Journal of Universal Computer Science, vol. 18, no. 17 (2012), 2474-2492
submitted: 16/1/12, accepted: 29/8/12, appeared: 1/9/12 © *J.UCS*

Service-Oriented Development of Web Information Systems

Valeria de Castro
(Rey Juan Carlos University, Móstoles, Madrid, Spain
valeria.decastro@urjc.es)

Juan Manuel Vara
(Rey Juan Carlos University, Móstoles, Madrid, Spain
Juanmanuel.vara@urjc.es)

Esperanza Marcos
(Rey Juan Carlos University, Móstoles, Madrid, Spain
esperanza.marcos@urjc.es)

Abstract: During the last years, Web Information Systems have evolved from simple information sources to systems offering services to end-users. The development of this kind of systems presents different challenges, such as the alignment between business services and their implementation, and the way business processes are placed in the system. To address this type of challenges this paper presents a methodological and technical proposal for Service-Oriented development of Web Information Systems. It follows a model-driven approach, defining a set of models at different levels of abstraction and the model transformations needed to connect them. Besides, the development of a conference management system is used as case study to illustrate the proposal.

Keywords: Service-Oriented, Model-Driven Development, Web Information Systems
Categories: H.5.0, H.5.2, H.5.4, D.2.2

1 Introduction

Service-Oriented Computing (SOC) is a computing paradigm which uses services as the basic constructs to support rapid, low-cost and easy composition of distributed applications [Papazoglou and Georgakopoulos, 03]. Thus, organizations could benefit from adopting SOC, using Web Services and Service-Oriented Architectures (SOA) to build their systems. In the last years, the development of Web Information Systems (WISs) has been influenced by the advent of these new technologies and the quantity and quality of the services offered through the Web. As the complexity of systems grows, so does the difficulty in using them. Unlike users of traditional Information Systems, who spent a long time familiarizing themselves with the features and design of a given system, users of WISs only know what they want to do, but not how to do it. Today, users can access many kinds of services, ranging from the simplest one (e.g. looking for information, sending a message, etc.) to more complex (e.g. booking a flight or processing a purchase order). Usually, the latter involve a business process that the user does not know, although it is obvious that in some case the common sense could help. However, it might be difficult for him to identify the right sequence of links in order to get what the service offer.

There is a great number of well known methodologies for WIS development ([Baresi et al., 00], [Ceri et al., 00], [Díaz and Aedo, 07], [Hennicker and Koch, 00], [Schewe and Thalheim, 05]). Although most of them were defined before the advent of the SOC paradigm, they already consider the concept of *service* ([Brambilla et al., 03], [Brambilla et al., 06], [Koch et al., 04], [Schmid and Rossi, 04]). However, they do not consider such element (the service) as first class modelling elements for the development of the system. Hence, there is a need for new methods and techniques to support a complete SO development process considering *services* as a first class elements from the earlier stages of the process. Note that following a SO approach does not imply that the system should be necessarily implemented by Web services: this is just one among several options (perhaps the most recommendable), but other options based also in services technology could be chosen.

Model-Driven Engineering is an evolving and promising approach to software development that deals with the provision of models, transformations between them and code generators to address software development [Schmidt, 06]. MDE proposals and more specifically the OMG specification, Model Driven Architecture (MDA) [Miller and Mukerji, 03], helps addressing one of the main issues in SO development: dealing with the alignment between business services and SOA architectures [Harmon, 04].

In this paper, we present a proposal for service-oriented development of WIS following a model-driven approach. Therefore, it is composed of a modelling process and the corresponding tool support. The inputs of the modelling process are the business services which are identified from a business model; the output is the navigational model from which is possible to generate the application code. In contrast with existing proposals for WIS development, that start from conceptual data model to build the navigation model [Escalona et al., 07], our proposal follows a service-oriented approach, that is, it takes into account the *services* required by the end-users. Moreover, the method defines the concept of *routes*, which corresponds to the business process that a given user has to follow to get a given service. Since the entry for every service is already identified in the main menu and the routes are signposted in the different pages, the user of the WIS can easily carry out the required service.

Our proposal has been validated in two ways. On the one hand, a set of case studies has served to validate the applicability of the proposal. In particular, this paper presents the development of a WIS for conference management. On the other hand, an empirical study has been carried out in order to test the easily of use of WIS which included routes [De Castro et al., 11]. The empirical study has permitted us to discover that, given two WISs with the same functionality and identical interface styles, users preferred the WIS with routes and perceived it to be more easily navigable.

The rest of the paper is structured as follows. First, in Section 2 analyses existing methodologies for WIS development. Section 3 presents the modelling process proposed. Section 4 describes the MDA tool developed to support such process and Section 5 illustrates the application of the modelling process proposed through a case study using the supporting tool. Finally, Section 6 concludes summarizing the main contributions and outlining future work.

2 Related Work

An analysis of current literature related to service-oriented development shows that, despite the impact of the SOC paradigm on software development and on the way systems can be constructed, there are still few methodologies facilitating software development based on it. The convergence of service orientation and model-driven software development is probably one of the most important frameworks for the promise of rapid, accurate development of software to serve users' goals [Watson, 08]. Our proposal aims to bring together the benefits of both approaches, model-driven and service orientation, and apply them for the Web development.

In the last few decades, many methodological proposals have arisen for the building of Web systems or applications in a systematic and rigorous way, giving rise to the new profession of Web Engineering [Ginige and Murugesan, 01]. In general, such proposals were mainly adaptations of classical methodologies for specific domains. Thus, some studies appeared in fields of hypermedia and multimedia, like: HDM [Garzotto et al., 93], RMM/ERMM [Isakowitz et al., 95], OOHDM [Schwabe et al., 01], UWE [Hennicker and Koch, 00] and W2000 [Baresi et al., 00], while other proposals came directly from approaches for Database development, like the ones from Fraternali [Fraternali, 99], Araneus [Mecca et al., 99], WebML [Ceri et al., 00], or the works [Schewe and Thalheim, 05] and [Schewe and Thalheim, 05b] which have stated bases for the conceptual modelling of large-scale data-intensive WIS. Moreover, methods like WSDM [De Troyer and Leune, 00] began to design Web sites in a user-oriented manner, and others extended development methodologies for general purposes [Gómez et al., 01].

However, over the last few years, SO paradigm and the new technological proposals for the Web (i.e., XML, Web services, business process automation, Web semantics, etc.), have brought about an evolution of techniques and methodologies for WIS development. Thus, new extensions have appeared. For instance, there are proposals for representing business processes in WIS, such as the extensions to WebML [Brambilla et al., 06], UWE and OO-H [Koch et al., 04], OOHDM [Schmid and Rossi, 04], and ADM [Díaz and Aedo, 07] among others. As well, there are a number of proposals for representing WISs that interact with Web services, such as [Brambilla et al., 03], [Koch et al., 09], and the work [Ma et al., 09] which is focused on data-intensive services. More recently, some methodologies have evolved to support also the development of Rich Internet Applications (RIA), such as ([Koch et al., 09], [Rossi et al., 08], [Comai and Toffetti Carughi, 07]).

