MODEL-DRIVEN NAVIGATION DESIGN FOR SEMANTIC WEB APPLICATIONS WITH THE UML-GUIDE

PETER DOLOG
L3S Research Center, University of Hannover,
Expo Plaza 1, 30539 Hannover, Germany,
E-mail: dolog@l3s.de

In this paper, we describe an extension to the UML-Guide for model driven navigation design of Semantic Web applications. The UML-Guide is used to specify platform independent navigation guides in web applications. We describe an OWL model for state machines which serves as a metamodel for semantic web descriptions of the navigation guides on the Semantic Web. Following the MDA approach, a state machine model of such navigation guide is generated from the UML state diagrams. The possible applications of such generated state machines are also discussed.

1 Introduction

The Semantic Web is treated as a next generation of the Web where "a well-defined meaning is given to information provided, better enabling computers and people to work in cooperation" [1]. Several technologies have been developed for shaping and describing the Semantic Web like RDF/S[a] and its extensions like OWL[b] to define metadata schemas, domain ontologies and resource descriptions. Other technologies are provided for describing computing resources for accessing and processing the information (resources, metadata) on the Semantic Web like OWL-S[c].

Recently, a big attention has been paid to develop a Semantic Web for information from user point of view; e.g. describing Semantic Web of web pages, learning objects, medical information and so on. To benefit from such information infrastructure, applications have to be developed to provide access to, to process, and to handle information on the (parts of) Semantic Web. The applications which process certain information on the Semantic Web can be considered as components which might be suitable for reuse in other applications. They are usually described in different design views such as those in the Unified Modelling Language (UML).

Several upper-level ontologies have been developed in the area of computer science and software engineering to describe such design views (e.g. entity-relationship model [2] for static aspects or various flavors of state machines like the state charts [3] for dynamic aspects). Such ontologies are used as web application design languages in methodologies like OOHDM [4], W2000 [5], UWE [6], WebML [7], UML extension for web engineering [8], and DEAHE [9,10]. The designs modelled in those languages describe the software artifacts used in web applications; i.e. they can be seen as semantic descriptions of the artifacts. To establish the semantic web of such arti-

---

[a] `http://www.w3.org/RDF/`
[b] `http://www.w3.org/2001/sw/WebOnt/`
[c] `http://www.daml.org/services/owl-s/1.0/`

facts, the semantic web representation of the upper-level ontologies used to describe their designs has to be developed. The transformations from those languages to their semantic web representation have to be studied as well. Such descriptions may contribute to better searchability of computation units (components) on the web and as such contribute to their better (re-)use.

Object Management Group (OMG)[d] has introduced the Model Driven Architecture (MDA) [11] as a promising approach to deal with transformations from platform independent models to platform specific implementations of applications. In this paper, we report on the UML-Guide [12] method extended for the Semantic Web applications following the MDA principle. We use the UML-Guide to specify platform independent guides (browsing trails) through information. We have developed the Semantic Web representation of the state machines as the platform specific model suited for the Semantic Web. We take advantage of the XML serialization of both models. The UML models are represented in the OMG XML Metadata Interchange (XMI). The W3C Resource Description Format (RDF) and Web Ontology Language (OWL) provide their XML serializations as well. Therefore, the W3C eXtensible Stylesheet Language Transformations (XSLT) can be employed to generate the Semantic Web representations of the state machine out of the XMI. Such an approach provides us with following advantages:

- A designer benefits from broadly adopted UML state machines used in many areas of software and web engineering;

- Semantic Web applications benefit from domain specific interpreters and reasoners of state machines which might be introduced independently of our approach as the operational semantics of the state machines is one of the standard models of computation;

- Model driven architecture provides us with a standard way to deal with transformations from domain independent to implementation specific Semantic Web models.

The rest of the paper is structured as follows. Section 2 illustrates a scenario about application of the OWL state machines in a distributed web service based environment. An overview of the UML-Guide can be found in sec. 3. A Semantic Web metamodel for the state machines is discussed in sec. 4. MDA view on a transformation from the UML state diagrams to the OWL based state machines is provided in sec. 5. Section 6 discusses possible usage scenarios of the approach. A discussion on related work is provided in sec. 7. The paper concludes with summary and some remarks on further work (sec. 8).

