

# Modelleren van CMS based WebApplications

---

Master Thesis, Plan van Aanpak

Versie: 1.0

20 maart 2008

M.J.B. (Thijs) Kupers

Radboud Universiteit Nijmegen

Master Computer Science



## Inleiding

Dit document beschrijft mijn plan van aanpak in het kader van mijn afstudeeronderzoek binnen de master-opleiding informatica aan de Radboud Universiteit Nijmegen. Mijn onderzoek vindt plaats binnen <GX> in Nijmegen.

GX is een web-development bedrijf welke zowel een standaard product als maatwerk biedt. Het standaard product omhelst een Webbased Content Management System, GX WebManager, waarmee op een eenvoudige manier webapplicaties beheerd kunnen worden. Daarnaast verzorgen ze maatwerk om domein specifieke oplossingen te kunnen bieden.

## Probleemstelling

In het verleden is er veel onderzoek gedaan naar ontwerp methodieken voor webapplicaties. Veel van deze methodes zijn afgeleid van al bestaande methodes en technieken. Echter, er is nog weinig onderzocht op het gebied van webengineering waarbij er gebruikt wordt gemaakt van een al bestaand Web-based Content Management System (WCMS) – Het meeste onderzoek gaat uit van web engineering van scratch (Weerd, et al. 2006).

GX WebEngineering Method (WEM) geeft een specificatie van het engineeringtraject van een CMS gebaseerde webapplicatie (Souer, et al. 2007). Hierbij wordt onderscheid gemaakt tussen een standaard traject en een complex traject. Standaard projecten zijn hoofdzakelijk gebaseerd op standaard functionaliteiten die in het CMS aanwezig zijn. Complexe projecten bevatten zowel nieuwe als een hoop aanpassingen op de standaard functionaliteiten.

Het projectmatige gedeelte van WEM is grotendeels beschreven. Uit case studies, blijkt dat de vertaalslag van functionaliteit naar standaard componenten en hun configuratie nog niet goed is uitgewerkt (Weerd, et al. 2006), (Souer, et al. 2007). De stap tussen requirements en Use Cases enerzijds en de geconfigureerde componenten anderzijds is nog te groot. Dit geldt zowel voor de ontwikkelaars als voor de klant.

## Onderzoeksvraag

Na aanleiding van bovengeschetste probleemstelling, zou de volgende onderzoeksvraag gedefinieerd kunnen worden:

*“Hoe zou een modelleertaal eruit moeten zien waarmee je op basis van requirements en use cases modules binnen een CMS kan specificeren?”*

Door een taal te creëren waarin op een abstract niveau de componenten en met name hun werking (inputs, validatie, routing, afhandelingen, etc.) kan worden gespecificeerd, wordt dit gat gedicht. Deze moet wel aansluiten op de requirements en Use Cases zodat de benodigde informatie hieruit afgeleid kan worden. Doordat de specificatie van de modelleertaal formeel vast ligt kunnen aan de hand van gemaakte modellen gedeelten van de WCMS gegenereerd worden (code generation/ automatische componenten selectie en configuratie/ transformatie).

De onderzoeksvraag kan als volgt worden opgebroken in deelvragen:

- Welke eigenschappen moet de modelleertaal hebben?
  - Wie zijn de gebruikers?
    - Wat zijn hun doelen?
  - Wat (welke informatie en functionaliteiten van een component) moet er gemodelleerd kunnen worden?
    - Op welke manier is deze beschikbaar / afleidbaar uit requirements en use cases?
- Vindt deze stap (van requirements en use cases naar geconfigureerde componenten) ook plaats in de ontwikkeling van niet-cms-applicaties?
  - Geeft deze stap problemen?
  - Hoe wordt het daar opgelost?

