Designing Web Content Management Systems Using the Method Association Approach

Lutzen Luinenburg[1], Slinger Jansen[2], Jurriaan Souer[1], Inge van de Weerd[2] and Sjaak Brinkkemper[2]

[1]GX, Wijchenseweg 111, 6538 SW Nijmegen, The Netherlands
{lutzen.luinenburg, jurriaan.souer}@gxwebmanager.com

[2]Department of Information and Computing Sciences,
Utrecht University, The Netherlands
{s.jansen, s.brinkkemper, i.vandeweerd}@cs.uu.nl

Abstract. Model-driven web approaches focus on creating robust web applications. There are two downsides to using these model-driven web approaches: they consist of a unique set of models and method and they are aimed at designing web applications from scratch. In this paper we present the Method Association Approach, which selects and constructs fitting methods from five model-driven web modeling approaches. The Method Association Approach creates modeling methods in specific web application domains for applications in different stages of their life cycle. The Method Association Approach positions existing model-driven approaches by means of meta-model concepts against key features of a specific web application domain. With the Method Association Approach a design method is constructed that seamlessly aligns with web application features, and makes use of proven web design concepts. The approach has been validated by means of an expert validation and two case studies.

1 Introduction

Since the beginning of web engineering [6], many web modeling methods have proposed in order to provide efficient and structured ways of developing web applications. However, these web modeling methods only provide design primitives for building web applications from scratch, and not including the possibility to design preexisting components related to a particular web domain. Web Content Management (WCM) systems are software products used for the management of and control of web content WCM systems have gradually merged with web applications resulting in CMS-based web applications [26]. The implementation of WCMSs is a complex task, since one has to cope with fast-changing requirements which have a high impact on architecture and design. Key in this process is the translation of business requirements into preexisting components. In order to make this process more transparent and effective, domain-specific web design methods are needed.

Current web modeling methods provide their own specific set of models including a various number of design primitives. These models address design primitives which have proven their usefulness in many cases. Therefore, it seems obvious to select existing web modeling methods for the creation of a situational web design method with regard to the WCM domain. Due to the high amount of web modeling methods, each with their set of models, it is complex to make a decision on which web modeling method to choose for applying this to the WCM domain.

In this research, we present a Method Association Approach wherein domain-specific feature groups are associated with web modeling methods in a uniform way. The Method Association Approach shows which feature groups are addressed by the selected web modeling methods, and so suitable for a particular application domain. The uniformity of the models has been established by meta-modeling. Another advantage of meta-modeling is that it is cost-effective and efficient in comparison to more empirical approaches (e.g. case studies).

The remainder of this paper is organized as follows. First, we present some related work. In section two we describe the Method Association Approach. Section three provides evaluation criteria used for method association. In section four, we show how the association table provides insight in the coverage of feature groups by several web modeling methods. Section five summarizes the validation of the situational web design method by means of an expert validation and two case studies. We end this paper with a brief discussion and conclusions.

In web engineering, many web modeling methods have been proposed in the course of time. Many comparative studies have been performed in order to provide an overview of these web modeling methods. Each of the comparative studies has its own goal and provides an overview by focusing on different modeling requirements, like: general method requirements [25], navigation modeling [8], requirements engineering [7] and rich internet applications [21]. These comparative studies help to get a better understanding of each of the web modeling methods, and mainly insight in its strengths and weaknesses. However, no comparative studies have been performed addressing modeling requirements of a particular web domain.

Hong, Van de Goor and Brinkkemper [13] provide a formal approach to the comparison of six object-oriented methods. By creating meta-models from a process, and data perspective as well, a uniform and formal representation of the methods is established. From all meta-models, a super set was constructed to which each of the methods were compared. By doing this, all similarities and differences between the six methods were presented. In this research, also a meta-modeling technique has been used for the uniformity of method comparison.

Besides the domain of web engineering, also the discipline of method engineering is related to this study. Method engineering is the engineering discipline is to design, construct and adapt methods, techniques and tools for the development of information systems" [3]. In method engineering, a special type of engineering is used in order to create methods that make it possible to adjust towards distinctive development project situations at hand, called situational method engineering [28]. Situational method engineering has been applied successfully for the assembly of situational methods [14, 28]. A situational method can be defined as an Information Systems (IS) method that is tuned to the situation of the project at hand [12]. Ralyte et al. [22] state that an assembly-based situational method engineering is an approach in which method fragments are extracted and stored in a method base driven by situational method requirements. According to Harmsen [12], a method fragment is a description of an IS engineering method, or any coherent part thereof. Method fragments allow for engineering situational methods in a way that it supports selection, storage and assembly.

