A MDA-Based Approach for Web Information System Development

Paloma Cáceres, Esperanza Marcos, Belén Vela

Kybele Research Group
Rey Juan Carlos University
Madrid (Spain)
{p.caceres, e.marcos, b.vela}@escet.urjc.es

Abstract. Nowadays, new technologies and platforms are emerging and changing constantly, which implies a high effort developing software products. This situation generates different problems related to portability, integration and interoperability. The Object Management Group (OMG) proposes the Model Driven Architecture (MDA), which improves portability of applications by allowing the same model to be realized on multiple platforms. MDA defines an architecture based on platform independent models (PIM) and platform specific models (PSM). In this work we present MIDAS, a model-driven methodology based on MDA for the development of Web Information Systems (WIS). We apply the MDA metamodel to the Web platform using XML and (object-) relational technology. MIDAS proposes different PIMs and PSMs and defines some mapping rules between models. This paper also presents part of a case study to illustrate this proposal.

Key Words: MDA, Model-Driven Methodology, Mapping Rules, Web Information System Modeling

1 Introduction

New technologies are emerging and becoming popular constantly, for example Java, HTML, XML, J2EE, UML, etc. The enterprises usually develop their information systems according to these modern technologies. For this reason, new problems have appeared in relation to the interoperability, the portability and the integration of the information systems.

To help computer users to solve mentioned integration problems, the Object Management Group (OMG) has developed the Model-Driven Architecture (MDA) [12,13,15]. MDA is a model-driven framework for software development that proposes to specify the whole system with platform independent models (PIM) to be able to develop the software system for different platforms. MDA proposes a model classification, the relationship and the mapping rules between these models, as we will see next:

- **PIM – Platform Independent Model**
  A PIM is a model with a high level of abstraction, independent of any implementation technology.

- **PSM – Platform Specific Model**
  A PSM is a model for a specific platform. A PIM can be transformed into one or more PSMs, which are for a specific implementation technology.

MDA also provides mappings between models. A mapping is a set of rules and techniques used to modify one model in order to get another model. Mappings are used to transform:

- **PIM to PIM**
  This transformation is used to map the analysis to design models. Usually, models progress during the development lifecycle with independence of any platform.
- **PIM to PSM**
  This transformation is used when a PIM is refined to be deployed in a platform specific model.
- **PSM to PSM**
  This transformation usually is related to platform specific model refinement.
- **PSM to PIM**
  This transformation is required to obtain a platform independent model from models existing in a particular technology.

In this paper we present MIDAS [9], a model-driven methodology based on MDA for the development of Web Information Systems (WIS). We apply the MDA metamodel to the Web platform using XML [1,19,20] and (object-)relational [3] technology. MIDAS proposes different PIMs and PSMs and defines some mapping rules between models, as we will see next.
The rest of the paper is organized as follows: section 2 is an overview of the MIDAS methodology; in section 3, a part of a case study of a WIS for a cinema consortium of Madrid, using MIDAS, is presented; finally, section 4 sums up the main conclusions and future works.

2 MIDAS: A Model-Driven Methodology

MIDAS [9] is a model-driven methodology for WIS development based on MDA. We apply the MDA metamodel to the Web platform using XML and (object-)relational technology. This methodology proposes some PIMs, PSMs and mapping rules between the models. As MIDAS proposes to use a unique notation to model the whole system, both the PIMs and PSMs will be represented with UML. Unfortunately, UML has some lacks to represent the whole WIS. Recently, some UML extensions for Web modeling [2,5] have emerged. MIDAS proposes to use some of these extensions and it also defines some new ones when necessary, as for example the UML extension for modeling Web services [6,7]; for (object-)relational database design [8,11,18]; or the UML extension for representing XML Schemas [16] or XLink [17].
According to the proposal of [14], the requirements of the modeling of a WIS can be categorized by means of three orthogonal dimensions: *levels*, *aspects* and *phases*. The first dimension comprises three different *levels* namely the content, the hypertext and the presentation level. The second dimension comprises the different *phases* of a software life cycle, ranging from analysis to implementation. The third dimension comprises the *aspects* of structure and behavior. We will describe MIDAS according

to this classification of aspects, considering on the one hand the structural models and on the other hand the behavior models, see figure 1.