[d] `http://www.omg.org`

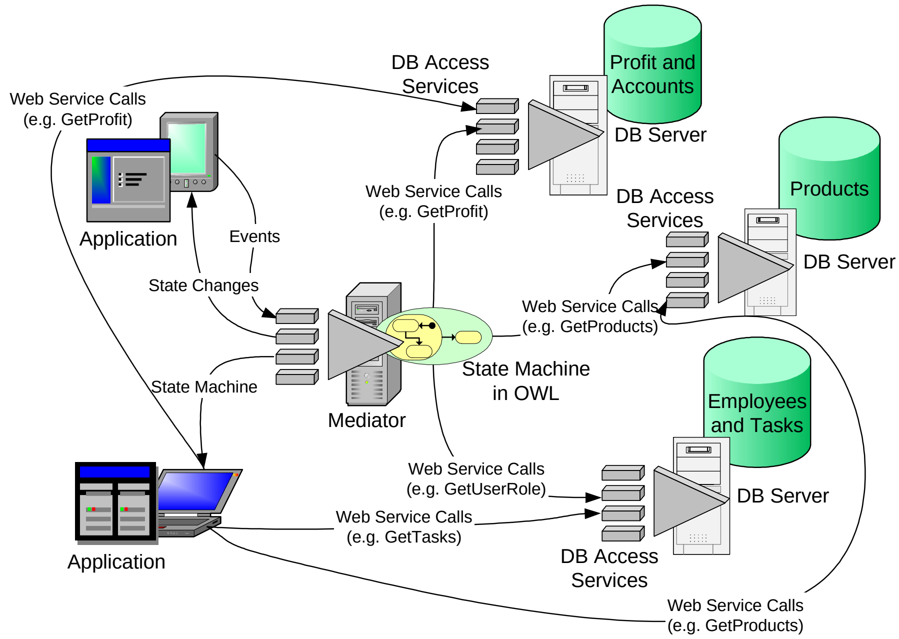


Figure 1. An example architecture that shows client applications interpreting a state machine which mediates an access to information.

2 Sample Scenario

To motivate our approach we refer to an example scenario depicted in Fig. 1. The state machine represented in OWL is interpreted by client applications. Some applications (as in case of applications at the PDA) may be able to deal just with current state and events. Those applications forward events raised by user interaction to the mediator and accept the state changes. Mediator is responsible for processing or forwarding the operation calls to other web service providers.

Another kind of application may be able to interpret the whole state machine and to handle web service calls at their site (as in the case of applications at the notebook). Those applications download a state machine from the mediator and then interact directly with the remote web services. The services can in addition customize information for example according to a user role within a company or device used (e.g. PDA, Laptop, Tablet PC).

Different states represent different chunks of information on a user trail which might reside on different sites. The sites provide access to their databases through defined interfaces provided as web services. In our case we refer to accounting, products and employees with their tasks databases as they are related to our application scenario for the UML-Guide specification later in this paper. The mediator and client applications follow the side effect actions of user triggered events and transform operation calls used in those actions into web service calls.

Such semantic model of state machine (or computation of user trails) can be exposed further to Semantic Web search or web service registries. The semantic model provides an information about higher level interactive services which might be used for search criteria. Web application designer can thus better shape his queries to find particular component needed for his web application.

In the following we will concentrate on how we adapted the UML-Guide method to design navigation trails which can be transformed into owl representation of the state machine.

3 UML-Guide Overview

UML State diagrams are used in UML-Guide for modelling the user navigation in a hypertext. Each *state* represents the production of a given information chunk on the device observed by a user, and each state *transition* represents an event caused by user interaction that leads to the production of a new chunk of information. State diagrams therefore provide an abstraction of *hypertext trails*, where each trail can be adapted by taking into account the user background, level of knowledge, preferences and so on [12].

Atomic states can be grouped into *superstates*. States usually refer to concepts of an application domain. *Parallel substates* represent information chunks to be presented simultaneously. *Fork* and *join* pseudostates are used respectively for splitting and joining computations and enabling parallelism. The *SyncState* pseudostate is used for synchronizing substates of parallel regions.