The proposal set out here has some similarities with the best known proposals in this area, as well as important differences. Our method has two main characteristics, both of them derived of proposing a service-oriented approach, that distinguish it from previous proposals: a) the focus or orientation followed for develop the WISs, which proposes a different process, focused on the services required by the user for its construction; and b) the characteristics of the navigation model proposed by the method, which facilitates users navigation over WIS by defining a main menu that identifies the services needed by the user and the signposting of the sequence of steps to properly carry out each service, that we call a "route".

A similar idea to the used here to build the routes was proposed in [Schewe and Thalheim, 05], [Ma et al., 09] with the notion of "storyboarding". The storyboard identifies the ways user may chose to interact with the system; however a storyboard is not necessarily associated to the idea of a service. In our work we want to focus on the identification of services as first class elements for the development of a WIS. Thus, our aim is to provide of a model-driven process which starting from the identification of services, allows the service-oriented development of WISs.

3 A modelling process for service-oriented development of WIS

The modelling process proposed in this work is part of a model-driven framework for development of WIS. The framework, which is depicted in [Fig. 1], can be thought as a methodological proposal composed of simple methods, each one focused on more specific problems, like developing the Database (DB) of the system [Vara et al., 09] (content), modelling its behaviour [De Castro et al., 09], or designing its navigation model [De Castro et al., 04] (hypertext). This work focuses on the latter. In particular, we present the modelling process whose final output is the navigation model and its implementation in a MDA tool. The models involved in the modelling process presented here are highlighted in [Fig. 1].

This section introduces the modelling process and the main concepts considered by the different models that compose it: services and routes.

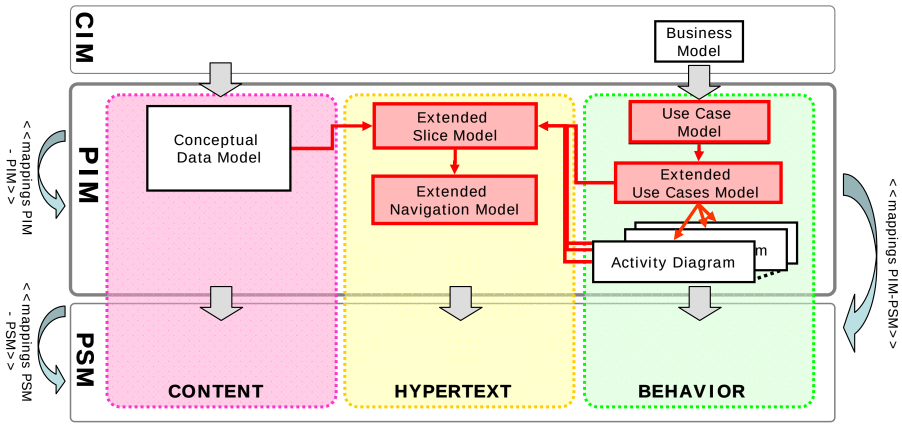

Figure 1: Model-driven framework for WIS development

A unifying concept of service is that it is a client-provider interaction that creates value [IBM], [Bergholtz et al., 10]. The value is created because this interaction allows satisfies a specific need of the client. In this work, we define a **service** as *"the functionality offered by a system to satisfy a need of a user"*.

Due to our aim is to provide a model-based process for the development of WIS, we decide to use standard UML modelling elements to represent the each on of the concepts defined in this work. Thus, we propose to use a Use Case Model to represent the service offered by a system, mainly because it represents functionality. In this work, we propose a taxonomy that collects different types of use cases and their relation with the traditional concept of Use Case (see [Fig. 2]):

- A ***business service*** is a special type of use case which represents a service offered by the system satisfying a specific need of the user, e.g. to submit a paper through a conference management system. This concept is represented in the **Use Case Model**.
- A ***basic use case*** is an atomic functionality needed to carry out a service, e.g. in the previous example, the act of login into the system is needed to submit a paper. Regarding just the presentation matter, a basic use case can be functional or structural. A basic use case is said to be ***structural*** when its functionality is only to provide with a data view to the user, e.g. viewing the data of an author. On the other hand, ***functional*** basic use cases imply some kind of user interaction, like entering some data. For instance, to submit a paper, the author has first to register himself as an author. Functional and Structural basic use cases are represented in the **Extended Use Case Model**.

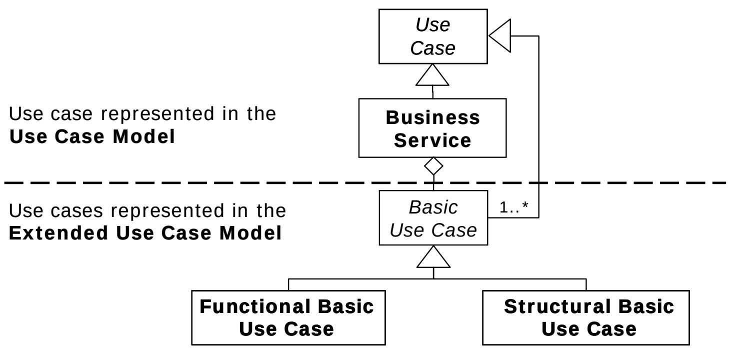

Figure 2: Use Case taxonomy for WIS development

The other basis of the modelling process proposed is the concept of ***route***, defined as “the sequence of steps that the user must follow to execute a service”. Whenever someone is planning a trip, it turns out very useful having some kind of map showing how to reach the destination. With a predefined route at hand it is easier to make the correct decision whenever a junction is found in the road. Likewise, when the user access a WIS for the first time, the only thing s/he knows is what s/he deserves to do but typically s/he does not know how to do it since a WIS, unlike traditional information systems, uses to lack of a user manual. Some kind of mark indicating the route to follow in order to obtain a given service will help the users to know exactly which hyperlink they should choose, which is the next activity they have to complete or how many activities rest to complete the route/service. To that end, both the **Extended Slice Model** and the **Extended Navigational Model** include a route for each service identified in the Use Case and Extended Use Case models (see [Fig. 1]). The route represents the sequence of steps to follow in order to complete the service and it is defined by means of an activity diagram. This way, a route in the navigation model is the materialization of the business process that implements a given service.

All the models using in the proposed modelling process are conforms to the standard UML 2.0 [OMG]. One of the most innovative aspects of the modelling process proposed in our work is the way the data and service are considered. While the vast majority of the existing methodologies that consider the navigational model propose building it from the conceptual data model to complete it later by adding functionality, our proposal propose to build the structure of the navigation model (represented by the extended slice model) based on the use case model and the extended use case model, in which only the services required by the user are represented. [Fig. 3] shows the proposed process.