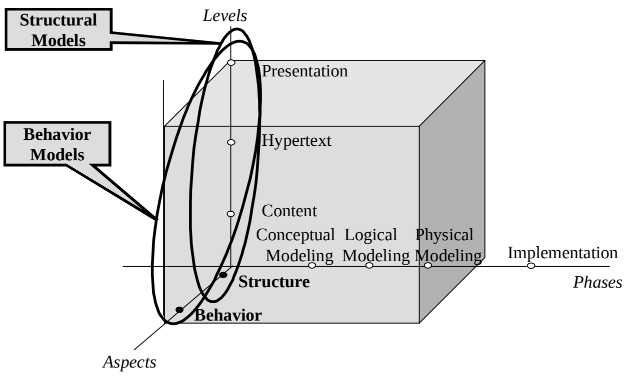

Fig. 1.- MIDAS structural and behavior dimension (based on [14])

MIDAS has a model-driven architecture. It has a system core, that represents the domain and business models. Over this central core, we define a ring which includes both the structural and the behavior dimension of the system. The core and this ring represent the technology and platform independent modeling. The external ring focuses on the different platforms and supported technologies. Based on this architecture (see figure 2), in the same way as in MDA, MIDAS defines a framework for the specification of information systems, which differentiate between the system functionality specification and the system functionality implementation specification [12].

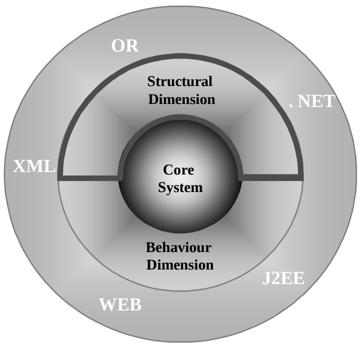

Fig. 2.- MIDAS architecture

MIDAS defines PIMs and PSMs according to the structural and behavior dimension. And also proposes some mapping rules between the different models, that is to say, between PIMs, between PSMs and between PIMs and PSMs.
In this work, we present a specific MDA application for the Web platform and for (object-)relational (OR) and XML technologies. We will focus on the structural

dimension of MIDAS, which is the marked part in figure 2, describing the specific PIMs and PSMs and the mappings between them.

2.1 Models of the Structural Dimension

The models of the structural dimension of MIDAS are divided into PIM and PSM, that is, platform independent and platform specific models:

- ***Structural* PIMs**
  The proposed **platform independent** models include the levels of content, hypertext and presentation at the conceptual modeling phase, as you can see in figure 1.
  At the content level, we propose to use the UML class diagram to carry out the data conceptual model. At the hypertext level, MIDAS proposes to realize the hypertext with two techniques: the slice and the navigation conceptual models proposed in RMM [4] using, respectively, the slice and navigation diagrams in extended UML. The proposed notation is based on the UML-based Web Engineering (UWE) approach [5]. At the presentation level we propose to use the presentation diagram defined in UWE [5] to carry out the presentation conceptual model.

- ***Structural* PSMs**
  The proposed **platform specific** models are based on (object-)relational and XML technology concerning the logical and physical modeling and implementation phases. (Object-)relational technology is used to represent the content level and XML technology to represent the hypertext and presentation levels.
  At the content level, the logical data model is carried out with the (object-) relational model according to the proposal defined in [11]. At the hypertext level, MIDAS proposes to realize the logical models of the hypertext (slice and navigation model). The slices are represented at a logical level with XML Schemas [19] in extended UML [16] and the slice and navigation logical models with XML Linking Language (XLink [20]) in extended UML [17]. The presentation logical model is deployed with XSL [21].

