven Web Engineering en Product Verticalisatie. Dit heeft geresulteerd in een methode, een ontwerp en een prototype die in de praktijk getoetst is. Tot slot dragen we een verbetering aan voor het analyseren en identificeren van gelijkenissen in bestaande implementaties om het CMS voor te bereiden op product verticalisatie. Dit kan leiden tot pre-configuratie van het CMS en efficiëntere implementaties.

Deze drie bijdragen bieden handreikingen voor organisaties die betrokken zijn bij de ontwikkeling en implementatie van op CMS-gebaseerde web applicaties in de vorm van richtlijnen, best practices, methodes en artefacten. Tegelijkertijd kunnen ontwikkelaars van Content Management Systemen het CMS en de manier waarop het geïmplementeerd wordt verrijken met een Model Driven Web Engineering model en Product Verticalisatie. Organisaties die zelf een op CMS-gebaseerde web applicatie willen ontwikkelen, hebben een gedetailleerd overzicht van relevante processen met best practices, mogelijke valkuilen en verbetervoorstellen.

Resume

Jurriaan Souer (1980) obtained his Master's degree in Computer Science at Utrecht University. Soon afterwards he started working at GX Software, where he combined his research interests as a part-time PhD student with a career at a software company. This synergetic combination resulted in several publications that became the foundation of this dissertation. In 2011 Jurriaan moved with his family to the United States where he is currently active as a Director at GX Software, Redwood City (USA). His academic and industrial activities are focused on product software for developing Web-based applications and web engineering in general.

SIKS PhD Theses

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects
1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information
1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective
1998-4 Dennis Breuker (UM)
Memory versus Search in Games
1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting
1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products
1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets
1999-3 Don Beal (UM)
The Nature of Minimax Search
1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects
1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems
1999-6 Niek J.E. Wijngaards (VU)
Re-design of compositional systems
1999-7 David Spelt (UT)
Verification support for object database design
1999-8 Jacques H.J. Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation
2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance
2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management
2000-3 Carolien M.T. Metselaar (UVA)
Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief
2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design
2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval
2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication
2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management
2000-8 Veerle Coup (EUR)
Sensitivity Analyis of Decision-Theoretic Networks
2000-9 Florian Waas (CWI)

Principles of Probabilistic Query Optimization
2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations, Algorithms and Architecture
2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management
2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks
2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models
2001-3 Maarten van Someren (UvA)
Learning as problem solving
2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets
2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style
2001-6 Martijn van Welie (VU)
Task-based User Interface Design
2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization
2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics
2001-9 Pieter Jan 't Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes
2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design
2001-11 Tom M. van Engers (VUA)
Knowledge Management: The Role of Mental Models in Business Systems Design
2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis
2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections
2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval
2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining
2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents
2002-06 Laurens Mommers (UL)
Applied legal epistemology; Building a knowledge-based ontology of the legal domain
2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications
2002-08 Jaap Gordijn (VU)
Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas
2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems
2002-10 Brian Sheppard (UM)
Towards Perfect Play of Scrabble
2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and Organisational Applications
2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems
2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications
2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems
2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling
2002-16 Pieter van Langen (VU)

The Anatomy of Design: Foundations, Models and Applications
2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance
2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments
2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems
2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy
2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology
2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach
2003-06 Boris van Schooten (UT)
Development and specification of virtual environments
2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks
2003-08 Yongping Ran (UM)
Repair Based Scheduling
2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour
2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture
2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks
2003-12 Roeland Ordelman (UT)
Dutch speech recognition in multimedia information retrieval
2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models
2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations
2003-15 Mathijs de Weerdt (TUD)
Plan Merging in Multi-Agent Systems
2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses
2003-17 David Jansen (UT)
Extensions of Statecharts with Probability, Time, and Stochastic Timing
2003-18 Levente Kocsis (UM)
Learning Search Decisions
2004-01 Virginia Dignum (UU)
A Model for Organizational Interaction: Based on Agents, Founded in Logic
2004-02 Lai Xu (UvT)
Monitoring Multi-party Contracts for E-business
2004-03 Perry Groot (VU)
A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving
2004-04 Chris van Aart (UVA)
Organizational Principles for Multi-Agent Architectures
2004-05 Viara Popova (EUR)
Knowledge discovery and monotonicity
2004-06 Bart-Jan Hommes (TUD)
The Evaluation of Business Process Modeling Techniques
2004-07 Elise Boltjes (UM)
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes
2004-08 Joop Verbeek(UM)
Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale
politiële gegevensuitwisseling en digitale expertise
2004-09 Martin Caminada (VU)
For the Sake of the Argument; explorations into argument-based reasoning
2004-10 Suzanne Kabel (UVA)