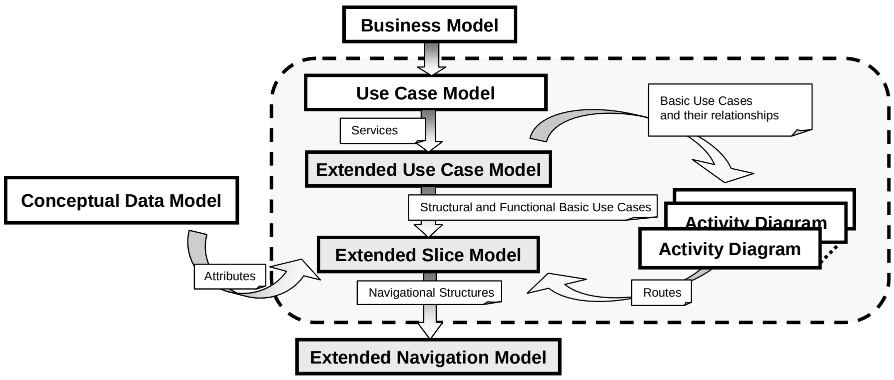

Figure 3: Model-driven process

For the sake of space in this paper we are not going to describe the metamodels defined for each model presented in the method. However a more detailed version of them can be found in [Cáceres et al., 06], and [Section 5] illustrates their use through a case study.

[Table 1] collects the transformation between models of our proposal in a set of rules defined in natural language. According to [Miller and Mukerji, 03], as some of the mapping rules of the transformation process proposed require design decisions, it is not possible to automate them completely. As a result, right column of [Table 1] makes the distinction between the mapping rules that can be completely (C) and partially (P) automated.

From the Use Case model (UCmodel) into extended use case model (XUCmodel)	**Aut.**
1. Every *Business Service* (BS) found in the *UCmodel* will be shredded into one or more *Structural Basic Use Cases or Functional Basic Use Cases* (SBUC or FBUC). The new *SBUC or FBUC* will be identified with the name of the respective *business Service*, followed by the name chosen for the new *Basic Use Case*	**P**
1.1 Every *Basic Use Case* will be identified as a *Structural* or *Functional Basic Use Case.*	**P**
From the use cases (UCmodel) and the extended use case model (XUCmodel) into activity diagram	
2. For every *BS* identified at the *UCmodel* there will be a new *Activity Diagram.*	**C**
3. For every *SBUC or FBUC* corresponding to a given *Business Service,* there will be an activity in the *Activity Diagram* generated for that *BS.*	**C**
4. Every *extend* association identified in the *XUCmodel* will be represented in the *Activity Diagram* by a fork. The activity corresponding to the source *Basic Use Case* of the *extend* association (*SAct*) must be previous to the activity corresponding to the target *Basic Use Case* of the *extend* association (*TAct*).	**C**
4.1 If the *extend* association has only one source, the fork will present the *SAct* as an alternative to another flow with no activities. Later, both flows will meet.	**C**
4.2 If the *extend* association has several sources, the fork will present the different *SActs* as mutual alternatives to another flow with no activities. Later, all these flows will meet.	**C**
5. Whenever an *include* association is found in the *XUCmodel*, the Activity corresponding to the source *Basic Use Case* of the *include* association (*SAct*) must be subsequent to the activity corresponding to the target *Basic Use Case* of the *include* association (*TAct*).	**C**
5.1 If the *include* association has several targets, the designer must decide the appropriate sequence for the different *TActs* (that will be obviously previous to the *SAct*).	**P**
From the activity diagrams (AD), **conceptual data model (CDmodel)** and **extended use case model into extended slice model (XSmodel)**	
6. For each *SBUC* in the *XUCmodel*, there will be a *Structural Slice* in the *XSmodel*	**C**
7. For each *FBUC* in the *XUCmodel*, there will be a *Functional Slice* in the *XSmodel*	**C**
8. Attributes for each *Slice* in the *XSmodel* will be obtained directly from the respective classes in *CDmodel*.	**P**
9. Every *AD* will be represented in the *XSmodel* as a business service *route* named by the *BS* corresponding to the *AD*.	**P**
10 Each fork in any *AD* will be represented in the *XSmodel* by a *sub-route*. The name for the *sub-route* will be composed by the name of the respective *BS* plus a new name to univocally identify the *sub-route*. That is *Servicename.IDSub-route*.	**P**
From the extended slice model (XSmodel) into extended navigational model (XNmodel)	
11. The *XNmodel* will include a main *menu* with one item for each *route* included in the XSmodel.	**C**
12. Each fork in the XS*model*, giving rise to a sub-route, will be mapped to an *index* in the *XNmodel*	**C**
13. Each association whose target element is a *Structural Slice* in the *XSmodel* will be mapped to an *index* or a *query* in the *XNmodel*. The designer will make the decision for each case.	**P**

Table 1: Mapping rules for the Service Oriented development of WIS

4 Supporting Tool

This section introduces the tool supporting the modelling process presented in previous section. It has been developed using the Eclipse framework (http://www.eclipse.org) and bundles the plug-ins needed to support the edition of the different models that has to be created during the development process, the model transformations that bridge them and the code generation templates that produce the navigational skeleton of the WIS.

4.1 Metamodel Implementation

The methodological proposal presented in this paper was initially defined by means of UML profiles [De Castro et al., 04]. However, we decided to shift to Domain-Specific Languages (DSLs) [Mernik et al., 05] when we faced the task of developing the mappings between them. In fact, existing technology for metamodelling and model transformations made it more convenient to express our models with a set of well-defined metamodels [Bézivin, 04].

In particular, we used the Eclipse Modelling Framework (EMF) [Budinsky et al., 08], a meta modelling framework that was devised to be extended and provides with the tools needed to define, edit and handle (meta)-models. We used EMF to define a set of MOF-based DSLs, starting from the behavioural modelling elements of UML 2.0 [OMG]. To that end, each one of these DSLs was defined in terms of the Ecore metamodel, the metamodelling language of EMF. From an ecore metamodel, EMF generates a set of plug-ins that bundle the JAVA code to provide with runtime support for graphically editing, manipulating, reading, and serializing models conforming to such metamodel. Nevertheless, this support is limited to simple tree-like editors.

Besides, in order to ease the task of handling models we developed a graphical editor for every DSL bundled in the tool. To that end, we used the Graphical Modelling Framework (http://www.eclipse.org/modeling/gmf/) that allow depicting the models graphically in a UML-like way. Some screen-captures from the different editors can be found in [Section 5].

4.2 Mapping Implementation

This work adopts the approach presented in [Cáceres et al., 06] for the development of model transformations. Following these guidelines, we defined the graph-transformation rules that formally specify the transformation rules described in [Table 1].