2.2 Mappings between the Models of the Structural Dimension

The MIDAS methodology also provides some mappings between the models described previously:

- **Mapping from PIM to PIM**
  MIDAS proposes guidelines based on RMM [4] to pass from the data conceptual model to the slice and navigation models. The methodology also provides some guidelines to obtain the presentation conceptual model from the data and hypertext conceptual models.

- **Mapping from PIM to PSM**
  The mapping rules from the structural PIM to the structural PSM start from the conceptual models (at the content, hypertext and presentation levels) into the logical models.
  The conceptual data model will be transformed into a data logical model following the transformation rules defined in [10,11,18] for (object-)relational database design. The conceptual hypertext models will be transformed into logical hypertext models using XML technology, XML Schemas [16] for the slices y XLink [17] for the slice and navigation diagrams, both in extended UML. The conceptual presentation models are mapped directly to a logical level by means of the definition of style sheets in XSL [21].

- **Mapping from PSM to PSM**
  The mapping rules between the structural PSMs allow making the transformation between different platform specific models, that is to say, between logical models, and between logical and implementation models.
  At the content level, MIDAS proposes some transformation guidelines to obtain the database implementation in the SQL of a specific product (for example, Oracle9*i*) [11]. At the hypertext and presentation levels, MIDAS provides some mapping rules to map the logical models into the implementation models. These rules describe how to obtain the final dynamic Web pages in XML, extracting the information from the database implemented at the content level. MIDAS provides also some guidelines to define how the integration from de XML pages with the database is made using some technology as ASP, JSP, etc.

3 A Case Study

In this section we are going to show part of a case study, which is a WIS for the consortium of several cinema chains of Madrid. This WIS can be visited at http://kybele.escet.urjc.es/ejemplos/cine_entradas/. It offers information about all the chains of the consortium and also a service to buy online cinema tickets. In this paper, we will focus on the structural dimension of the WIS.
As we have mentioned before, the models of the structural dimension of MIDAS are divided into PIM and PSM, that is, platform independent and platform specific models. At first, we will show the obtained conceptual models: data, slice and navigation conceptual models (Structural PIMs). And afterwards, we will show the logical data and hypertext models and their implementation (Structural PSMs).

- ***Structural* PIMs**

  The platform independent models of the case study, include the levels of content, and hypertext at the conceptual modeling phase. We have left out the presentation level for the sake of the brevity.

  At the **content level**, we propose to use the UML class diagram to carry out the data conceptual model. We will only present a reduced part of the data conceptual model. This partial data conceptual model, from which we start, includes several cinema chains, which can be composed of more than one cinema. In each of the cinemas there are played several movies and a movie can be played in more than one cinema. A movie can be shown in various sessions, but in a session only one movie can be played. A movie has only one producer who can produce more than one movie (see figure 3).

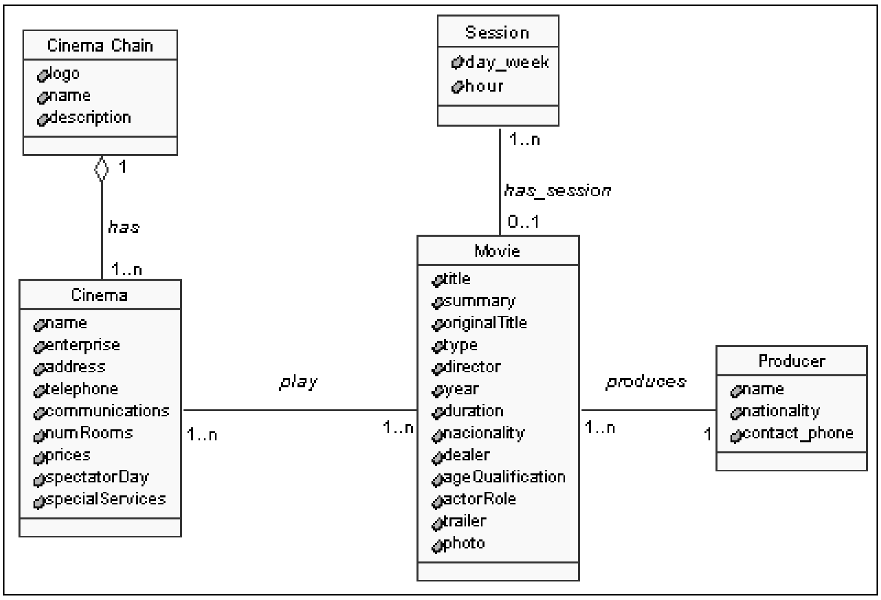

Fig. 3.- Partial Data Conceptual Model (PIM)