Transitions represent active interconnections between information chunks. *Events* raise transitions in a state machine; they include user-generated or system-generated events, and the latter include time events. *Guards* can be used to constrain transitions by adaptation rules. Usually, they consist of a predicate over user profile attributes or context information.

Actions can be assigned to transitions and states. The side effect actions of a transition are performed after the transition is raised and before entering a state. Such side effect actions can for example modify a user profile, or the choice of presentation styles for a given chunk of information. Actions can also process parameters used in transition guards. Side effect actions, as well as adaptation rules, can be assigned to *entry*, *exit*, and *do* actions of states.

Tagged values are domain-specific properties used to extend the semantics of elements in UML diagrams. These values can refer, for example, to concepts of the structural model of the application domain, or to specific terminologies which might be useful to identify additional navigation requirements.

Figure 2 depicts an excerpt of state diagram representing a navigation model for account managers and product managers of a CRM application. The diagram represents one use case: browsing daily assets. Each day, an account manager looks for accounts to contact in order to provide them a support. The application provides

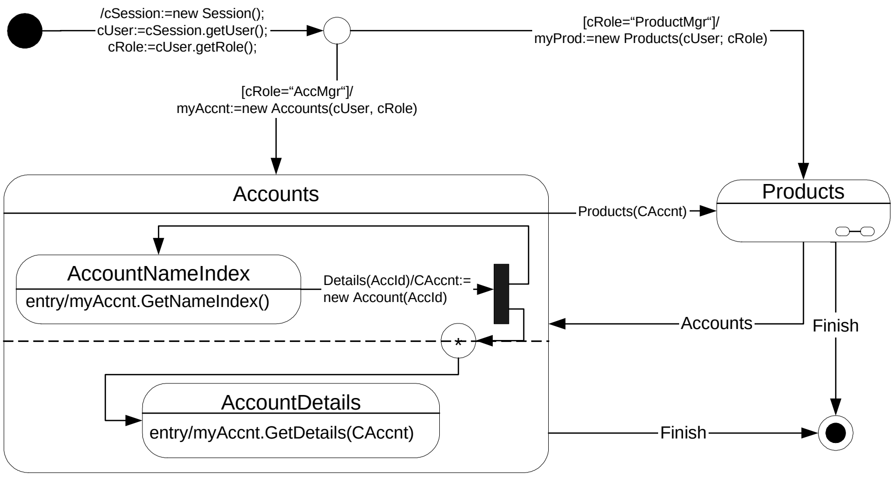


Figure 2. An excerpt of navigation model for browsing assets in a CRM application.

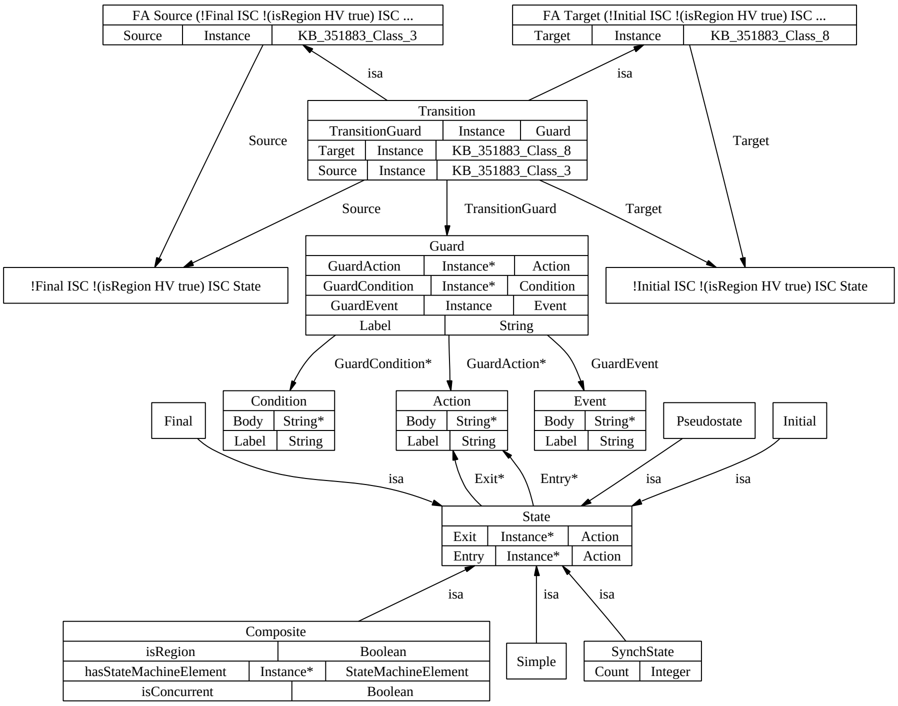
a view on a list of the accounts (the `AccountNameIndex` state). He can browse into account details if he needs them in communication with account representatives (the `AccountDetails` state). As the details should be displayed simultaneously with the list of accounts, two concurrent regions are depicted in the model. They are synchronized on click based on an account identifier taken from the record which was clicked (the transition from the `AccountNameIndex` state raised by the `Details(AccId)` event). To better support a customer, he can browse details about products soled, problems reported and so on (the `Products` state collapsed due to space limitations). Product managers start with products they have to support with summary of accounts which have bought the products.