In order to implement these rules we use ATL [Jouault and Kurtev, 05], a model transformation language framed in Eclipse. ATL uses EMF to handle models, to serialize, to navigate and to modify them. We have defined three ATL model transformations to support the automation of the mappings between models depicted in [Fig. 1], starting from the Use Case model.

[Fig. 4] illustrates the model transformation process for the specific scenario of mapping Extended Slice models to Extended Navigation models. The root of the process is the Ecore metamodel. Next, the source and target metamodels are defined by instantiating the abstractions provided by the Ecore metamodel. They are said to conform to the metamodel. Finally, the model transformation engine executes the

ESM2ENM model transformation. Such transformation specifies a set of rules that encodes the relationships between the elements from the source and target metamodels in an ATL program. Once the ATL transformation is executed, the ATL engine tries to match the source pattern of each rule with some construction from the source model.

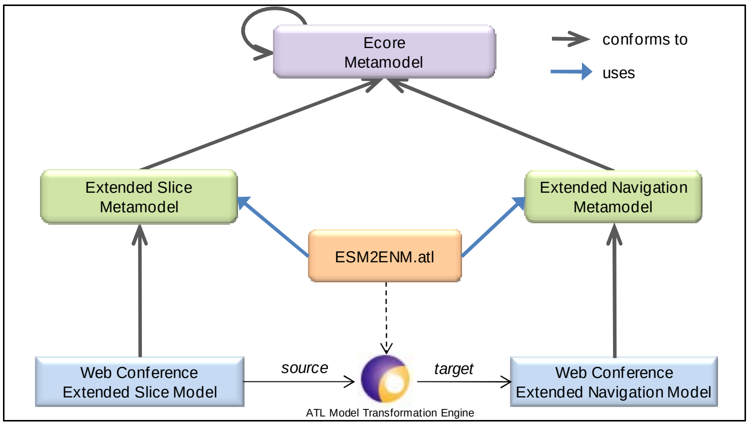

Figure 4: ATL Model Transformation: Extended Slice model to Extended Navigation model

To illustrate this process, [Fig. 5] shows the rule for mapping slice objects between the Extended Slice model and the Extended Navigation model.

The source pattern of the `StructuralSliceESM2StructuralSliceENM` rule matches every `Structural Slice` object found in the source model (Extended Slice model). The target pattern states that a `StructuralSlice` object is to be created in the target model (Extended Navigation model) for each matching. Besides, the different properties of the source object are copied to the target one by means of simple bindings (lines 94-96). A more complex binding is needed to match the input flows (referred to as `steps`). If the source slice is the beginning of a route, it does not contain any input step (line 98). Then, a new step between the target slice and the main menu has to be added to reflect this fact. To that end, the `Source2Step` rule is called (lines 98-99). Otherwise, the input steps from the source slice are copied to the target slice (line 101).

4.3 Introducing design decisions to drive the mapping

As we have mentioned in [Section 3], there are some mapping rules that require design decisions from the developer. In [Vara, 09] we show how weaving models are used as annotation models to drive the execution of model transformations. In particular, we use the ATLAS Model Weaver (AMW) [Didonet Del Fabro et al., 06] tool for defining and handle weaving models in the context of M2DAT. In this work we use a variation of such approach. The idea is to use a weaving model to state how the concepts of different (source) models have to be combined to produce the target model. For instance, [Fig. 6] shows an overview of process for producing the Extended Navigation model.

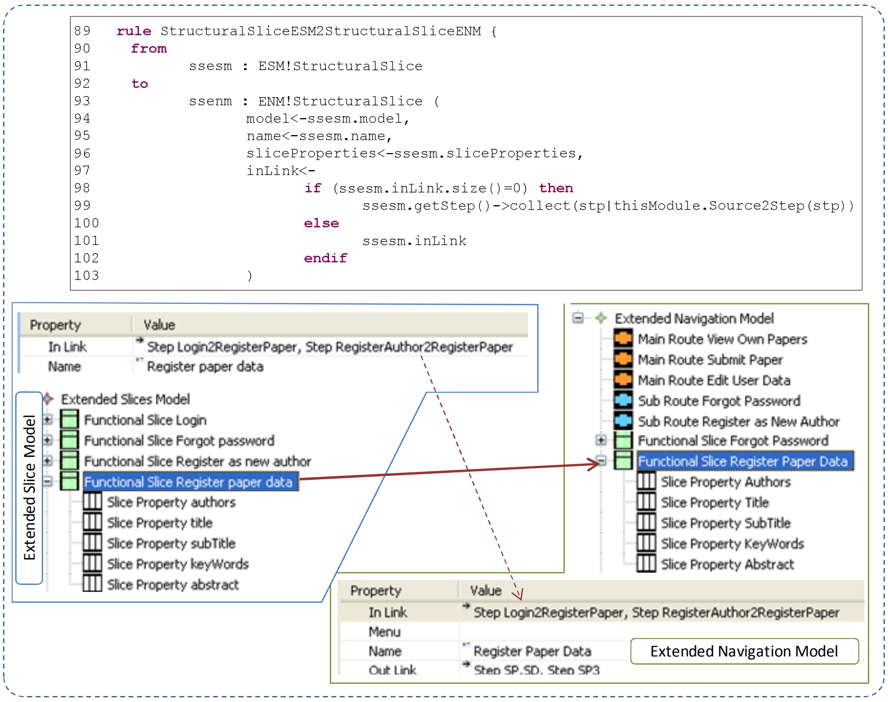

Figure 5: ATL code excerpt and partial views of involved models: mapping Slice objects

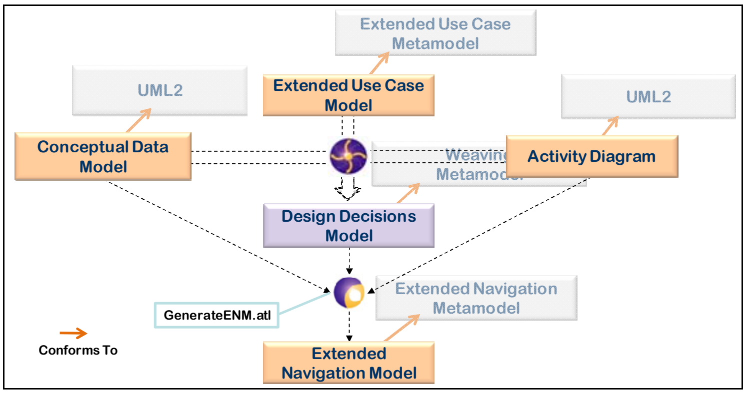

Figure 6: Introducing design decisions on model transformations

The decisions on how to combine the data from the Conceptual Data model, the Extended Use Case model and the Activity Diagrams are collected in a weaving model. Such model (next to the three mentioned) are consumed by the `GenerateENM` ATL transformation to produce the Extended Navigation model. To ease its understanding, the corresponding metamodels are shown in the background.