At the **hypertext level**, MIDAS proposes to realize the hypertext with two techniques: the slice and the navigation conceptual models. From the data conceptual model we obtain the slices and the slice diagram, which we represent in the extended UML notation based on the proposal of UWE, see figure 4. Each slice is represented with a navigational class named in the same way as the class from the data conceptual model from which the main information is obtained. The information presented in each of the slices corresponds with the information that would be shown in the pages and can belong to one or more classes of the data conceptual model. There can also be some classes of the data conceptual model that will not be included as a navigational class. In our data conceptual model exists a class Producer, which does not appear in the slice conceptual model. The only information of this class that will be shown is the name of the producer and it is included as an attribute producer in the navigational class Movie.

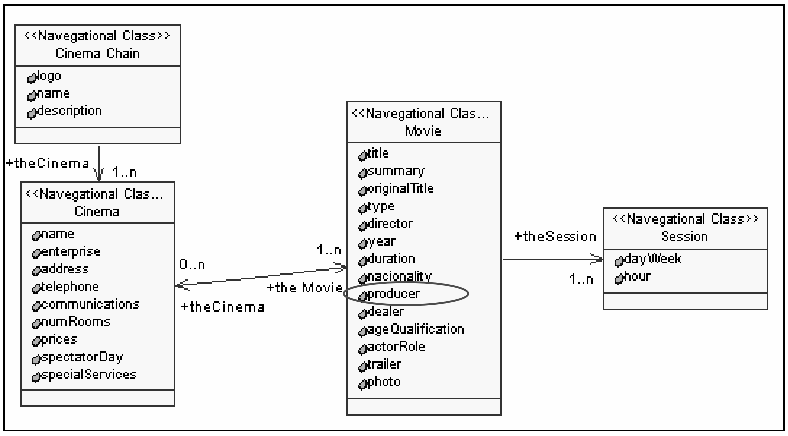

Fig. 4.- Partial Slice Conceptual Model (PIM)

In the *navigational conceptual model*, the access elements (indexes, guided tours, queries and menus) are added to the slice conceptual model. So in our case study, we include four indexes to navigate between the navigational classes: Cinema Chain, Cinema, Movie and Session (see figure 5).

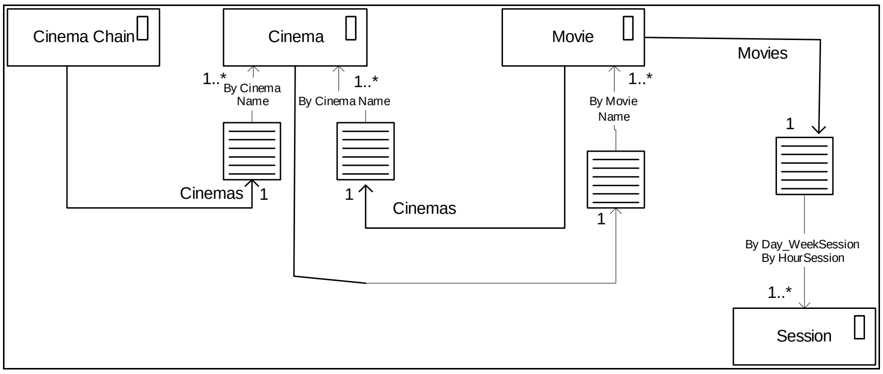

Fig. 5.- Partial Navigational Conceptual Model (PIM)

- ***Structural* PSMs**

The proposed platform specific models are based on (object-)relational and XML technology concerning the logical and physical modeling and implementation phases. (Object-)relational technology is used to represent the content level and XML technology to represent the hypertext level.