2 A Method Association Approach

In this section, we describe a situational method engineering approach that compares, selects and develops web modeling methods for specific domains. Since many web modeling methods ignore the design and development of web applications that use preexisting features and components, we have adapted an existing assembly-based situational method engineering approach [28]. The approach has been adapted by adding two steps (step 2 and 5) that allow for association and selection of relevant web modeling methods. We call this approach a *Method Association Approach (MAA)*.

Figure 1 gives an overview of the MAA. The approach consists of the following seven steps:

1. Identify web engineering project situations
2. Identify feature groups of a software product
3. Select candidate methods for the identified feature groups
4. Model relevant method fragments in a method base
5. Associate feature groups with candidate method concepts
6. Assemble situational web design method
7. Validate situational web design method.

In the first step, project situations are identified by means of categorizing unique development project characteristics. In essence, development situations allow for the construction of situational methods . Three development project situations have been defined, namely: standard, complex and migration development projects. Standard development projects were defined as development projects that only need a little customization from a software development perspective. For the standard and complex development project situations, specific needs have been identified such as: ‘applying UML for modeling, since all developers have knowledge about this modeling language.’ Migration project situations have not been included, since this type of project not in the scope of this research.

During the second step, fourteen key feature groups have been identified. These feature groups can be employed as selection and association criteria of a given domain for the construction of situational web design methods. These feature groups are explained in more detail in section 3.1.

Based on a literature study, four mature web modeling methods and a propriety web design method were selected: Object-Oriented Hypermedia method (OO-H method) [11], UML-based Web Engineering (UWE) [15], Object-Oriented Web Solutions (OOWS) [20], Web Modeling Language (WebML) [5] and GX-Design method. The reason for incorporating these candidate web modeling methods were supported by three criteria: acceptance in the community, evolution of the approach and advanced CASE tool support. In literature, WebML, OO-H and UWE are perceived as 'prominent', since these are well documented and well-supported by design tools [29]. Also OOWS can be perceived as prominent, since it is based upon OO-method – an automated software production environment - which has been developed continuously since its inception in the mid 90s [19].

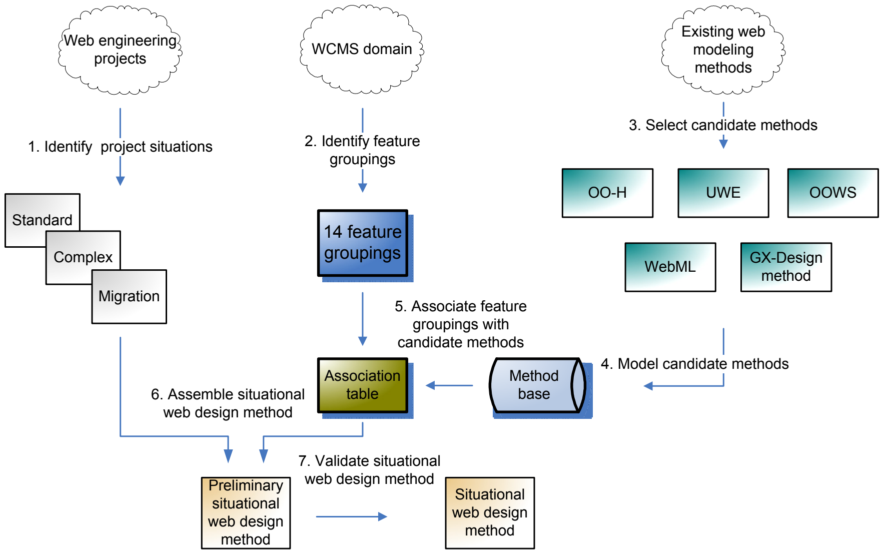

Fig. 1. The Method Association Approach

For modeling relevant method fragments (i.e. requirements engineering, design and implementation method fragments), a meta-modeling technique called Process Deliverable Diagram (PDD) was used [28]. The meta-modeling technique is based on UML standards. For more on the modeling of candidate methods, see section 3.

In step 5, the method association and selection is performed by using the association table. In the association table, key feature groups identified in step 2 are associated with candidate method. This method association step is explained in more detail in section 4.

Using the results extracted from the association table and taking in account the development project needs identified in step 1, a preliminary situational web design method was constructed. The situational web design method consists of four main activities: conceptual design, architecture design, presentation design and detail component design. All these design activities and their corresponding design deliverables contribute to the translation of requirements into designed components. Subsequently, these designed components will be developed and implemented in the development phases of the implementation project.