To illustrate this process, [Fig. 7] shows a simple example using some partial views of the models from the case study. In particular we focus on the rules 7 and 8 from [Table 1]. They dictate that Functional Basic Use Case objects have to be mapped into Functional Slices whose properties have to be collected from the Conceptual Data model. Such rules are coded in the ATL rule `SBUC2StructuralSlice` shown in the lower part of the figure.

Note that the weaving model includes a Link object that relates the `Paper Class` object from the Conceptual Data model and the `Show Paper Data` Structural Basic Use Case from the Extended Use Case model. When the `GenerateENM` transformation processes those models, the `SBUC2StructuralSlice` rule matches the later (line 157). As a response, the target pattern of the rule creates a Structural Slice object in the target model (line 159) and copies its name from the source object. Next, the rule has to create the properties of the slice. To that end, the `getSliceProperties` is called (line 161). It is a helper (ATL auxiliary function) that uses the information provided by the Link object from the weaving model, to identify which is the object from the Conceptual Data model whose properties have to be mapped to the just created slice. For each property, the `UmlProperty2 SliceProperty` rule creates a property in the slice object (line 162).

Working this way we benefit from several advantages such as those decisions are persisted in a model, thus we can retrieve them at any time and re-generate the target model. Moreover we avoid polluting the models handled along the development process with details from other domains.

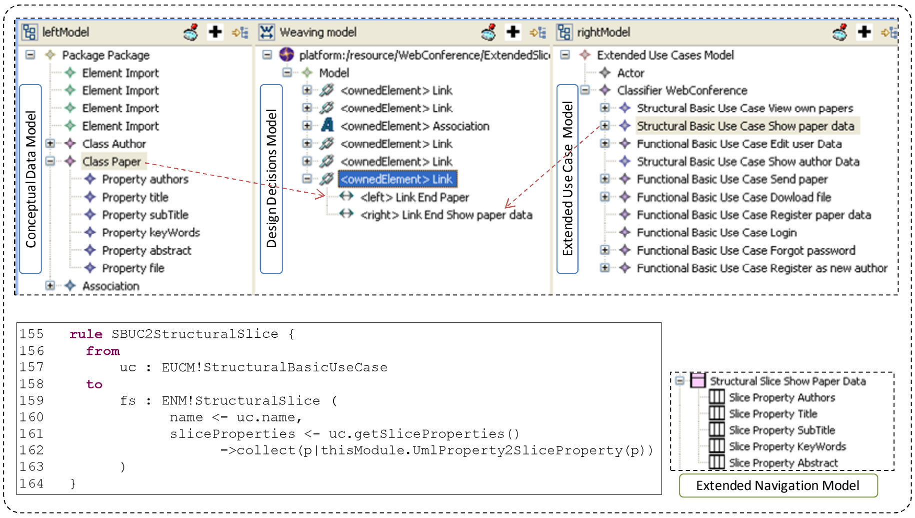

Figure 7: ATL code excerpt and partial views of involved models: introducing design decisions

5 Case Study

In this section, we apply our proposal for Service Oriented development of WIS to a case study in order to illustrate the process, the models and the mapping rules in depth. The case study is a WIS for conference management called *WebConference* [WebConference]. It supports typical tasks involved in conference management systems such as: Program Committee (PC) members' nomination, paper submission, etc. To present the case study we use some screen captures from the supporting prototype.

5.1 Generating the Use Case Model

In the use case model, we represent the actors and the business services required by the users of the WIS. [Fig. 8] shows the use case model for the WebConference WIS. In this application there are at least three different roles that a user interacting with the system may play (`Author`, `PC Chair` and `PC Member`). Each role is represented as an actor in the use case model. Here we will just focus on the business services that an author needs from the system. Therefore, by the end of the process we will obtain the Extended Navigation model for authors.

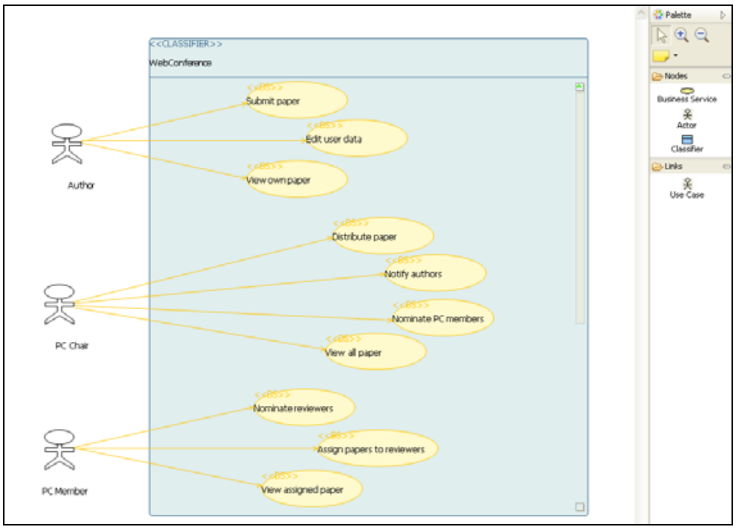

Figure 8: Use case model for Author, PC Chair and PC Member

In the model, we identify as business services the services provided by the system, taking into account just those required by the user. In our example, an author may need to submit a paper, to view his papers or to edit his author data. Thus, in the use case model we have identified the next business services (for Author): `Submit Paper`, `View Own Papers` and `Edit User Data`.

5.2 Generating the Extended Use Case Model

This model aims at extending the Use Case Model to include some use cases that do not represent business services, but are needed in order to execute the business services previously identified in the Use Case Model. Next, we focus on the Submit Paper service to illustrate the process.

As can be seen in [Fig. 9], firstly, we have to identify the functionalities required by the system to carry out each service. For instance, to submit a paper the author must log in and then he has to register the paper data (other authors, abstract, etc.) to finally, send it. Thus, we have decomposed `Submit Paper` into four Basic Use Cases: `Login`, `Register Paper Data`, `Show Paper Data` and finally `Send Paper`.

Then we have to identify each Basic Use Case as Functional or Structural as well as the Include and Extend relationships between them. `Send Paper` and `Register Paper Data` are Functional Basic Use Cases because they are stand-alone services that represent an interaction with the user. `Show Paper Data` is a Structural Basic Use Case associated with the Use Case `Register Paper Data` by means of an Extend relationship. `Login` is a Functional Basic Use Case associated with the `Send Paper` business service. `Login` is also associated with another two Functional Basic Use Cases by means of Extend relationships: `Register as New Author` and `Forgot Password`. Finally, we use an Activity Diagram to describe the business process associated to each business service in order to properly execute it (for the sake of space we skip it here).