At the **content level**, in parallel to the hypertext design, the logical data model is carried out with the (object-)relational model; we will carry out the database design in the standard SQL:1999, applying the transformation rules defined in [11].

In this case of study, the database is independent from the design of the hypertext and may have other uses. Each of the classes of the refined conceptual model: Cinema Chain, Cinema, Movie, Producer and Session are transformed into object types. The relationships between the classes are transformed by means of a reference REF or an ARRAY of references to the class with which it is related. The type (REF or ARRAY) used to represent the relationship depends on the multiplicity of the relationship. So, for example the many-to-many relationship between the classes Cinema and Movie will be represented including in the object type Cinema an attribute play that is an ARRAY of references to the movies that are played in the cinema. To represent the two-way relationship, we include in the Movie object type also an attribute is_played, which is an ARRAY of references to the Cinemas, which play the movie. The user defined types will be represented with <<udt>> stereotyped classes, as for example, Type_Price and Type_Address. The graphical notation used to represent the database design is the extended UML [11]. Figure 6 shows the data logical design in UML represented for the standard SQL:1999.

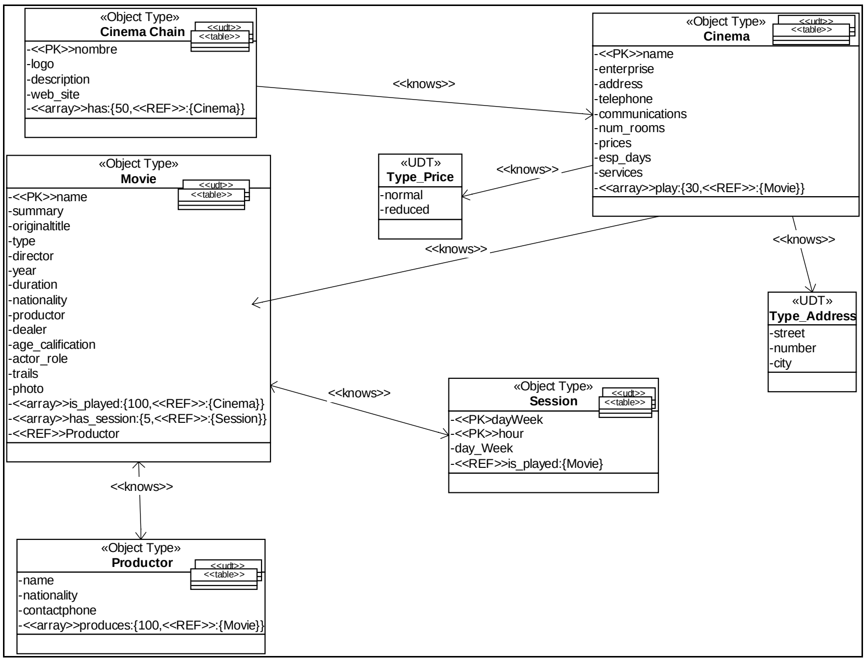

Fig. 6.- Logical Database Design in SQL:1999 (PSM)

At the **hypertext level**, we have to carry out the logical slice and navigation model. To obtain the logical slice model, each of the slices obtained in the conceptual slice model is transformed into an XML Schema using the UML notation proposed in [16]. In figure 7 we present the XML Schema obtained from the Cinema Chain slice in UML notation as well as the corresponding XML Schema code in figure 8.