Knowledge-rich indexing of learning-objects
2004-11 Michel Klein (VU)
Change Management for Distributed Ontologies
2004-12 The Duy Bui (UT)
Creating emotions and facial expressions for embodied agents
2004-13 Wojciech Jamroga (UT)
Using Multiple Models of Reality: On Agents who Know how to Play
2004-14 Paul Harrenstein (UU)
Logic in Conflict. Logical Explorations in Strategic Equilibrium
2004-15 Arno Knobbe (UU)
Multi-Relational Data Mining
2004-16 Federico Divina (VU)
Hybrid Genetic Relational Search for Inductive Learning
2004-17 Mark Winands (UM)
Informed Search in Complex Games
2004-18 Vania Bessa Machado (UvA)
Supporting the Construction of Qualitative Knowledge Models
2004-19 Thijs Westerveld (UT)
Using generative probabilistic models for multimedia retrieval
2004-20 Madelon Evers (Nyenrode)
Learning from Design: facilitating multidisciplinary design teams
2005-01 Floor Verdenius (UVA)
Methodological Aspects of Designing Induction-Based Applications
2005-02 Erik van der Werf (UM))
AI techniques for the game of Go
2005-03 Franc Grootjen (RUN)
A Pragmatic Approach to the Conceptualisation of Language
2005-04 Nirvana Meratnia (UT)
Towards Database Support for Moving Object data
2005-05 Gabriel Infante-Lopez (UVA)
Two-Level Probabilistic Grammars for Natural Language Parsing
2005-06 Pieter Spronck (UM)
Adaptive Game AI
2005-07 Flavius Frasincar (TUE)
Hypermedia Presentation Generation for Semantic Web Information Systems
2005-08 Richard Vdovjak (TUE)
A Model-driven Approach for Building Distributed Ontology-based Web Applications
2005-09 Jeen Broekstra (VU)
Storage, Querying and Inferencing for Semantic Web Languages
2005-10 Anders Bouwer (UVA)
Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments
2005-11 Elth Ogston (VU)
Agent Based Matchmaking and Clustering - A Decentralized Approach to Search
2005-12 Csaba Boer (EUR)
Distributed Simulation in Industry
2005-13 Fred Hamburg (UL)
Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen
2005-14 Borys Omelayenko (VU)
Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics
2005-15 Tibor Bosse (VU)
Analysis of the Dynamics of Cognitive Processes
2005-16 Joris Graaumans (UU)
Usability of XML Query Languages
2005-17 Boris Shishkov (TUD)
Software Specification Based on Re-usable Business Components
2005-18 Danielle Sent (UU)
Test-selection strategies for probabilistic networks
2005-19 Michel van Dartel (UM)
Situated Representation