To complete this stage on the WebConference WIS development, we would have to apply the same process we have applied to the `Submit Paper` business service, to `View Own Papers` and `Edit User Data` business services.

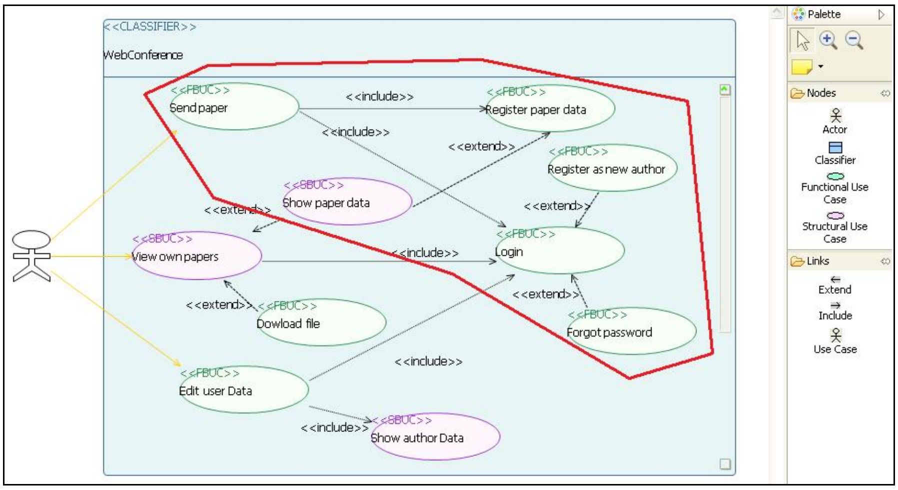

Figure 9: Extended use case model

5.3 Generating the Extended Slice Model

As we have already mentioned, the Extended Slice model is a model that represents the navigation map, without including the navigation structures (such as indexes, menus, etc.). The corresponding model for the Case Study is shown in [Fig. 10].

First, note that each business service gives rise to a Route. In our case we identify three routes: SP, VP and EU related to the `Submit Paper`, `View Own Papers` and `Edit User Data` business services, respectively. Besides, each fork from the corresponding activity diagram gives rise to a sub-route. Thus, the Submit Paper route includes two sub-routes: SP.FP {Submit Paper – Forgot Password} and SP.RA {Submit Paper – Register Author}. Finally, each Basic Use Case (whether Functional or Structural) is mapped into a Slice whose attributes are taken from the corresponding Class in the conceptual Data model. For instance, the `Show Paper Data` slice takes its attributes from the `Paper` class.

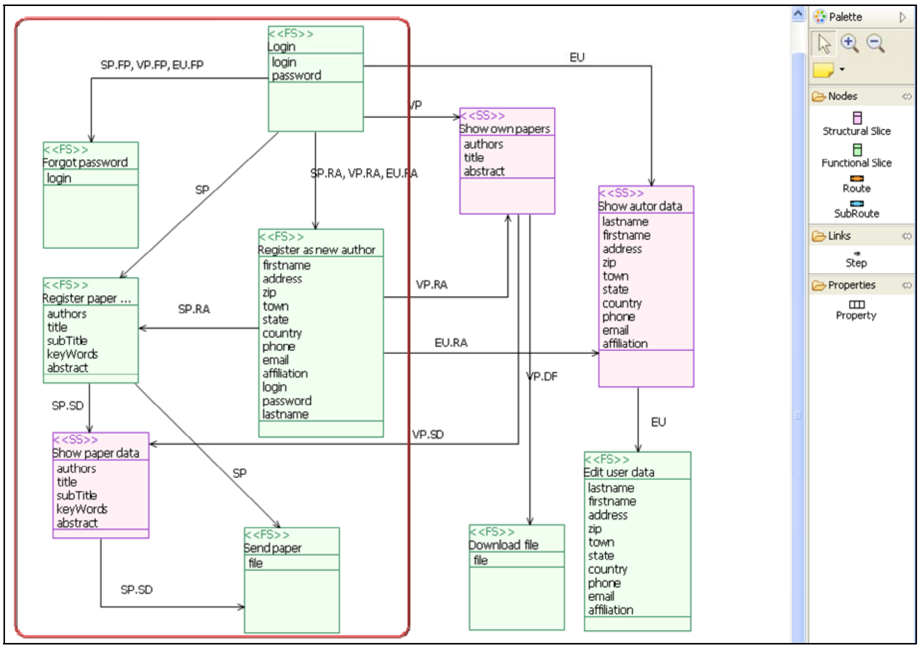

Figure 10: Extended Slice model

5.4 Generating the Extended Navigation Model

The Extended Navigation model ([Fig. 11]) is an Extended Slice model that includes the navigational structures *(menu, index, query and guidetour).* First, we propose introducing a *main menu with an entry for each service* required by the user, representing the beginning of each route.