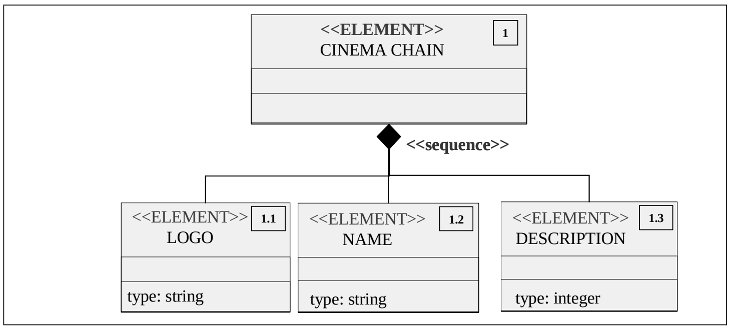

Fig. 7.- A slice represented in XML Schema in UML notation (PSM)

```
<?xml version="1.0"?>
<!-- File Name: CinemaChain.xsd -->
<xsd: schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Presentation">
    <xsd:complexType>
      <xsd:sequence>
            <xsd:element name="LOGO" type="xsd:string"/>
            <xsd:element name="NAME" type="xsd:string"/>
            <xsd:element name="DESCRIPTION" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Fig. 8.- A slice represented in XML Schema in the corresponding code

After obtaining the XML Schemas of each of the slices of the conceptual slice model, we have to build the logical navigation model. This model describes how to navigate through the pages, including access elements to them. Applying the mappings rules between models, conceptual navigational model is transformed to a logical level. So, we have to transform it into XLink, following the rules defined in [17].

You can see in figure 9, the complete navigation conceptual diagram is transformed, at the logical level, into an <<Extended Link>>, which associates an arbitrary number of resources. The extended link will be represented in XLink with an abstract class called 'Logical Navigation Model'. This class is related with all the slices that compose the logical navigation model by means of a composition association <<is_composed>>. Each slice of the slice conceptual model will be transformed at the logical level in a local resource represented with a <<XLink Locator>> element, named with the corresponding slice name. In this way, we have obtained a resource for the slices Cinema Chain, Cinema, Film and Session. The access elements of the navigational logical model: indexes, guided tours, queries and menus, can also be transformed at the logical level in a local resource represented with a <<XLink Locator>>. The navigation between two slices or between a slice and an access element (in this case, four indexes) are represented with an <<arc>> association. The information of how this traverse is done, is included in the arc association specifying its tagged values. The corresponding XLink code is shown in figure 10.

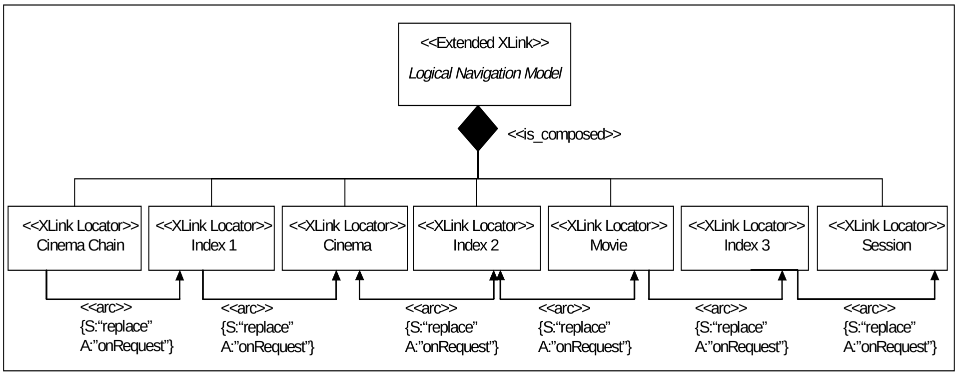

Fig. 9.- Logical Slice Model represented in XLink (PSM)