Literatuurstudie:

- Hoe is een modelleertaal gedefinieerd?
- bekijken van bestaande modelleertalen en hun eigenschappen en deze vervolgens vergelijken met de gewenste modelleertaal
  - Welke bestaande modelleertalen doen iets gelijksoortigs?
  - Welke aspecten zijn bruikbaar?
- Hoe kan er voor worden gezorgd dat een modelleertaal flexibel en uitbreidbaar is? (bv voor nieuwe componenten)

## Aanverwante Literatuur

Een literatuurstudie naar modelleertechnieken in aanverwante disciplines, geeft een overzicht van bestaande methodes. Hieruit kunnen bepaalde aspecten worden afgeleid, die bruikbaar zijn in de te ontwikkelen methode. Zo zijn er verschillende web-modellerings technieken. Hoewel deze uitgaan van het bouwen van een webapplicatie van scratch, kunnen hier toch bruikbare concepten in zitten.

Object Oriented Hypermedia Design Model (**OOHDM**) is een methode waarin een webapplicatie wordt ontworpen met verschillende modellen (Schwabe and Rossi 1998). Dit werd later uitgebreid met een definitie fase voor requirements door middel van User Interaction Diagrams (**UIDs**) (Vilain, Schwabe and Sieckenius de Souza 2000). De Web Modeling Language (**WebML**) is een notatie voor het specificeren van websites op conceptueel niveau. Het bestaat uit vier verschillende modellen (Ceri, Fraternali and Bongio 2000). UML-based Web Engineering (**UWE**) beschrijft een op UML gebaseerde methode voor de ontwikkeling van web applicaties (Koch and Kraus 2002). Een conceptueel modelleer principe gebaseerd op OO wordt beschreven door **OOWS** (Object Oriented Web Solution) (Pastor, Fons and Pelechano 2003). De architectuur van webapplicaties kan worden gespecificeerd met Web Software Architectuur (**WebSA**) (Meliá, Gómez and Koch 2005).

Daarnaast kan er gekeken worden naar Enterprise Resource Planning (ERP)-systemen. Met een ERP-systeem wordt getracht om alle processen en gegevens binnen een organisatie in één systeem onder te brengen. ERP software is uiterst configureerbaar om aan de verschillende behoeften van organisaties te voldoen. Hierbij worden verschillende componenten op maat gesneden voor de specifieke requirements van een organisatie (Klaus, Rosemann and Gable 2000). Zowel bij de implementatie van een CMS als bij een ERP-systeem, moeten er dus bestaande componenten geconfigureerd worden. ERP-systemen, en het daarbij horende implementatie traject, zijn echter veel uitgebreider dan een CMS systeem; het uitgebreide ontwikkeltraject is niet nodig, maar er kunnen wel interessante concepten in zitten.

Het Enterprise Object Model (**EOM**), beschrijft een methode voor het verzamelen van requirements en deze te vertalen in configuratie settings voor ERP software (Arinze and Anandarajan 2003). De Enterprise Process Modeling Language (**EPML**) is een proces modelleer taal die gebaseerd is op bestaande procesmodelleertechnieken. Het grafische model wordt via XML geformaliseerd door middel van petri-nets (Dalal, et al. 2004). Dynamic Enterprise Modeling (**DEM**) werd geïntroduceerd door Baan. Met vier verschillende modellen kunnen gebruikers de organisatie en de proces-flow modeleren. Daarnaast wordt door middel van toolsupport de ontwikkeltijd verkort (Es and Post 1996).

## Planning

De planning geeft een tijdschema weer waarin ik probeer antwoord te geven op de verschillende deelvragen.

	Weeknr:	12	13	14	15	16	17
		17-mrt-08	24-mrt-08	31-mrt-08	7-apr-08	14-apr-08	21-apr-08
Uitzoeken wat er gemodelleerd moet worden Abstraheren tot concepten en domeinmodel							
Uitzoeken welke info beschikbaar is in requirements/use cases Gebruikersanalyse							
Bestaande modelleertalen mappen met concepten							
Uitzoeken hoe een modelleertaal gedefinieerd is Uitzoeken hoe een modelleertaal uitbreidbaar te maken is							
Scriptie schrijven							