2005-20 Cristina Coteanu (UL)
Cyber Consumer Law, State of the Art and Perspectives
2005-21 Wijnand Derks (UT)
Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics
2006-01 Samuil Angelov (TUE)
Foundations of B2B Electronic Contracting
2006-02 Cristina Chisalita (VU)
Contextual issues in the design and use of information technology in organizations
2006-03 Noor Christoph (UVA)
The role of metacognitive skills in learning to solve problems
2006-04 Marta Sabou (VU)
Building Web Service Ontologies
2006-05 Cees Pierik (UU)
Validation Techniques for Object-Oriented Proof Outlines
2006-06 Ziv Baida (VU)
Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling
2006-07 Marko Smiljanic (UT)
XML schema matching – balancing efficiency and effectiveness by means of clustering
2006-08 Eelco Herder (UT)
Forward, Back and Home Again - Analyzing User Behavior on the Web
2006-09 Mohamed Wahdan (UM)
Automatic Formulation of the Auditor's Opinion
2006-10 Ronny Siebes (VU)
Semantic Routing in Peer-to-Peer Systems
2006-11 Joeri van Ruth (UT)
Flattening Queries over Nested Data Types
2006-12 Bert Bongers (VU)
Interactivation - Towards an e-cology of people, our technological environment, and the arts
2006-13 Henk-Jan Lebbink (UU)
Dialogue and Decision Games for Information Exchanging Agents
2006-14 Johan Hoorn (VU)
Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change
2006-15 Rainer Malik (UU)
CONAN: Text Mining in the Biomedical Domain
2006-16 Carsten Riggelsen (UU)
Approximation Methods for Efficient Learning of Bayesian Networks
2006-17 Stacey Nagata (UU)
User Assistance for Multitasking with Interruptions on a Mobile Device
2006-18 Valentin Zhizhkun (UVA)
Graph transformation for Natural Language Processing
2006-19 Birna van Riemsdijk (UU)
Cognitive Agent Programming: A Semantic Approach
2006-20 Marina Velikova (UvT)
Monotone models for prediction in data mining
2006-21 Bas van Gils (RUN)
Aptness on the Web
2006-22 Paul de Vrieze (RUN)
Fundaments of Adaptive Personalisation
2006-23 Ion Juvina (UU)
Development of Cognitive Model for Navigating on the Web
2006-24 Laura Hollink (VU)
Semantic Annotation for Retrieval of Visual Resources
2006-25 Madalina Drugan (UU)
Conditional log-likelihood MDL and Evolutionary MCMC
2006-26 Vojkan Mihajlovic (UT)
Score Region Algebra: A Flexible Framework for Structured Information Retrieval
2006-27 Stefano Bocconi (CWI)
Vox Populi: generating video documentaries from semantically annotated media repositories
2006-28 Borkur Sigurbjornsson (UVA)

Focused Information Access using XML Element Retrieval
2007-01 Kees Leune (UvT)
Access Control and Service-Oriented Architectures
2007-02 Wouter Teepe (RUG)
Reconciling Information Exchange and Confidentiality: A Formal Approach
2007-03 Peter Mika (VU)
Social Networks and the Semantic Web
2007-04 Jurriaan van Diggelen (UU)
Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach
2007-05 Bart Schermer (UL)
Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance
2007-06 Gilad Mishne (UVA)
Applied Text Analytics for Blogs
2007-07 Natasa Jovanovic' (UT)
To Whom It May Concern - Addressee Identification in Face-to-Face Meetings
2007-08 Mark Hoogendoorn (VU)
Modeling of Change in Multi-Agent Organizations
2007-09 David Mobach (VU)
Agent-Based Mediated Service Negotiation
2007-10 Huib Aldewereld (UU)
Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols
2007-11 Natalia Stash (TUE)
Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System
2007-12 Marcel van Gerven (RUN)
Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty
2007-13 Rutger Rienks (UT)
Meetings in Smart Environments; Implications of Progressing Technology
2007-14 Niek Bergboer (UM)
Context-Based Image Analysis
2007-15 Joyca Lacroix (UM)
NIM: a Situated Computational Memory Model
2007-16 Davide Grossi (UU)
Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems
2007-17 Theodore Charitos (UU)
Reasoning with Dynamic Networks in Practice
2007-18 Bart Orriens (UvT)
On the development an management of adaptive business collaborations
2007-19 David Levy (UM)
Intimate relationships with artificial partners
2007-20 Slinger Jansen (UU)
Customer Configuration Updating in a Software Supply Network
2007-21 Karianne Vermaas (UU)
Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005
2007-22 Zlatko Zlatev (UT)
Goal-oriented design of value and process models from patterns
2007-23 Peter Barna (TUE)
Specification of Application Logic in Web Information Systems
2007-24 Georgina Ramrez Camps (CWI)
Structural Features in XML Retrieval
2007-25 Joost Schalken (VU)
Empirical Investigations in Software Process Improvement
2008-01 Katalin Boer-Sorbn (EUR)
Agent-Based Simulation of Financial Markets: A modular,continuous-time approach
2008-02 Alexei Sharpanskykh (VU)
On Computer-Aided Methods for Modeling and Analysis of Organizations
2008-03 Vera Hollink (UVA)
Optimizing hierarchical menus: a usage-based approach
2008-04 Ander de Keijzer (UT)