```
<Links xmlns:xlink=http://www.w3.org/1999/xlink>
<logicalNavigationModel xlink:type="extended"
title="Cinema Consortium"
<!--definition of all local resources-->
<cinemaChain xlink:type "resource"
Xlink:label="sliceCinemaChain"
</cinemaChain>
<index1 xlink:type "resource"
Xlink:label="sliceIndex1"
</index1>
<cinema xlink:type "resource"
Xlink:label="sliceCinema"
</cinema>
<index2 xlink:type "resource"
Xlink:label="sliceIndex2"
</index2>
<film xlink:type "resource"
Xlink:label="sliceFilm"
</film>
<index3 xlink:type "resource"
Xlink:label="sliceIndex3"
</index3>
<session xlink:type "resource"
Xlink:label="sliceSession"
</session>
<!-- definition of the links
           between slices-->
<link1 xlink:type="arc"
xlink:from ="sliceCinemaChain"
xlink:to="sliceIndex1"
xlink:show="replace"
xlink:actuate="onRequest" />
<link2 xlink:type="arc"
xlink:from ="sliceIndex1"
xlink:to="sliceCinema"
xlink:show="replace"
xlink:actuate="onRequest" />
<link3 xlink:type="arc"
xlink:from ="sliceCinema"
xlink:to="sliceIndex2"
xlink:show="replace"
xlink:actuate="onRequest" />
<link4 xlink:type="arc"
xlink:from ="sliceIndex2"
xlink:to="sliceMovie"
xlink:show="replace"
xlink:actuate="onRequest" />
<link5 xlink:type="arc"
xlink:from ="sliceMovie"
xlink:to="sliceIndex3"
xlink:show="replace"
xlink:actuate="onRequest" />
<link6 xlink:type="arc"
xlink:from ="sliceIndex3"
xlink:to="sliceSession"
xlink:show="replace"
xlink:actuate="onRequest" />
<link7 xlink:type="arc"
xlink:from ="sliceMovie"
xlink:to="sliceIndex2"
xlink:show="replace"
xlink:actuate="onRequest" />
<link8 xlink:type="arc"
xlink:from ="sliceIndex2"
xlink:to="sliceCinema"
xlink:show="replace"
xlink:actuate="onRequest" />
</logicalNavigationModel>
</Links>
```

Fig. 10.- Corresponding XLink Code

4 Conclusions and Further Research Topics

In this paper, we have described MIDAS, a model-driven methodology for WIS development. This methodology is a specific application of MDA for the Web platform using XML and (object-)relational technologies. MIDAS proposes some platform independent models (PIMs) and some platform specific models (PSMs). It also defines mappings between the previously mentioned models. We make a first classification of these models according to the structural and behavior dimensions defined in MIDAS.

In this paper, we have focused on the structural dimension of a WIS, describing at first the PIMs for the content, hypertext and presentation levels, and afterwards, summing up the PSMs for these levels. Next, we have shown the mappings between the models and finally, to validate the proposal, we have developed a case study of a WIS for a cinema consortium in Madrid, using the proposed UML extension and techniques of MIDAS.

Now, we are also defining the PIMs and PSMs for the behaviour dimension, in the same way that we have done for the structural one. We are also working on extending MIDAS to other platforms and technologies as J2EE or .NET and also on the implementation of the rest of the defined mappings between the models to (semi-) automate the generation of an IS.

Acknowledgements

This research has been supported by the Spanish Ministry of Science and Technology (DAWIS project, ref: TIC 2002-04050-C02-01).

References

1. Bray, T., Paoli, J, Sperberg-McQu4een, C. M. and Maler, E., Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation. Retrieved from: http://www.w3.org/TR/2000/REC-xml-20001006/, 2000.
2. Conallen, J., *Building Web Applications with UML.* Addison Wesley, 2000.
3. Eisenberg A. and Melton J., *SQL:1999, formerly known as SQL3.* ACM SIGMOD Record, Vol. 28, No. 1, pp. 131-138, March, 1999.