In the final step, the situational web design method has been validated by means of an expert group and two case studies in the WCM domain. The expert validation and the case studies have been carried out in GX, a software product company that develops and implements a WCMS called GX WebManager [27]. With regard to the expert validation, the situational design method was reviewed by two consultants and two software architects. With each of the experts, we have walked through the situational web design method and they were asked whether the method's process and deliverables were complete and correct. Based on the comments, the situational design method has been adjusted. Next to that, the constructed situational web design method has been tested in two development projects, whereof one was recently completed [30]. In both projects a WCMS has been designed conceptually, leaving out architecture and detailed component design.

3 Setting the Evaluation Criteria for Method Association

In this section, we discuss the evaluation criteria used for method association. First, we elaborate on the web modeling methods used for association, and second on feature groups that establish a web application domain.

Ceri et al. [5] state that the Web Modeling Language (WebML) is a notation for specifying complex web sites at a conceptual level. WebML is supported by the CASE tool WebRatio. WebML is also provisioned with a development approach based on Boehm's spiral model in order to build data-intensive web applications using the notation. WebML provides modeling primitives for the conceptual level by means of E-R schemas and, for hypertext modeling by their own propriety graphical hypertext model. Furthermore, WebML has evolved in the course of time by including design concepts like: workflow [2] and rich internet applications [1].

The UML-based Web Engineering (UWE) approach is an object-oriented, iterative and incremental approach for the development of web applications [15]. The approach focuses mainly on customized and adaptive systems. UWE uses 'pure' UML wherever is it possible and extends UML for navigation and presentation design with UML extension mechanisms, also called a UML profile. UWE is supported by a CASE tool called ArgoUWE. ArgoUWE facilitates the design process, namely: conceptual, navigational and presentation design. Recently, UWE has been evolved towards a Model Driven Development (MDD) approach [17].

The Object-Oriented-Hypermedia Method (OO-H) can be defined as a generic model based on the object-oriented paradigm that captures relevant specifics related to web application design [11]. Besides, OO-H extends OO-method with two inter-dependent diagrams addressing navigation and presentation. OO-H is supported by a CASE tool called VisualWade that deals with the design and implementation of web applications. Since its inception, OO-H has been evolved by the addition of behavioral models that deal with business processes [16] and a personalization framework [10].

Pastor et al. [20] describe the Object Oriented Web Solutions (OOWS) method as the extension of the software production method OO-method that introduces expressiveness to capture the navigational and presentational requirements of Web applications. OOWS is supported by a commercial MDA tool called OlivaNova. OOWS has evolved and adopted new design features, like business modeling and semantic web technologies [9].

The GX-Design method is a design method used for the implementation of GX-WebManager. The GX-Design method consists of three steps: architecture design, presentation design and detail component design. In architecture design, the relative context between the components is designed using the 4+1 view model of Kruchten [18], and this finally results in an architecture document. During presentation design, deliverables such as XHTML and CSS files are translated to dynamic pages. Parallel to presentation design, customer-specific components (e.g. connectivity) are described in a detailed component document containing design models, descriptions, and rationale. The GX-Design method took in a special status with regard to method association, since it is WCMS-specific and therefore different than all other methods.

In step 4 of the MA approach the web modeling methods have been modeled by means of meta-models to provide a uniform way of representation. Meta-models have also been applied effectively for the evaluation and comparison of methods [13]. During the process of storing all web modeling methods, a total of 32 method fragments have been extracted and stored. From these 32 method fragments, 20 method fragments represented the main design models of the web modeling methods and the other fragments represented abstract views on the web modeling methods. These method fragments were extracted from main literature resources of the web modeling methods. Besides, also available meta-models created in the MDWE project have been used in order to speed up the modeling process. Aiming at method integration, this project has presented meta-models of the following web modeling methods: WebML and OO-H [29].