	Weeknr:	18 <sup>1</sup>	19	20	21	22	23	24	25	26	27
		28-apr-08	5-mei-08	12-mei-08	19-mei-08	26-mei-08	2-jun-08	9-jun-08	16-jun-08	23-jun-08	30-jun-08
Definitie wat wordt er gemodelleerd Definitie modelleertaal (grafisch + formeel)											
Prototypetool maken Testen met gebruikers											
Scriptie schrijven											

Door het proces op te delen in korte iteraties, blijft het geheel overzichtelijk en is de voortgang goed te bewaken. Bij eventuele vertraging kan de planning makkelijk worden bijgesteld. Hieronder volgt een verklaring van de verschillende onderdelen.

### *Uitzoeken wat er gemodelleerd moet worden*

Hierbij wordt precies uitgezocht welke dingen er binnen het CMS van toepassing zijn binnen het onderzoek. Zo heb je bij formulieren te maken met invoervelden, flow en stappen, handlers, routers, validators, personalisatie, rapportages, etc.

**Deliverable:** Lijst en beschrijvingen van alle zaken die van belang zijn.

<sup>1</sup> Week 18 telt vanwege Koninginnedag en Hemelvaart maar twee werkdagen

### *Abstraheren tot concepten en domeinmodel*

Hierbij wordt een domeinmodel samengesteld waarin de verbanden tussen alle zaken duidelijk wordt. Doordat zaken worden geabstraheerd tot concepten, valt dit beter te vergelijken met

**Deliverable:** Domeinmodel en beschrijving objecten.

### *Uitzoeken welke informatie beschikbaar is in requirements/ use cases*

De requirements en use cases vormen de input voor de modelleertool. Omdat deze hierop aan moet sluiten is het belangrijk te weten tot op welk niveau er informatie beschikbaar is.

**Deliverable:** Lijst van beschikbare informatie.

### *Gebruikersanalyse*

Het zijn uiteindelijk de gebruikers die met de tool moeten werken. Het is belangrijk dat de tool aansluit bij de belevingswereld en het mental model van de gebruiker. Deze worden beschreven aan de hand van de Goal-Directed Design (GDD) methode (Cooper, Reimann en Cronin 2007).

**Deliverable:** Een lijst van personas, goals en scenario's

### *Bestaande modelleertalen mappen met concepten*

Hierbij worden verschillende eigenschappen van bestaande modelleertalen vergeleken met de benodigde eigenschappen.

**Deliverable:** Matrix welke weergeeft in hoeverre de gewenste eigenschappen met bestaande modelleertalen overeenkomen.

### *Uitzoeken hoe een modelleertaal gedefinieerd is*

Wat is een modelleertaal eigenlijk? Hoe wordt deze geformaliseerd? Hoe past deze definitie in XML, zodat er communicatie met een CMS mogelijk is.

**Deliverable:** Beschrijving hoe een modelleertaal is opgebouwd.

### *Uitzoeken hoe een modelleertaal uitbreidbaar te maken is*

Het is belangrijk dat de te vormen modelleertaal uitbreidbaar is. Hierbij wordt uitgezocht hoe je daar rekening mee kan houden.

**Deliverable:** Beschrijving hoe een modelleertaal uitbreidbaar is.

### *Definitie wat wordt er gemodelleerd*

Hier wordt bepaald wat er uiteindelijk gemodelleerd gaat worden. Welke concepten? Hoe verhouden de concepten zich tot elkaar? Op welk abstractieniveau (ook afhankelijk van de gebruikersanalyse)?

**Deliverable:** Lijst van te modelleren concepten.

### *Definitie modelleertaal (grafisch + formeel)*

Hierbij wordt de eigenlijke modelleertaal gedefinieerd. Zowel de grafische opbouw, als de formele definitie, als de mapping naar XML.

**Deliverable:** Definitie van de modelleertaal.

### *Prototypetool maken*

Hierbij wordt een prototype-implementatie van de modelleertaal gemaakt in een metamodelleertool (Xiao, et al. 2007). Ook de mapping met het CMS wordt gemaakt.