4 Semantic Web Model for State Machines

Semantic Web meta model for state machines can be described in RDFS or OWL. RDFS is sufficient when we need to express classes and relations. The model defines a language for expressing concrete state machines. OWL also provides us with possibility to express constrains posed on those language elements.

We have developed an OWL ontology for state machines. Figure 3 shows an excerpt of the Semantic Web state machine metamodel[e] visualized using OntoViz plugin for protege[f]. State can be `Composite`, `Simple`, `Initial`, `Final`, and `Pseudostate` similarly to the metamodel of the UML. Each state can have `Entry` and `Exit` actions. The fact that composite state is a container for other states is

[e] for details visit `http://www.l3s.de/~dolog/fsm/`
[f] `http://protege.stanford.edu/`



*	…	Multiple instances allowed	ISC	…	IntersectionOf
!	…	ComplementOf	FA	…	Forall
HV	…	hasValue			

Figure 3. An excerpt of OWL model of transition in a state machine.

reflected by a property which refers to multiple `StateMachineElement`-s. The composite state can be a region or concurrent. This fact is represented by two boolean properties `isConcurrent` and `isRegion`. `SynchState` has an integer parameter which reffers to a difference between number of times an incomming and outgoing transition from the `SynchState` is fired.

`Transition`-s are connected to their states through their `Target` and `Source` properties. However, there are restrictions which restrict values of the source and target properties. A target of transition is a `State` different from `Initial`. A source of transition is a `State` different from `Final`. Transitions cannot connect regions. The restrictions are rendered as rectangles in fig. 3 as they generate new anonymous classes different from those modelled explicitly. Each transition can have a `Guard` which has an `Event`, `Condition`, and `Action`.

The OWL provides its RDF/XML bindings as an exchange format. Figure 4 depicts an excerpt of the XML serialization of the OWL model defining the `Transition` class. The description shows a restricted `Target` property which

```
<owl:Class rdf:ID="Transition">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:allValuesFrom>
          <owl:Class>
            <owl:intersectionOf rdf:parseType="Collection">
              <owl:Class>
                <owl:complementOf>
                  <owl:Class rdf:about="#Initial"/>
                </owl:complementOf>
              </owl:Class>
              <owl:Class>
                <owl:complementOf>
                  <owl:Restriction>
                    <owl:onProperty>
                      <owl:DatatypeProperty rdf:about=
                      "#isRegion"/>
                    </owl:onProperty>
                    <owl:hasValue
                    rdf:datatype="XMLSchema:boolean">
                    true
                    </owl:hasValue>
                  </owl:Restriction>
                </owl:complementOf>
              </owl:Class>
              <owl:Class rdf:about="#State"/>
            </owl:intersectionOf>
          </owl:Class>
        </owl:allValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#Target"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    ...
</owl:Class>
```

Figure 4. An excerpt of RDF/XML bindings of the OWL model for the Transition class.

can hold the instances of the `State` class different from instances of the `Initial` class and the instances having the `isRegion` property true.