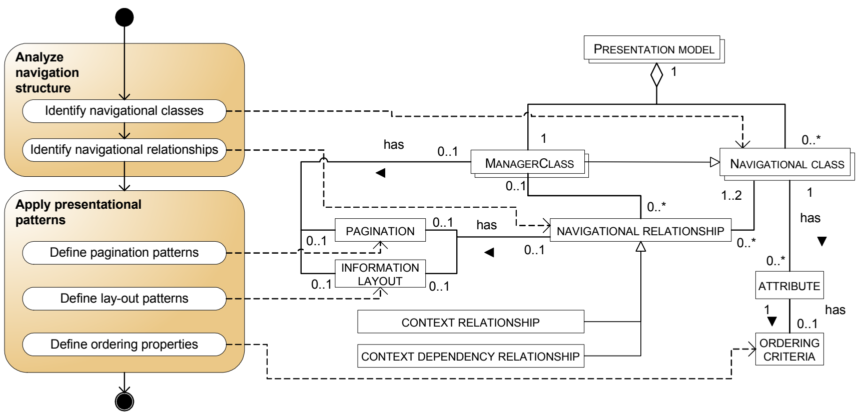

Fig. 2. A Process Deliverable Diagram of the presentational modeling phase in OOWS.

In figure 2, a PDD is presented of the presentational modeling phase in OOWS. This phase consists of two main activities and five sub-activities, in which presentational requirements are specified by means of patterns. In table 1 and 2, a description of all activities and concepts is given.

Table 1. Activity table for presentational modeling in OOWS.

Activity	**Sub-Activity**	**Description**
Analyze navigation structure	Identify navigational classes	For the presentational model, the NAVIGATIONAL CLASSES from the navigation model can be used, and therefore need to be identified..
	Identify navigational relationships	Similar to the NAVIGATIONAL CLASS, also the NAVIGATIONAL RELATIONSHIPS need to be identified before adding presentational requirements.
Apply presentational patterns	Define pagination patterns	In this sub-activity, a PAGINATION-pattern that specifies information "scrolling" and is applied to the MANAGER CLASS and NAVIGATIONAL RELATIONSHIPS
	Define lay-out patterns	A LAYOUT-pattern is defined that shows how content is displayed. The pattern is applicable to the MANAGER CLASS of NAVIGATIONAL RELATIONSHIPS.
	Define ordering properties	By defining ordering properties, a designer can specify a class population ordering using the value of one or more attributes.

Table 2. Concept table for presentational modeling in OOWS.

Concept	**Definition**
PRESENTATION MODEL	An abstract model that integrates the properties of the user interface into the existing Conceptual Schema.
NAVIGATIONAL CLASS	A class that provides a navigational view over a class in the Class Diagram.
ATTRIBUTE	A feature within a classifier that describes a range of values that instances of the classifier may hold.
NAVIGATIONAL RELATIONSHIP	An unidirectional binary relationship connecting navigational classes.
CONTEXT DEPENDENCY RELATIONSHIP	A relationships that represents basic information retrieval by crossing a structural relationship between classes.
CONTEXT RELATIONSHIP	A relationship that represents basic information retrieval, adding a navigation capability to a target navigational context.
MANAGERCLASS	A primary navigational class of an AIU.
PAGINATION	A pattern that specifies information "scrolling".
LAYOUT	A pattern that specifies the layout of the information presented.
ORDER	A pattern that defines a class population ordering using the value of one or more attributes.

3.1 Feature Groups

One added step (*step 2*) of the MA approach represents together with meta-models a main ingredient that enables the comparison of web modeling methods, called feature groups. When developing a suitable design method for a specific web application domain, one should have a complete overview of the design requirements of a particular web domain. Since these design requirements that represent preexisting components originate from business requirements these influence design. We define these design requirements *feature groups* as “a class of functional design requirements.” In short, feature groups enable comparison of existing web modeling methods by providing means to select or find the method or web design model that fits best with functional design requirements of a web domain.

Since no reference models are available in literature [26], feature groups have been gathered by means of analyzing seven useful WCM market resources and a document analysis on architecture and requirements documents, resulting in 113 features. By means of categorization and an expert validation, the list of features has been narrowed to a final list of 14 key feature groups.

The 14 identified feature groups are: (1) Authoring; (2) Authorization management; (3) Community technologies; (4) Connectivity management; (5) Content repository; (6) Deployment and replication; (7) Digital Asset Management; (8) E-forms / transaction management, (9) Layout and presentation management; (10); Multi-channel delivery and syndication; (11) Personalization; (12) Site Management; (13) Web usage Mining; and (14) Workflow. These 14 identified feature groups have been used for association with candidate method concepts, which is explained in the following section.

4 The Association Table

In the fifth step of the MA approach (figure 1), four web modeling methods have been associated by positioning feature groups against method fragments in a association table. In table 3 an excerpt of the association table is given showing the association of all identified feature groups with WebML. As identified in step 2 of the MA approach, feature groups are listed in the right 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 WCMS 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 product’s functionalities in relation to key feature groups of a given web domain.