**Deliverable:** Een prototype modelleertool voor de gemaakte modelleertaal.

### *Testen met gebruikers*

Hierbij wordt getest of de modelleertaal aansluit bij de behoeften van de gebruikers. Eventueel kunnen er nog dingen worden aangepast, of een lijst van aanbevelingen worden gemaakt. De tool wordt getest aan de hand van een aantal cases die overeenkomen met de scenario's van de gebruikers.

**Deliverable:** Testplan en beschrijving van de testresultaten.

### *Scriptie schrijven*

Gedurende het hele traject worden vorderingen, ideeën en conclusies bijgehouden in een werkdocument wat uiteindelijk de scriptie moet vormen.

**Deliverable:** Afstudeerscriptie.

## Literatuurlijst

- Arinze, B., and M. Anandarajan. "A framework for using OO mapping methods to rapidly configure ERP systems." *Communications of the ACM* 46, no. 2 (2003): 61-65.
- Ceri, S., P. Fraternali, and A. Bongio. "Web Modeling Language (WebML): a modeling language for designing Web sites." *Proceedings of the 9th international World Wide Web Conference on Computer Networks : the international Journal of Computer and Telecommunications Networking*, 2000: 137-157.
- Cooper, A., R. Reimann, and D. Cronin. *About Face 3: The Essentials of Interaction Design*. New York, NY, USA: Wiley publishing inc., 2007.
- Dalal, N. P., M. Kamath, W. J. Kolarik, and E. Sivaraman. "Toward an integrated framework for modeling enterprise processes." *Communications of the ACM* 47, no. 3 (2004): 83-87.
- Es, R.M. van, and H.A. Post. *Dynamic enterprise modeling : a paradigm shift in software implementation*. Dordrecht: Kluwer, 1996.
- Klaus, H., M. Rosemann, and G.G. Gable. "What is ERP?" *Information Systems Frontiers* 2, no. 2 (2000): 141-162.
- Koch, N., and A. Kraus. "The expressive Power of UML-based Web Engineering." *Proc. of IWWOST02, CYTED*, 2002: 105-119.
- Meliá, S., J. Gómez, and N. Koch. "Improving Web Design Methods with Architecture Modeling." *Lecture Notes in Computer Science* 3590 (2005): 53-64.
- Pastor, O., J. Fons, and V. Pelechano. "OOWS: A Method to Develop Web Applications from Web-Oriented Conceptual Models." *International Workshop on Web Oriented Software Technology (IWWOST)*, 2003: 65-70.
- Schwabe, D., and G. Rossi. *Developing hypermedia applications using OOHDM*. Paper, Pittsburgh, USA: Workshop on Hypermedia Development, 1998.
- Souer, J., I. van de Weerd, J. Versendaal, and J. Brinkkemper. "Situational requirements engineering for the development of Content Management System-based web applications." *International Journal of Web Engineering and Technology* 3, no. 4 (2007): 420-440.
- Vilain, P., D. Schwabe, and C. Sieckenius de Souza. "A Diagrammatic Tool for Representing User Interaction in UML." *Lecture Notes in Computer Science* 1939 (2000): 133-147.
- Weerd, I. van de, S. Brinkkemper, J. Souer, and J. Versendaal. "A Situational Implementation Method for Web-based Content Management System-applications: Method Engineering and Validation in Practice." *Software Process Improvement and Practice* 11 (2006): 521-538.

Xiao, H., M. Zhiyi, S. Weizhong, and L. Ge. "A metamodel for the notation of graphical modeling languages." *compsac 1* (2007): 219-224.

## Meta-informatie

### *Studentgegevens*

Thijs Kupers (0521450)  
mjbkopers@student.ru.nl

### *Contactgegevens GX*

GX creative online development  
Wijchenseweg 111  
6538 SW Nijmegen  
024-38 88 261  
info@gx.nl

### *Supervisors*

Jurriaan Souer (GX) – jurriaans@gx.nl  
Patrick van Bommel (Radboud Universiteit) – p.vanbommel@cs.ru.nl