Management of Uncertain Data - towards unattended integration
2008-05 Bela Mutschler (UT)
Modeling and simulating causal dependencies on process-aware information systems from a cost perspective
2008-06 Arjen Hommersom (RUN)
On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective
2008-07 Peter van Rosmalen (OU)
Supporting the tutor in the design and support of adaptive e-learning
2008-08 Janneke Bolt (UU)
Bayesian Networks: Aspects of Approximate Inference
2008-09 Christof van Nimwegen (UU)
The paradox of the guided user: assistance can be counter-effective
2008-10 Wauter Bosma (UT)
Discourse oriented summarization
2008-11 Vera Kartseva (VU)
Designing Controls for Network Organizations: A Value-Based Approach
2008-12 Jozsef Farkas (RUN)
A Semiotically Oriented Cognitive Model of Knowledge Representation
2008-13 Caterina Carraciolo (UVA)
Topic Driven Access to Scientific Handbooks
2008-14 Arthur van Bunningen (UT)
Context-Aware Querying; Better Answers with Less Effort
2008-15 Martijn van Otterlo (UT)
The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains
2008-16 Henriette van Vugt (VU)
Embodied agents from a user's perspective
2008-17 Martin Op 't Land (TUD)
Applying Architecture and Ontology to the Splitting and Allying of Enterprises
2008-18 Guido de Croon (UM)
Adaptive Active Vision
2008-19 Henning Rode (UT)
From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search
2008-20 Rex Arendsen (UVA)
Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven
2008-21 Krisztian Balog (UVA)
People Search in the Enterprise
2008-22 Henk Koning (UU)
Communication of IT-Architecture
2008-23 Stefan Visscher (UU)
Bayesian network models for the management of ventilator-associated pneumonia
2008-24 Zharko Aleksovski (VU)
Using background knowledge in ontology matching
2008-25 Geert Jonker (UU)
Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency
2008-26 Marijn Huijbregts (UT)
Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled
2008-27 Hubert Vogten (OU)
Design and Implementation Strategies for IMS Learning Design
2008-28 Ildiko Flesch (RUN)
On the Use of Independence Relations in Bayesian Networks
2008-29 Dennis Reidsma (UT)
Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans
2008-30 Wouter van Atteveldt (VU)
Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content
2008-31 Loes Braun (UM)
Pro-Active Medical Information Retrieval
2008-32 Trung H. Bui (UT)
Toward Affective Dialogue Management using Partially Observable Markov Decision Processes