5 MDA view on transformation in the UML-Guide

The main idea of MDA is based on a well-known principle of separating the system operation specification from the details of system implementation [11]. The system

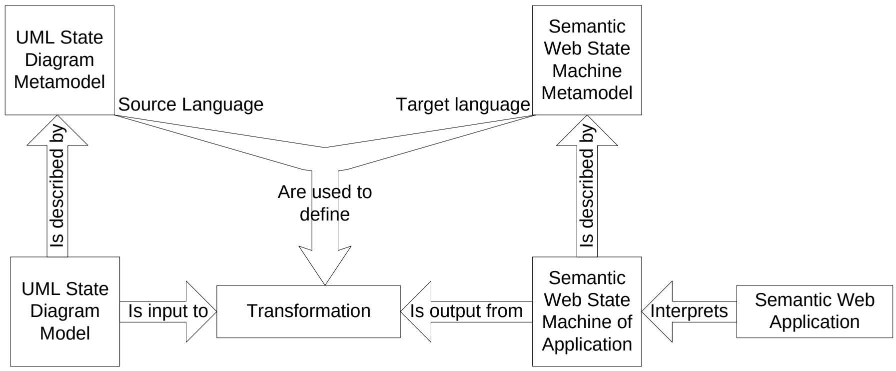


Figure 5. A MDA view on transformation from the UML state diagrams to Semantic Web state machines.

operation is specified as a platform independent model. The platform independent model is transformed to a model suited for particular platform utilizing a transformation which adds platform specific extensions based on the platform model.

The UML-Guide approach relies on the UML state machines as the platform independent model for navigation trails in web applications. The platform specific model is presented as the OWL Semantic Web metamodel for state machines. The transformation defines mappings from the UML state machine metamodel elements to the state machine ontology classes. A diagram showing the process of transformation from the UML state diagrams (source language) to the Semantic Web state machines (target language) is depicted in figure 5. We use XML serialization in both cases — XMI in case of the UML and XML serialization of OWL — thus the transformation can be defined in XSLT.

For the implementation experiments we used commercial tool Poseidon[g] to produce XMI representation of the UML-Guide navigation models. The instance of state machine described in the OWL is then generated through a transformation method, whose input is the state diagram encoded in XMI as produced by Poseidon.

Figure 6 depicts an excerpt of the generated OWL instance from the UML state diagram. The `Account` composite state has two parallel regions each serving as a container for `AccountNameIndex` and `AccountDetails`. The two simple states point to the entry actions used to instantiate content. Other elements from figure 2 are transformed similarly.

6 Discussion

The generated Semantic Web annotations may serve as an input for already existing tools used to interpret and reason over the OWL or RDF based annotations. The