4. Isakowitz, T., Kamis, A. and Koufaris, M., *The Extended RMM Methodology for Web Publishing.* Working Paper IS-98-18, Center for Research on Information System. Retrieved from: http://rmm-java.stern.nyu.edu/rmm/, 1998.
5. Koch, N., Baumeister, H. and Mandel, L., *Extending UML to Model Navigation and Presentation in Web Applications.* In Modeling Web Applications, Workshop of the UML´2000. Ed. Geri Winters and Jason Winters, York, England, october, 2000.
6. Marcos, E., De Castro, V. and Vela, B. *Modelado de Servicios Web con UML: Un caso de estudio.* Workshop on Advances in Database and Information Retrival. Apizaco, Mexico, september, 2003. Accepted.
7. Marcos, E., De Castro, V. and Vela, B. *Representing Web Services with UML: A Case Study.* The First International Conference on Service Oriented Computing (ICSOC03). Submitted.
8. Marcos E., Vela B. and Cavero J. M., *Extending UML for Object-Relational Database Design*. Fourth Int. Conference on the Unified Modelling Language, UML 2001, Toronto (Canadá), LNCS 2185, Springer Verlag, pp. 225-239, 2001.
9. Marcos, E. Cáceres, P., Vela, B. and Cavero, J.M., *MIDAS/DB: a Methodological Framework for Web Database Design.* DASWIS 2001. Yokohama (Japan), November, 2001. LNCS-2465. Springer Verlag. ISBN 3-540-44122-0. September, 2002.
10. Marcos, E., Vela, B., Cavero J.M., Cáceres, P. *Aggregation and Composition in Object-Relational Database Design*. 5 th. Conference on Advances in Databases and Information Systems. Ed.: Albertas Caplinskas and Johann Eder, pp.195-209, 2001.
11. Marcos, E. , Vela, B. and Cavero J.M., Methodological Approach for Object-Relational Database Design using UML. Journal on Software and Systems Modeling (SoSyM).

Springer-Verlag. Ed.: R. France y B. Rumpe. ISSN: 1619-1366. Volumen SoSyM 2, pp.59-72, 2003..

12. Miller, J. and Mukerji, J. (Eds). (2001) Model Driven Architecture. Document number ormsc/2001-07-01. Retrieved from: http://www.omg.com/mda, 2003.
13. Miller, J. and Mukerji, J. (Eds). (2001) MDA Guide Version 1.0. Document number omg/2003-05-01. Retrieved from: http://www.omg.com/mda, 2003.
14. Retschitzegger, W. y Schwinger W. Towards Modeling of Data Web Applications- A Requirement's Perspective. In Proceedings of the America's Conference on Information Systems. V.I, pp. 149-155, 2000.
15. Soley, R. And the OMG Staff Strategy Group. *Model driven Architecture.* Object Management Group, White Paper. November 27, 2000.
16. Vela, B. and Marcos E., *Extending UML to represent XML Schemas.* The 15th Conference On Advanced Information Systems Engineering (CAISE '03). CAISE'03 FORUM. Klagenfurt/Velden (Austria). 16-20 June 2003. Ed. J. Eder, T. Welzar. Short Paper Proceedings. ISBN: 86-435-0549-8. 2003.
17. Vela, B. and Marcos E., *Hypertext Design in UML: From Conceptual to Logical Slice and Navigation Models in XLink*. 20th Int. Conference on Data Engineering (ICDE 2004), Submitted.
18. Vela, B., Cavero, J. M. and Marcos, E. *Diseño de Bases de Datos Objeto-Relacionales con UML*. 1ª Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC'2001). Anales de las Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento. Ed. Silvia Teresita Acuña y Cecilia María Lasserre. Editorial Universidad Nacional del Jujuy, 2001, pp. 59-68. Buenos Aires (Argentina). 13-15 June 2001
19. W3C XML Schema Working Group. *XML Schema Parts 0-2:[Primer, Structures, Datatypes]*. W3C Recommendation. Retrieved from: http://www.w3.org/TR/xmlschema-0/, http://www.w3.org/TR/xmlschema-1/ and http://www.w3.org/TR/xmlschema-2/, 2001.
20. W3C XML Linking Language (XLink) Version 1.0. W3C Recommendation. Retrieved from: http://www.w3.org/TR/xlink/, 2001.
21. W3C Extensible Stylesheet Language (XSL) Version 1.0. W3C Recommendation. Retrieved from: http://www.w3.org/TR/xsl/, 2003.