2008-33 Frank Terpstra (UVA)
Scientific Workflow Design; theoretical and practical issues
2008-34 Jeroen de Knijf (UU)
Studies in Frequent Tree Mining
2008-35 Ben Torben Nielsen (UvT)
Dendritic morphologies: function shapes structure
2009-01 Rasa Jurgelenaite (RUN)
Symmetric Causal Independence Models
2009-02 Willem Robert van Hage (VU)
Evaluating Ontology-Alignment Techniques
2009-03 Hans Stol (UvT)
A Framework for Evidence-based Policy Making Using IT
2009-04 Josephine Nabukenya (RUN)
Improving the Quality of Organisational Policy Making using Collaboration Engineering
2009-05 Sietse Overbeek (RUN)
Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality
2009-06 Muhammad Subianto (UU)
Understanding Classification
2009-07Ronald Poppe (UT)
Discriminative Vision-Based Recovery and Recognition of Human Motion
2009-08 Volker Nannen (VU)
Evolutionary Agent-Based Policy Analysis in Dynamic Environments
2009-09 Benjamin Kanagwa (RUN)
Design, Discovery and Construction of Service-oriented Systems
2009-10 Jan Wielemaker (UVA)
Logic programming for knowledge-intensive interactive applications
2009-11 Alexander Boer (UVA)
Legal Theory, Sources of Law & the Semantic Web
2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin)
perating Guidelines for Services
2009-13 Steven de Jong (UM)
Fairness in Multi-Agent Systems
2009-14 Maksym Korotkiy (VU)
From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)
2009-15 Rinke Hoekstra (UVA)
Ontology Representation - Design Patterns and Ontologies that Make Sense
2009-16 Fritz Reul (UvT)
New Architectures in Computer Chess
2009-17 Laurens van der Maaten (UvT)
Feature Extraction from Visual Data
2009-18 Fabian Groffen (CWI)
Armada, An Evolving Database System
2009-19 Valentin Robu (CWI)
Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets
2009-20 Bob van der Vecht (UU)
Adjustable Autonomy: Controling Influences on Decision Making
2009-21 Stijn Vanderlooy (UM)
Ranking and Reliable Classification
2009-22 Pavel Serdyukov (UT)
Search For Expertise: Going beyond direct evidence
2009-23 Peter Hofgesang (VU)
Modelling Web Usage in a Changing Environment
2009-24 Annerieke Heuvelink (VUA)
Cognitive Models for Training Simulations
2009-25 Alex van Ballegooij (CWI)
RAM: Array Database Management through Relational Mapping
2009-26 Fernando Koch (UU)
An Agent-Based Model for the Development of Intelligent Mobile Services

2009-27 Christian Glahn (OU)
Contextual Support of social Engagement and Reflection on the Web
2009-28 Sander Evers (UT)
Sensor Data Management with Probabilistic Models
2009-29 Stanislav Pokraev (UT)
Model-Driven Semantic Integration of Service-Oriented Applications
2009-30 Marcin Zukowski (CWI)
Balancing vectorized query execution with bandwidth-optimized storage
2009-31 Sofiya Katrenko (UVA)
A Closer Look at Learning Relations from Text
2009-32 Rik Farenhorst (VU) and Remco de Boer (VU)
Architectural Knowledge Management: Supporting Architects and Auditors
2009-33 Khiet Truong (UT)
How Does Real Affect Affect Affect Recognition In Speech?
2009-34 Inge van de Weerd (UU)
Advancing in Software Product Management: An Incremental Method Engineering Approach
2009-35Wouter Koelewijn (UL)
Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling
2009-36 Marco Kalz (OUN)
Placement Support for Learners in Learning Networks
2009-37 Hendrik Drachsler (OUN)
Navigation Support for Learners in Informal Learning Networks
2009-38 Riina Vuorikari (OU)
Tags and self-organisation: a metadata ecology for learning resources in a multilingual context
2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin)
Service Substitution – A Behavioral Approach Based on Petri Nets
2009-40 Stephan Raaijmakers (UvT)
Multinomial Language Learning: Investigations into the Geometry of Language
2009-41 Igor Berezhnyy (UvT)
Digital Analysis of Paintings
2009-42 Toine Bogers
Recommender Systems for Social Bookmarking
2009-43 Virginia Nunes Leal Franqueira (UT)
Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients
2009-44 Roberto Santana Tapia (UT)
Assessing Business-IT Alignment in Networked Organizations
2009-45 Jilles Vreeken (UU)
Making Pattern Mining Useful
2009-46 Loredana Afanasiev (UvA)
Querying XML: Benchmarks and Recursion
2010-01 Matthijs van Leeuwen (UU)
Patterns that Matter
2010-02 Ingo Wassink (UT)
Work flows in Life Science
2010-03 Joost Geurts (CWI)
A Document Engineering Model and Processing Framework for Multimedia documents
2010-04 Olga Kulyk (UT)
Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments
2010-05 Claudia Hauff (UT)
Predicting the Effectiveness of Queries and Retrieval Systems
2010-06 Sander Bakkes (UvT)
Rapid Adaptation of Video Game AI
2010-07 Wim Fikkert (UT)
Gesture interaction at a Distance
2010-08 Krzysztof Siewicz (UL)
Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments
2010-09 Hugo Kielman (UL)
A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging

2010-10 Rebecca Ong (UL)
Mobile Communication and Protection of Children
2010-11 Adriaan Ter Mors (TUD)
The world according to MARP: Multi-Agent Route Planning
2010-12 Susan van den Braak (UU)
Sensemaking software for crime analysis
2010-13 Gianluigi Folino (RUN)
High Performance Data Mining using Bio-inspired techniques
2010-14 Sander van Splunter (VU)
Automated Web Service Reconfiguration
2010-15 Lianne Bodenstaff (UT)
Managing Dependency Relations in Inter-Organizational Models
2010-16 Sicco Verwer (TUD)
Efficient Identification of Timed Automata, theory and practice
2010-17 Spyros Kotoulas (VU)
Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications
2010-18 Charlotte Gerritsen (VU)
Caught in the Act: Investigating Crime by Agent-Based Simulation
2010-19 Henriette Cramer (UvA)
People's Responses to Autonomous and Adaptive Systems
2010-20 Ivo Swartjes (UT)
Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative
2010-21 Harold van Heerde (UT)
Privacy-aware data management by means of data degradation
2010-22 Michiel Hildebrand (CWI)
End-user Support for Access to Heterogeneous Linked Data
2010-23 Bas Steunebrink (UU)
The Logical Structure of Emotions
2010-24 Dmytro Tykhonov
Designing Generic and Efficient Negotiation Strategies
2010-25 Zulfiqar Ali Memon (VU)
Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective
2010-26 Ying Zhang (CWI)
XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines 2010-27 Marten Voulon (UL)
Automatisch contracteren
2010-28 Arne Koopman (UU)
Characteristic Relational Patterns
2010-29 Stratos Idreos(CWI)
Database Cracking: Towards Auto-tuning Database Kernels
2010-30 Marieke van Erp (UvT)
Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval
2010-31 Victor de Boer (UVA)
Ontology Enrichment from Heterogeneous Sources on the Web
2010-32 Marcel Hiel (UvT)
An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems
2010-33 Robin Aly (UT)
Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval
2010-34 Teduh Dirgahayu (UT)
Interaction Design in Service Compositions
2010-35 Dolf Trieschnigg (UT)
Proof of Concept: Concept-based Biomedical Information Retrieval
2010-36 Jose Janssen (OU)
Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification
2010-37 Niels Lohmann (TUE)
Correctness of services and their composition
2010-38 Dirk Fahland (TUE)
From Scenarios to components
2010-39 Ghazanfar Farooq Siddiqui (VU)
Integrative modeling of emotions in virtual agents

2010-40 Mark van Assem (VU)
Converting and Integrating Vocabularies for the Semantic Web
2010-41 Guillaume Chaslot (UM)
Monte-Carlo Tree Search
2010-42 Sybren de Kinderen (VU)
Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach
2010-43 Peter van Kranenburg (UU)
A Computational Approach to Content-Based Retrieval of Folk Song Melodies
2010-44 Pieter Bellekens (TUE)
An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain
2010-45 Vasilios Andrikopoulos (UvT)
A theory and model for the evolution of software services
2010-46 Vincent Pijpers (VU)
e3alignment: Exploring Inter-Organizational Business-ICT Alignment
2010-47 Chen Li (UT)
Mining Process Model Variants: Challenges, Techniques, Examples
2010-48 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets
2010-49 Jahn-Takeshi Saito (UM)
Solving difficult game positions
2010-50 Bouke Huurnink (UVA)
Search in Audiovisual Broadcast Archives 2010-51 Alia Khairia Amin (CWI)
Understanding and supporting information seeking tasks in multiple sources
2010-52 Peter-Paul van Maanen (VU)
Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention
2010-53 Edgar Meij (UVA)
Combining Concepts and Language Models for Information Access
2011-01 Botond Cseke (RUN)
Variational Algorithms for Bayesian Inference in Latent Gaussian Models
2011-02 Nick Tinnemeier(UU)
Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language
2011-03 Jan Martijn van der Werf (TUE)
Compositional Design and Verification of Component-Based Information Systems
2011-04 Hado van Hasselt (UU)
Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms
2011-05 Base van der Raadt (VU)
Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline
2011-06 Yiwen Wang (TUE)
Semantically-Enhanced Recommendations in Cultural Heritage
2011-07 Yujia Cao (UT)
Multimodal Information Presentation for High Load Human Computer Interaction
2011-08 Nieske Vergunst (UU)
BDI-based Generation of Robust Task-Oriented Dialogues
2011-09 Tim de Jong (OU)
Contextualised Mobile Media for Learning
2011-10 Bart Bogaert (UvT)
Cloud Content Contention
2011-11 Dhaval Vyas (UT)
Designing for Awareness: An Experience-focused HCI Perspective
2011-12 Carmen Bratosin (TUE)
Grid Architecture for Distributed Process Mining
2011-13 Xiaoyu Mao (UvT)
Airport under Control. Multiagent Scheduling for Airport Ground Handling
2011-14 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets
2011-15 Marijn Koolen (UvA)
The Meaning of Structure: the Value of Link Evidence for Information Retrieval
2011-16 Maarten Schadd (UM)