[g] `http://www.gentleware.com/`

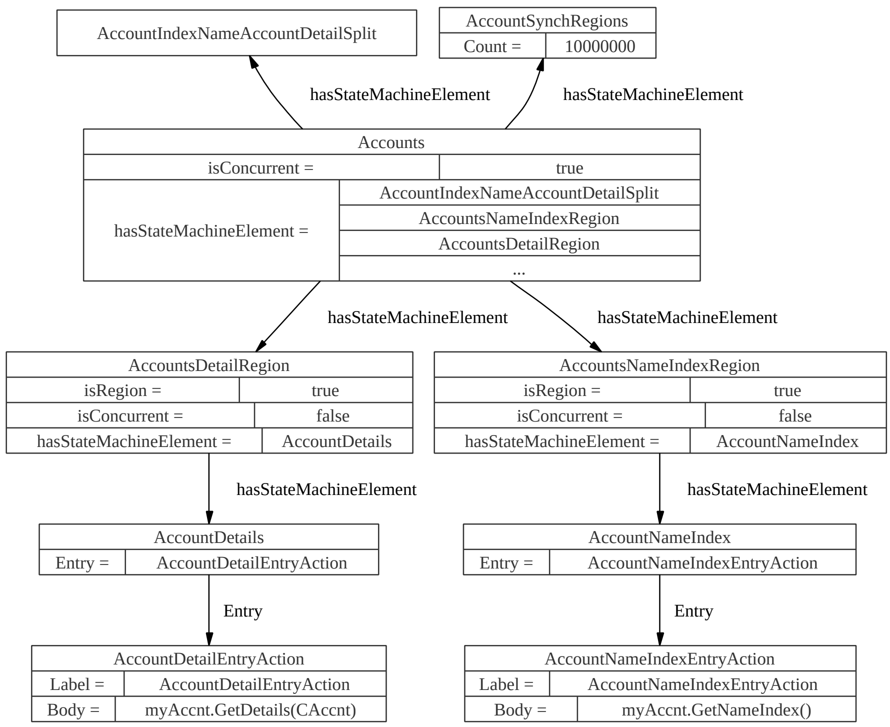


Figure 6. An OWL instance of state machine generated from the UML diagram.

model can be used in protege[h] to derive additional information by using reasoners connected with the OWL plugins. The Semantic Web model of state machines can be used in Semantic Web based eLearning systems similarly as we used UML-Guide method for modelling specific personalized eLearning curricula [13]. The Personal Reader [14] can take advantage from particular guides through additional resources generated from Semantic Web. The TRIPLE [15], a query and transformation language for the Semantic Web, can be used to reason over the state machine model and to provide additional transformations for example in the context of the Personal Learning Assistant [16] where it can be used to specify a learning path. The TRIPLE views can be used also as information sources for content being displayed in particular state when user enters the state. The Semantic Web models can also be exposed in the Edutella [17] P2P infrastructure for sharing resources on the Semantic Web.

[h] http://protege.stanford.edu/

7 Related Work

Besides classical model-driven development of Web applications [18,19,20,4], we would like to point especially to those which already considered Semantic Web oriented extensions. OOHDM was discussed in the light of the Semantic Web description formats. OOHDM primitives for conceptual and navigation models have been described as DAML classes and RQL queries [21]. The RDF and RDFS was employed in an extensible web modelling framework [22]. The RDFS is used to define web schemata as vocabulary for web application descriptions. Such descriptions are then converted into web application implementation related technologies. The HERA [23] methodology is an RMM [24] based methodology extended for adaptive web-based application engineering. It uses Semantic Web description formats for models representation. Our approach is based on transformation from platform independent UML models to Semantic Web specific formats to provide complemental operational view on Semantic Web application.

The UML-Guide was already used together with two platforms: web sites [12] and WebML platform [13] where a model driven approach has been followed.

8 Conclusions and Further Work

In this paper we described an OWL ontology for state machines we have developed. We described how to get the OWL based state machine annotations for Semantic Web applications using the UML-Guide method. Additional generator has been introduced in the UML-Guide to generate such Semantic Web state machines to provide additional view for interpreters and reasoners for Semantic Web.

In our further work we would like to concentrate on experiments with operations used as side effect actions in the state diagrams. The web service interfaces mentioned in the motivation scenario in sec. 2 might vary. We would like to investigate design approaches to handle those variabilities and how they influence generators. We also would like to incorporate Semantic Web oriented view to other models considered in DEAHE to provide further information about existing software components on the semantic web.

References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.
2. P. P. Chen. The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.
3. D. Harel. Statecharts: A visual formulation for complex systems. *Science of Computer Programming*, 8(3):231–274, 1987.
4. D. Schwabe and G. Rossi. An object-oriented approach to web-based application design. *Theory and Practise of Object Systems (TAPOS), Special Issue on the Internet*, 4(4):207–225, October 1998.