In this study, functionalities of the WCMS called GX WebManager are mapped onto the feature groups. The design of a typical e-form within in WebManager is based on the following core concepts: step, handler, validation, router and field. In table 3, three web design models of WebML are presented including their main concepts. The excerpt shows that a handler can be designed with an entity, operation unit and activity in three different web design models. First, a handler represents a kind of service or operation performed on a (data) object during a transactional process part of e-form design in WebManager. A typical handler would be ‘create new user in database.’ In this case, the user can be represented as an entity in the E-R diagram of WebML. In a hypertext model of WebML, navigational requirements can be designed, which results in an create unit that is related to an entity unit. In this create unit (subtype of operation unit) a database query on the entity can be specified based on specific parameters. In a handler, also parameters can be initialized, queries can be fetched from a query manager, the query can be executed resulting in a new database record and a final redirect to a follow-up page. Alternatively, a handler can be specified by an activity performed by a web application in the Business Process Diagram (BPD). Additionally, WebML also provides the possibility for a designer to create generic operational units. As depicted in table 3, only six of the fourteen key concepts have been mapped onto WCMS functionalities, since not all feature groups could be mapped perfectly.

Table 3. An association table representing WebML (excerpt).

		WebML																					
		E-R schema								Hypertext model										BPD			
		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	Activity	Flow	Gateway	Swimminglane
E-Form / transact. Mgmt	Step								X				X		X				X	X			
	Handler	X											X							X			
	Validation												X					X					
	Router												X				X				X	X	
	Field	X																					
Personalization	Personalization rule	X										X	X			X		X					
	User profile	X		X				X					X										
	User access									X			X										
Site management	Page										X												
	Link																X						
	Structure									X													
	Media list													X									
	Anchor													X									
	Media overview													X									
	Page collection										X			X									
Connectivity management	Connection external app.												X				X	X					
	DB Query											X	X				X	X					
Authorization	User			X														X	X				X
	Role									X		X											X
	Permission									X		X	X					X					
Workflow	Activity											X							X	X			
	Status											X							X				
	Transition (flow)																X		X		X		
Authoring															X								
Content repository												X											
Deployment and replication										X													
Digital Asset Management												X	X	X									
Layout and presentation man.																							
Multi-channel delv. Sync										X				X									
Community technologies				X								X	X		X								
Web user mining											X		X				X	X					

The targets to which the feature groups are positioned are the method fragments of the four web modeling methods decomposed into three granularity levels: method, model, and concept [4]. Some web design models have not been included, since they have not given much attention in the main literature sources and other related proposals. We found that relationships between the original web design models, alternative proposals and proposals mutually are not always that clear. Next, we explicitly selected the main concepts of all method fragments for reasons of conciseness.

The ‘X’ in the association table implies that a method fragment or concept is needed for the design of a feature group or WCMS functionality. When a cell is left ‘blank’, a feature group or WCMS functionality is not addressed by any method fragment. The notation represents two extremes and does not allow for providing nuances and more rich information. We view the definition of a more detailed scale as future work. Two association strategies have been applied for the qualitative analysis on the association results being a feature group strategy and a web modeling strategy. In other words, one can find out what feature groups are addressed by all included web modeling methods, and the vice-versa. The purpose of the analysis is to find out how feature groups of a particular web application are addressed by all selected web modeling methods and more specific by its models and concepts. By performing this analysis, a functional overview of all web modeling methods from a WCMS perspective can be obtained.

The first association strategy used for qualitative analysis is called the ‘feature group strategy’. In this strategy, the starting point of analysis is a feature group, and its mapping on WCMS functionalities. From a WCMS functionality or feature group perspective, one can find out by which web design model or concept the feature group can be designed. Another objective of this strategy is to find differences between the proposed design concepts of the different web modeling methods.

The second association strategy used for analysis is called the ‘web modeling’ strategy. By taking the meta-model concepts and models as a starting point, the relative weighting of concepts and web design models part of a method is revealed. For instance, if a concept does not address a feature group, one could wonder whether this meta-model concept is useful. On the other hand, when a meta-model concept addresses many feature groups in comparison to others, one could say that this concept seems to be relatively more useful than others.

In table 4, some statistics of the three granularity levels of the web modeling methods are given. In total, 20 models (method fragments) containing 92 concepts of the four web modeling methods are used for association and positioned against the key feature groups and WCMS functionalities.

Table 4. Overview of models and concepts per web modeling method.