Selective Search in Games of Different Complexity
2011-17 Jiyin He (UVA)
Exploring Topic Structure: Coherence, Diversity and Relatedness
2011-18 Mark Ponsen (UM)
Strategic Decision-Making in complex games
2011-19 Ellen Rusman (OU)
The Mind's Eye on Personal Profiles
2011-20 Qing Gu (VU)
Guiding service-oriented software engineering - A view-based approach
2011-21 Linda Terlouw (TUD)
Modularization and Specification of Service-Oriented Systems
2011-22 Junte Zhang (UVA)
System Evaluation of Archival Description and Access
2011-23 Wouter Weerkamp (UVA)
Finding People and their Utterances in Social Media
2011-24 Herwin van Welbergen (UT)
Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior
2011-25 Syed Waqar ul Qounain Jaffry (VU)
Analysis and Validation of Models for Trust Dynamics
2011-26 Matthijs Aart Pontier (VU)
Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots
2011-27 Aniel Bhulai (VU)
Dynamic website optimization through autonomous management of design patterns
2011-28 Rianne Kaptein(UVA)
Effective Focused Retrieval by Exploiting Query Context and Document Structure
2011-29 Faisal Kamiran (TUE)
Discrimination-aware Classification
2011-30 Egon van den Broek (UT)
Affective Signal Processing (ASP): Unraveling the mystery of emotions
2011-31 Ludo Waltman (EUR)
Computational and Game-Theoretic Approaches for Modeling Bounded Rationality
2011-32 Nees-Jan van Eck (EUR)
Methodological Advances in Bibliometric Mapping of Science
2011-33 Tom van der Weide (UU)
Arguing to Motivate Decisions
2011-34 Paolo Turrini (UU)
Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations
2011-35 Maaike Harbers (UU)
Explaining Agent Behavior in Virtual Training
2011-36 Erik van der Spek (UU)
Experiments in serious game design: a cognitive approach
2011-37 Adriana Burlutiu (RUN)
Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference
2011-38 Nyree Lemmens (UM)
Bee-inspired Distributed Optimization
2011-39 Joost Westra (UU)
Organizing Adaptation using Agents in Serious Games
2011-40 Viktor Clerc (VU)
Architectural Knowledge Management in Global Software Development
2011-41 Luan Ibraimi (UT)
Cryptographically Enforced Distributed Data Access Control
2011-42 Michal Sindlar (UU)
Explaining Behavior through Mental State Attribution
2011-43 Henk van der Schuur (UU)
Process Improvement through Software Operation Knowledge
2011-44 Boris Reuderink (UT)
Robust Brain-Computer Interfaces

2011-45 Herman Stehouwer (UvT)
Statistical Language Models for Alternative Sequence Selection
2011-46 Beibei Hu (TUD)
Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work
2011-47 Azizi Bin Ab Aziz(VU)
Exploring Computational Models for Intelligent Support of Persons with Depression
2012-01 Terry Kakeeto (UvT)
Relationship Marketing for SMEs in Uganda
2012-02 Muhammad Umair (VU)
Adaptivity, emotion, and Rationality in Human and Ambient Agent Models
2012-03 Adam Vanya (VU)
Supporting Architecture Evolution by Mining Software Repositories