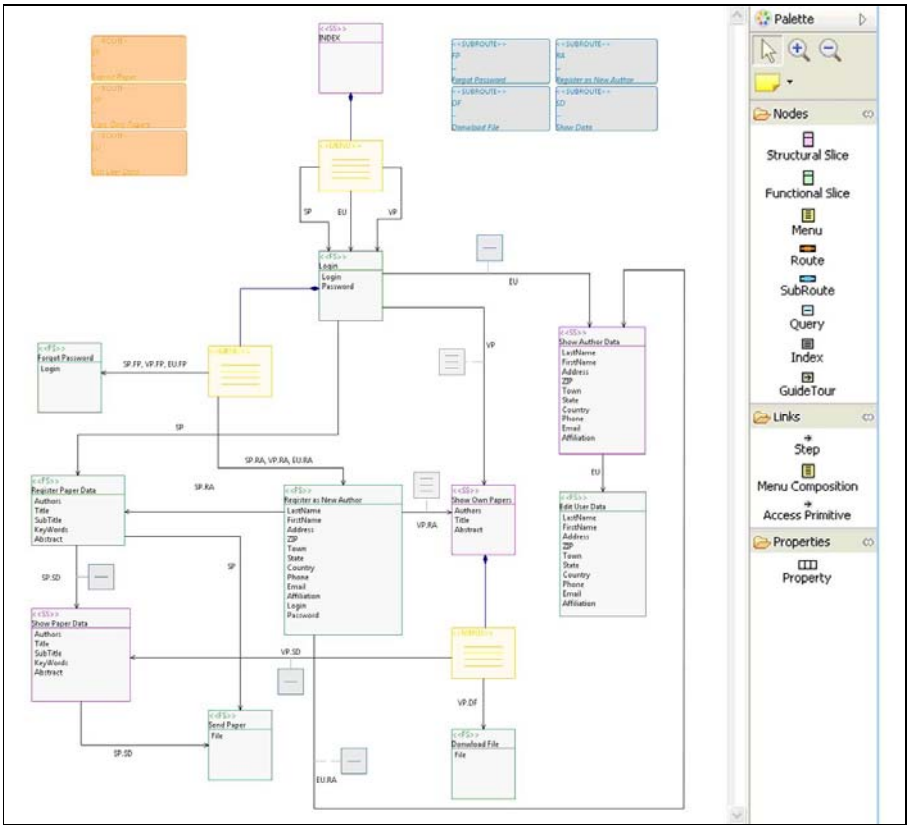

Figure 11: Extended Navigation model

As shown in [Fig. 11], we include a main menu with an entry for each service required by the user. In this case, the menu owns three entries: `Submit Paper`, `View Own Paper` and `Edit User Data`. Besides, additional menus are included for each sub-route. For instance, a menu with two entries is included after `Login`, one for each alternative sub-route: `Forgot Password` and `Register as New Author`. If this new menu comprises a single entry, it can be omitted. Finally, indexes, queries and guided tours are added according to the guidelines proposed in [Hennicker and Koch, 00]. We propose introducing one index or query for each Structural Slice included in the Extended Navigation model.

5.5 Generating the Web Site

Finally, the previous models are taken as input by a set of Xpand [Klatt, 08] templates in order to generate the working code of the Web Information System. [Fig. 12] shows the navigation through the Web pages that implement the *Edit User Data* business service.

While the main menu displays the entrance to each one of the three business services provided by the WIS, note that the route for the business service is shown at any moment. Indeed, the upper part of all the slices that compose the route of a given business service displays the path as well as the current state, that is, which is the current step among those composing the route.

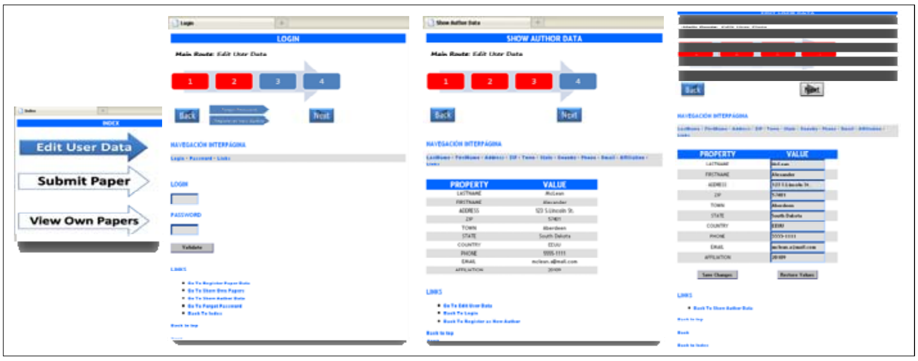

Figure 12: Navigation through the Edit User Data business service

6 Conclusions and Future Work

In this work we have presented a proposal for service-oriented development of WIS. To that end, we have introduced a modelling process to obtain the navigational model of the WIS and the MDA tool that supports it. In contrast with existing proposals that follow a traditional approach to build the navigation model, we propose to address the problem from a service-oriented perspective. The main benefit of our approach is that it allows obtaining a navigation model that includes certain characteristics that ease the navigation through the WIS for users, helping them to perform the needed tasks.

To illustrate the proposed modelling process we showed its application to the development of a conference management system. This case study was used in a first validation stage to test the applicability of the method. Then, we study in the same case study the benefits in terms of navigability of our proposal. Thus we carried out an empirical study that compared the navigability of two different WISs for conference management what showed that users preferred the WIS built with our method and indeed they perceived it to be more easily navigable.

We are already working to integrate this work with the rest of models proposed by MIDAS, like the ones for modelling high-level business processes. This way, we aim at providing some guidelines to facilitate the alignment between high-level business processes and the SOA architectures that implement them.

Acknowledgements

This research is partially funded by the MASAI project financed by the Spanish Ministry of Science and Technology (Ref. TIN-2011-22617) and Agreement Technologies (CONSOLIDER CSD2007-0022).

References

[Baresi et al., 00] Baresi, L., Garzotto, F., Paolini, P.: "From Web Sites to Web Applications: New Issues for Conceptual Modeling"; Proc. International Workshop on the World Wide Web and Conceptual Modeling, Salt Lake City, Utah, USA (2000), 89-100.

[Bergholtz et al., 10] Bergholtz, M., Andersson, B., Johannesson, P.: "Abstraction, Restriction, and Co-creation: Three Perspectives on Services": Proc. of ER Workshops, Vancouver, BC, Canada (2010) 107-116.

[Bézivin, 04] Bézivin, J.: "In search of a Basic Principle for Model Driven Engineering"; Novatica/Upgrade, 2, (2004) 21-24.

[Brambilla et al., 03] Brambilla, M., Ceri, S., Comai, S., Fraternali, P., Manolescu, I.: "Model-driven Development of Web Services and Hypertext Applications"; Proc. the 7th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, Florida, USA (2003), 37-42.

[Brambilla et al., 06] Brambilla, M., Ceri, S., Fraternali, P., Manolescu, I.: "Process Modeling in Web Applications"; ACM Transaction on Software Engineering and Methodology 15, 4 (2006) 360–409.

[Budinsky et al., 08] Budinsky, F., Merks, E., Steinber, D.: "Eclipse Modelling Framework 2.0, 2º Ed."; Addison-Wesley Professional (2008).

[Cáceres et al., 06] Cáceres, P., de Castro, V., Vara, J.M., Marcos, E.: "Model Transformations for Hypertext Modeling on Web Information Systems"; Proc. of the ACM Symposium on Applied Computing, Vol. 2, (2006) 1232-1239.

[Ceri et al., 00] Ceri, S., Fraternali, P., Bongio, A.: "Web Modeling Language (WebML): a modeling language for designing Web sites"; Computer Networks 33, 1,6 (2000) 137-157.

[Comai and Toffetti Carughi, 07] Comai, S., Toffetti Carughi, G.: "A Behavioral Model for Rich Internet Applications"; Proc. 7th International Conference on Web Engineering, LNCS 4607, Como, Italy (2007) 364-369.

[De Castro et al., 04] De Castro, V., Marcos, E., Cáceres, P.: "A User Service Oriented Method to model Web Information Systems"; Proc. of the Web Information Systems Engineering, LNCS 3306, Brisbane, Australia (2004) 41-52.

[De Castro et al., 09] De Castro, V., Marcos, E., Wieringa, R.: "Towards a Service-oriented MDA-Based Approach to the Alignment of Business Processes with it Systems: From the Business Model to a Web Service Composition Model"; Int. J. on Cooperative Systems 18, 2 (2009) 225-260.

[De Castro et al., 11] De Castro, V., Genero, M., Marcos, E., Piattini, M.: "Empirical study to assess whether the use of routes facilitates the navigability of web information systems"; IET Software, 5, 6 (2011) 528-542.