Method	**Models**	**Concepts**
WebML	3	23
UWE	6	25
OO-H	4	16
OOWS	7	28
Total	*20*	*92*

In short, the association table shows how each of the feature groups is covered by the web design models and concepts of the web modeling methods. Having knowledge about the coverage of WCMS feature groups by the web modeling methods, a selection of fitting method fragments can be supported. The association table provides a route map for existing web modeling methods and situational web design methods addressing a specific domain.

4.2 From Method Association to Method Assembly

By drawing conclusions upon the association table and taking into account the needs that come along with the different project situations, relevant method fragments can be selected. Based on the coverage of feature groups and the requirements specified in the development needs a situational web design method has been assembled. The assembled situational method consists of four main activities: conceptual design, architecture design, presentation design and detail component design. For conceptual design, four models have been added: a domain model and user model based on a UML class diagram, an UWE navigation model and a BPD derived from OOWS. With regard to the quality of the situational web design method assembly, assembly guidelines have been applied such as provided by Brinkkemper et al. [4]. These logic rules support the assembly of a meaningful method taking in account five method assembly criteria, such as completeness, consistency, efficiency, applicability and reliability.

5 Method Validation

In the last step of the MA approach the situational web design method has been validated by means of an expert group and two case studies in the domain of WCM. The expert group consisted of two software architects and two consultants. The experts have been selected based upon job function and experience with working procedures. In the expert validation questions were asked about the completeness and correctness of the methods, and more specific whether the relationships were perceived as correct and the steps as logical precedents. Based on the expert validation, 30 changes were requested varying from changes on a aesthetic level to changes with high impact. The changes have led to some refinements on the situational web design method.

After the expert validation, we have performed two case studies. One case study organization is a large Dutch Telecommunications Provider (TP) and the other a Dutch Governmental Organization (GO). The TP project had an estimated effort of 1500 man hours and the GO project's effort was estimated on 1400 man hours. In both studies, the complex route map of the conceptual design activity has been validated. For the TP project, sixteen use cases regarding a personal user environment have been designed from retrospective, and for the GO project, two use cases for an intranet application were designed. After the first design, in total three 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, tool supportability and applicability of the conceptual design deliverables. With regard to the GO process, also questions concerning the process in terms of structure, involvement and communication were asked. After filling-out the survey, informal interviews were held in order to discuss the usefulness of all deliverables and the situational web design method as a whole, superfluity and redundancy of information, and the availability of models in order to support customer communication.

In order to judge the quality of case study research, one can test the validity of data [30]. Construct validity was guaranteed by using multiple sources of evidence, which includes interviews, surveys, requirements documents and direct observations. A chain of evidence was created in order to establish the reliability of a case study. In this research, a case study protocol was used which functioned as an outline for performing the case studies and for the evaluation of the results. Next to that, all relevant data gathered during the case studies were managed in a case study database.

From both case studies we observed that the respondents were positive concerning the overall method, however, some differences were noted between roles and models. First, software developers were more positive about using domain models than consultants, since the latter considered the models too complex for customers. Both agreed upon the usefulness for designing customized components. With regard to the navigation model, the consultants had contradicting opinions about that. The GO consultant perceived the navigation model as helpful in order to gain more insight in the information structure, whereas the TP consultant questioned the supportability of the model with regard to customer communication and the added value to design as a whole. Besides, all respondents indicated that the user model should not be modeled separately from the domain model. Finally, a BPD was considered as the most readable and applicable model in comparison to other models. One software developer stated that the BPD makes complex transaction flows of an e-form more insightful in comparison to textual descriptions.