5. L. Baresi, F. Garzotto, and P. Paolini. Extending UML for modeling web applications. In *Proceedings of 34th Anual Hawaii International Conference on System Sciences (HICSS'34)*, Maui, Hawai, January 2001. IEEE Press.
6. R. Hennicker and N. Koch. A UML-based methodology for hypermedia design. In S. Stuart A. Evans and B. Selic, editors, *Proceedings of UML 2000 — The Unified Modeling Language. Model Languages and Applications*, volume 1939 of *LNCS*, York, England, October 2000. Springer.
7. S. Ceri, P. Fraternali, and M. Matera. Conceptual modeling of data-intensive web applications. *IEEE Internet Computing*, 6(4), August 2002.
8. D. Lowe, B. Henderson-Sellers, and A. Gu. Web extensions to uml: Using the MVC triad. In S. Spaccapietra, S.T. March, and Y. Kambayashi, editors, *Proceedings of 21nd International Conference on Conceptual Modeling*, pages 105–119, Tampere, Finland, October 2002. Springer, LNCS 2503.
9. P. Dolog and W. Nejdl. Using UML-based feature models and UML collaboration diagrams to information modelling for web-based applications. In *Proceedings of UML 2004 — The Unified Modeling Language. Model Languages and Applications. 7th International Conference*, LNCS. Springer, 2004.
10. P. Dolog and M. Bieliková. Towards variability modelling for reuse in hypermedia engineering. In Y. Manolopoulos and P. Návrat, editors, *Proceedings of ADBIS 2002 — Advances in Databases and Information Systems*, volume 2435 of *LNCS*, pages 388–400, Bratislava, Slovakia, September 2002. Springer.
11. OMG. MDA guide version 1.0.1, June 2003. Available at http://www.omg.org/. Accessed on June 1, 2003.
12. P. Dolog and W. Nejdl. Using UML and XMI for generating adaptive navigation sequences in web-based systems. In P. Stevens, J. Whittle, and G. Booch, editors, *Proceedings of UML 2003 — The Unified Modeling Language. Model Languages and Applications. 6th International Conference*, volume 2863 of *LNCS*, pages 205–219, San Francisco, CA, USA, October 2003. Springer.
13. S. Ceri, P. Dolog, M. Matera, and W. Nejdl. Model-driven design of web applications with client-side adaptation. In N. Koch, P. Fraternali, and M. Wirsing, editors, *ICWE 2004 - International Conference on Web Engineering*, volume 3140 of *LNCS*, pages 201–214, Munich, Germany, July 2004. Springer.
14. P. Dolog, N. Henze, W. Nejdl, and M. Sintek. The personal reader: Personalizing and enriching learning resource using semantic web technologies. In W. Nejdl and P. De Bra, editors, *Proceedings of AH2004 — International Conference on Adaptive Hypermedia*, volume 3137 of *LNCS*, Einghoven, The Netherlands, August 2004. Springer.
15. M. Sintek and S. Decker. TRIPLE—A query, inference, and transformation language for the semantic web. In I. Horrocks and J. A. Hendler, editors *Proceedings of ISWC 2002 — 1st International Semantic Web Conference*, volume 2342 of *LNCS*, Sardinia, Italy, June 2002. Springer.
16. P. Dolog, N. Henze, W. Nejdl, and M. Sintek. Personalization in distributed e-learning environments. In *Proceedings of WWW2004 — The Thirteen International World Wide Web Conference*, New Yourk, May 2004. ACM Press.

17. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: a P2P Networking Infrastructure based on RDF. In *Proceedings of 11th World Wide Web Conference*, Hawaii, USA, May 2002.
18. J. Conallen. *Building Web Applications with UML*. Addison Wesley, 2000.
19. P. Fraternali. Tools and approaches for developing data-intensive Web applications: A survey. *ACM Computing Surveys*, 31(3):227–263, September 1999.
20. F. Garzotto and P. Paolini. HDM — a model-based approach to hypertext application design. *ACM Transactions on Information Systems*, 11(1):1–26, January 1993.
21. F. Lima and D. Schwabe. Modeling applications for the semantic web. In J. M. C. Lovelle, B. M. G. Rodríguez, L. J. Aguilar, J. E. L. Gayo, and M. del P. P. Ruíz, editors, *Proceedings of ICWE 2004 - International Conference on Web Engineering*, volume 2722 of *LNCS*, pages 417–426, Oviedo, Spain, July 2003. Springer.
22. R. Klapsing, G. Neumann, and W. Conen. Semantics in web engineering: Applying the resource description framework. *IEEE MultiMedia*, April–June 2001.
23. R. Vdovjak, F. Frasincar, G.-J. Houben, and P. Barna. Engineering semantic web information systems in hera. *JWE — Journal on Web Engineering*, 2003.
24. T. Isakowitz, A. Kamis, and M. Koufaris. Extending the capabilities of RMM: Russians dolls and hypertext. In *Proceedings of the 30th Annual Hawaii International Conference on System Sciences*, January 1997.