[De Troyer and Leune, 00] De Troyer, O., Leune, C.J.: "WSDM: A User Centered Design Method for Web Sites"; Computer Networks and ISDN Systems, 30,1,7 (2000) 85-94.

[Díaz and Aedo, 07] Díaz, P., Aedo, I.: "Towards efficient web engineering approaches through flexible process models"; Journal of Systems and Software 80,8 (2007) 1375–1389.

[Didonet Del Fabro et al., 06] Didonet Del Fabro, M., Bézivin, J., Valduriez P.: "Weaving Models with the Eclipse AMW plugin"; Eclipse Modelling Symposium, E. Summit Europe, Esslingen, Germany (2006).

[Escalona et al., 07] Escalona, M.J., Torres, J., Mejías, M., Gutiérrez, J.J., Villadiego, D.: "The treatment of navigation in web engineering"; Advances in Engineering Software 38,4, (2007) 267–282.

[Fraternali, 99] Fraternali, P.: "Tools and approaches for developing data-intensive Web applications: A survey"; ACM Computing Surveys, 31, 3 (1999) 227-263.

[Garzotto et al., 93] Garzotto, F., Paolini, P., Schwabe, D.: "HDM - a Model-Based Approach to Hypertext Application Design"; ACM Transactions on Database Systems, 11, 1 (1993) 1-26.

[Ginige and Murugesan, 01] Ginige, S. and Murugesan, A.: "Web Engineering: An Introduction"; IEEE MultiMedia, 8, 1 (2001) 4 -18.

[Gómez et al., 01] Gómez, J., Cachero, C., Pastor, O.: "Conceptual Modeling of Device-Independent Web Applications"; IEEE Multimedia, 8, 2 (2001) 26-39.

[Harmon, 04] Harmon, P.: "The OMGs Model Driven Architecture and BPM"; Newsletter of Business Process Trends. Available at: http://www.bptrends.com/publications.cfm (2004).

[Hennicker and Koch, 00] Hennicker, R. and Koch, N.: "A UML-based Methodology for Hypermedia Design"; Proc. UML Conference, LNCS 1939, York, UK (2000) 410-424.

[IBM] IBM Research, Service Science, Management and Engineering. Services Definition, Available at: http://researchweb.watson.ibm.com/ssme/services.shtml (2008)

[Isakowitz et al., 95] Isakowitz, T., Stohr, E.A.S., Balasubramanian, P.: "RMM: A Methodology for Structured Hypermedia Design": Communications of the ACM, 58, 8 (1995) 34-43.

[Jouault and Kurtev, 05] Jouault F. and Kurtev, I.: "Transforming Models with ATL"; Proc. of the Model Transformations in Practice Workshop at MoDELS Conference, Jamaica (2005), 128-138.

[Klatt, 08] Klatt, B.: "Xpand: A Closer Look at the model2text Transformation Language"; Available at: http://www.bar54.de/publikationen.html (2008).

[Koch et al., 04] Koch, N., Kraus, A., Cachero, C., Meliá, S.: "Integration of Business Processes in Web Application Models"; Journal of Web Engineering 3,1 (2004) 22-49.

[Koch et al., 09] Koch, N., Pigerl, M., Zhang G. and Morozova, T.: "Patterns for the Model-based Development of RIAs"; Proc. 9th International Conference on Web Engineering, LNCS 5648, San Sebastián, Spain (2009), 283-291.

[Ma et al., 09] Ma, H., Schewe, K.-D., Thalheim, B., Wang, Q.: "A Theory of Data-Intensive Software Services"; Service Oriented Computing and Its Applications, 3, 4 (2009) 263-283.

[Mecca et al., 99] Mecca, G., Merialdo, P., Atzeni, P., Crescenzi, V.: "The ARANEUS guide to Web-site development"; Available at: http://poincare.inf.uniroma3.it/ (1999).

[Mernik et al., 05] Mernik, M., Heering, J., Sloane, A.: "When and How to Develop Domain-Specific Languages"; ACM Computing Surveys, 37, 4, (2005) 316-344.

[Miller and Mukerji, 03] Miller, J., Mukerji, J. (eds.): "MDA Guide Version 1.0.1". Document number omg/2003-06-01, Available at: http://www.omg.com/mda (2003).

[OMG] OMG: "UML Superstructure 2.0", Draft adopted Specification, Available at: http://www.uml.org/.

[Papazoglou and Georgakopoulos, 03] Papazoglou, M.P., Georgakopoulos, D.: "Serviced-Oriented Computing"; Communications of ACM, 46,10 (2003) 25-28.

[Rossi et al., 08] Rossi, G., Urbieta, M., Ginzburg, J., Distante, D., Garrido, A.: "Refactoring to Rich Internet Applications. A Model-Driven Approach"; Proc. of the 8º International Conference on Web Engineering, Yorktown Heights, New York, USA (2008) 1-12.

[Schewe and Thalheim, 05] Schewe, K.-D., Thalheim, B.: "Conceptual Modelling of Web Information Systems"; Data & Knowledge Engineering, 54, 2 (2005) 147-188.

[Schewe and Thalheim, 05b] Schewe, K.-D., Thalheim, B.: "The Co-Design Approach to Web Information Systems Development"; International Journal on Web Information Systems, 1, 1 (2005) 5-14.

[Schmid and Rossi, 04] Schmid, D., Rossi. G.: "Modeling and Designing Processes in E-Commerce Applications"; IEEE Internet Computing 8, 1 (2004) 19-27.

[Schmidt, 06] Schmidt, D.: "Model-driven engineering"; IEEE Computer 39, 2 (2006) 25-31.

[Schwabe et al., 01] Schwabe, D., Esmeraldo, L., Rossi, G., Lyardet, F.: "Engineering Web Applications for Reuse"; IEEE MultiMedia 8, 1 (2001) 20-31.

[Valderas et al., 05] Valderas, P., Fons, J., Pelechano, V.: "From Web Requirements to Navigational Design - A Transformational Approach"; Proc. of International Conference of Web Engineering, LNCS 3579, Sydney, Australia (2005) 506-511.

[Vara et al., 09] Vara, J. M., Vela, B., Bollati, V., Marcos, E.: "Supporting Model-Driven Development of Object-Relational Database Schemas: a Case Study"; Proc of the International Conference on Model Transformation, Zurich, Switzerland (2009) 181-196.

[Vara, 09] Vara, J.M.: "M2DAT: a Technical Solution for Model-Driven Development of Web Information Systems". PhD Thesis. University Rey Juan Carlos. 2009. Available at: http://hdl.handle.net/10115/5145 (2009).

[Watson, 08] Watson, A.:" A, Brief History of MDA"; Upgrade, The European Journal for the Informatics Profesional IX, 2 (2008) 7-11.

[WebConference] WebConference. URL: http://kybele.escet.urjc.es/webconference-app/ (2005)