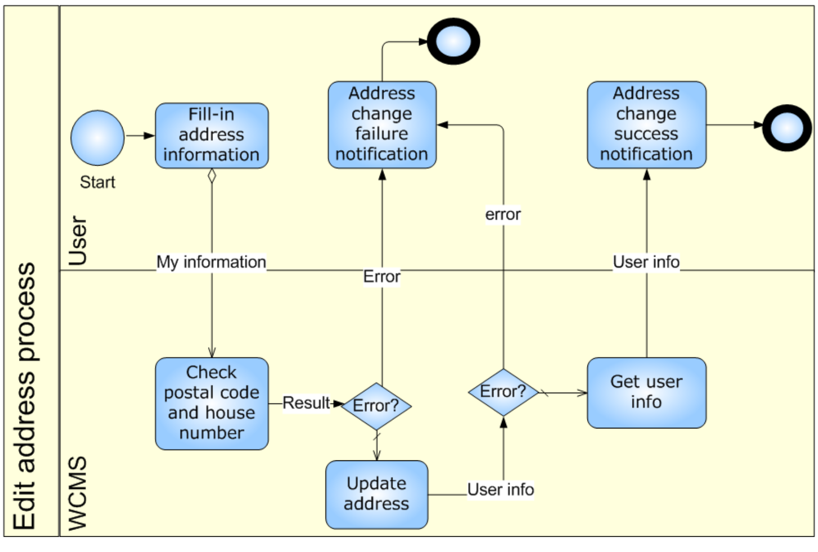

Fig. 3. A BPD representing an 'edit address process' of the TP project.

The BPD in figure 3 displays an edit address process for mobile phone subscribers which is part of a personal user environment. After the modification of the address, the new address information is validated. When properly validated and updated in the database, the result of the address modification is presented to the subscriber on a web page.

6 Discussion

Applying the association table in the MA approach has some limitations. In contrast to another formal comparison approach proposed by Hong et al. [13], no automatic build-in validation mechanism like a 'supermethodology' is present, which is a target to compare. Although, the absence of such validation mechanism does not lead to a high subjectivity of results in the association table, because feature groups have been described and validated, and meta-models derived from main literature resources have been used as a formal description of the web modeling methods.

Although, the qualitative and textual description of feature groups introduces another limitation in a way that a description does not exclude or bring insight into overlapping feature groups. Therefore, we suggest to add a formalization step in the MA approach with the aim to decrease the ambiguity between and of all feature groups. This can be achieved by creating a feature group ontology for a specific web domain.

Another limitation in the association table is that transformational relationships are not made clearly visual and traceable in the association table. With this we mean that it is hard to include transformational concepts multiple times in the association table, since this will make the association table lose its clearness. Due to the lacking rules of representing transformational relationships in the association table, we therefore suggest to use the association table together with the meta-models in order to get a complete overview of a particular web design model. Alternatively, another way to solve this issue is to associate the main concepts to the feature groups that are most related to a particular design concern (e.g. navigation, workflow, etc.).

7 Conclusion and Further Research

In this paper we present an approach for the development of situational web design methods by means of method association. By identifying feature groups and creating meta-models of all candidate web modeling methods, we were able to associate feature groups of a web application domain with web modeling methods by means of a association table. Method association is a helpful technique in order to select and assembly method fragments that fit with domain situational factors. We have analyzed five mature web modeling methods. The approach has been validated by means of an expert group and two case studies.

In future research, the MA approach can be improved by means of providing metrics for the selection of relevant methods. Rossi and Brinkkemper [23] have proposed a set of metrics that measures the method's complexity, and in this way a means to method association and selection is given.

References

1. Bozzon, A., Comai, S., Fraternali, P., and Carughi, G. T. 2006. Conceptual Modeling and Code Generation for Rich Internet Applications. In Proc. of the 6th International Conference on Web Engineering (ICWE 2006), Palo Alto, CA, USA. ACM, 353–360.
2. Brambilla, M., Ceri, S., Fraternali, P., & Manolescu, I. (2006). Process modeling in web applications. ACM Trans. Softw. Eng. Methodol., 15(4), 360-409.
3. Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information & Software Technology*, 38(4), 275-280.
4. Brinkkemper, S., Saeki, M., & Harmsen, F. (1999). Meta-modelling based assembly techniques for situational method engineering. *Inf. Syst.*, 24(3), 209-228.
5. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S. & Matera, M. (2003). *Designing Data-Intensive Web Applications*. Morgan Kaufman.
6. Deshpande, Y., Murugesan, S., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M. & White, B. Web Engineering. *J. Web Eng.*, 1, 3-17 (2002).
7. Escalona, M. & Koch, N. (2004). Requirements engineering for web applications: A comparative study. *Journal on Web Engineering*, 2(3), 193-212.
8. Escalona, M. J., Torres, J., Mejias, M., Guti_errez, J. J., & Villadiego, D. (2007). The treatment of navigation in web engineering. *Adv. Eng. Softw.*, 38(4), 267-282.
9. Fons, J., Pelechano, V., Pastor, O., Albert, M., Valderas, P., & Torres, V. (2007). *Web Engineering: Modelling and Implementing Web Applications*, chapter Applying the oows model-driven approach for developing web-applications. the internet movie data-base case-study, (pp. 65-108). Springer London.
10. Garrigos, I., Gomez, J., Barna, P., & Houben, G.-J. (2005). A reusable personalization model in web application design. *In Proceedings of WISM'05* (pp. 40-49). Sydney, Australia: University of Wollongong, School of IT and Computer Science.
11. Gomez, J. & Cachero, C. (2003). *Information Modeling for Internet Applications*, chapter OO-H Method: extending UML to model web interfaces, (pp. 144-173). IGI Publishing: Hershey, PA, USA.
12. Harmsen, F, Brinkkemper, S and Oei, H. (1994). Situational method engineering for information system project approaches', In Verrijn Stuart, A A and Olle T W (eds), *Methods and associated tools for the information systems life cycle*. Proc. of the IFZP WG8.1 Working Conference CRIS'94 Maastricht, September 1994, North-Holland, Amsterdam, pp 169-194.
13. Hong, S., van den Goor, G., & Brinkkemper, S. (1993). A formal approach to the comparison of object-oriented analysis and design methodologies. *In Proceedings of 26th Hawaii International Conference on System Sciences*, Wailea, HI, USA (pp. 689-698).
14. Karlsson, F. & Agerfalk, P. (2004). Method configuration: adapting to situational characteristics while creating reusable assets. *Information & Software Tech.*, 46(9), 619-633.
15. Koch, N. & Kraus, A. (2002). The expressive power of UML-based web engineering. In D. Schwabe, O. Pastor, G. Rossi and L. Olsina (Eds.) *Second International Workshop on Web-oriented Software Technology (IWWOST02)*, pp.105–119.
16. Koch, N., Kraus, A., Cachero, C., & Meli_a, S. (2004). Integration of business processes in web application models. *J. Web Eng.*, 3(1), 22-49.
17. Koch, N. (2007). Classification of model transformation techniques used in UML-based web engineering. *IET Software Journal*, 1(3).
18. Kruchten, P. (1995). The 4+1 view model of architecture. *IEEE Softw.*, 12(6), 42-50.
19. Pastor, O., Insfran, E., Pelechano, V., Romero, J., & Merseguer, J. (1997). OO-method: An OO software production environment combining conventional and formal methods. *In*

CAiSE '97: Proceedings of the 9th International Conference on Advanced Information Systems Engineering (pp. 145-158). London, UK: Springer-Verlag.

20. Pastor, O., Fons, J., Pelechano, V., & Abrahao, S. (2006). *Web Engineering: Theory and Practice of Metrics and Measurement for Web Development*, chapter Conceptual Modeling of Web Applications: The OOWS Approach, (pp. 277-301). Springer-Verlag Berlin.
21. Preciado, J. C., Trigueros, M. L., Sanchez, F., and Comai, S. (2005). Necessity of methodologies to model Rich Internet Applications. *In Proc. of the 7th IEEE International Workshop on Web Site Evolution*, Budapest, Hungary. 7–13.
22. Ralyte, J., Deneckere, R., & Rolland, C. (2003). Towards a generic model for situational method engineering. *Lecture Notes in Computer Science*, 2681, 95-110.
23. Rossi, M., Brinkkemper, S. (1996). Complexity Metrics For Systems-Development Methods And Techniques, *Information Systems*, 21, 2.
24. Siau, K., Rossi M. (1998). Evaluation of Information Modeling Methods - A Review. *In Proceedings of the 31st Annual HICCS*, 5, (1998), p. 314.
25. Schwinger, W. and Koch, N. 2006. *Modeling Web Applications*. In Web Engineering – Systematic Development of Web Applications, G. Kappel, B. Pröll, S. Reich, and W. Retschitzegger, Eds. Wiley, 39–64.
26. Souer, J., van de Weerd, I., Versendaal, J., & Brinkkemper, S. (2007). Situational requirements engineering for the development of content management system-based web applications. *Int. J. Web Eng. Technol.*, 3(4), 420.
27. Van Berkum, M., Brinkkemper, S., & Meyer, A. (2004). A combined runtime environment and web-based development environment for web application engineering. In *Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004)*, Lecture Notes in Computer Science 3084, Springer Berlin: Heidelberg, 307–321.
28. Van de Weerd, I., Brinkkemper, S., Souer, J., & 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), 521-538.
29. Wimmer, M., Schauerhuber, A., Schwinger, W., & Kargl, H. (2007). On the integration of web modeling languages: Preliminary results and future challenges. *Workshop on Model-driven Web Engineering (MDWE),* Como, Italy, July, 2007.
30. Yin, R. K. (2003). *Case Study Research : Design and Methods*. SAGE Publications.