Model Driven Development of Secure XML Databases

Belén Vela[1], Eduardo Fernández-Medina[2], Esperanza Marcos[1] and Mario Piattini[2]

(1) Kybele Research Group. Languages and Computing Systems Department
Rey Juan Carlos University. C/ Tulipán, s/n - 28933 Móstoles, Madrid, Spain
{belen.vela, esperanza.marcos}@urjc.es
(2) Alarcos Research Group. Information Systems and Technologies Department
UCLM-Soluziona Research and Development Institute
University of Castilla-La Mancha. Paseo de la Universidad, 4 – 13071 Ciudad Real, Spain
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es

ABSTRACT

In this paper, we propose a methodological approach for the model driven development of secure XML databases (DB). This proposal is within the framework of MIDAS, a model driven methodology for the development of Web Information Systems based on the Model Driven Architecture (MDA) proposed by the Object Management Group (OMG) [20]. The XML DB development process in MIDAS proposes using the data conceptual model as a Platform Independent Model (PIM) and the XML Schema model as a Platform Specific Model (PSM), with both of these represented in UML. In this work, such models will be modified, so as to be able to add security aspects if the stored information is considered as critical. On the one hand, the use of a UML extension to incorporate security aspects at the conceptual level of secure DB development (PIM) is proposed; on the other, the previously-defined XML schema profile will be modified, the purpose being to incorporate security aspects at the logical level of the secure XML DB development (PSM). In addition to all this, the semi-automatic mappings from PIM to PSM for secure XML DB will be defined.

1 Introduction

Though relational database (DB) technology still plays a central role in the data management arena today, we have seen numerous evolutions of this technology, such as the XML DBs. A key requirement underlying those recent data management systems is a demand for adequate security. Fine-grained flexible authorization models and access control mechanisms, in particular, are being called for [1]. Traditionally, the information of XML documents was stored directly in XML files or in conventional Database Management Systems (DBMSs), by mapping the XML data to relational data stored in relational tables or by using the data types supplied for supporting file management, as for example the CLOB (Character Large OBject) type. The XML DBs are now emerging as the best alternative for storing and managing XML documents. At present, there are different solutions to store XML documents, and they could be roughly categorized, according to [25], into two main groups: native XML DBMSs like Tamino [23]; and XML DB extensions enabling the storage of XML documents within conventional, usually relational or Object-Relational (OR) DBMSs such as Oracle. This latter includes, since version 9*i* release 2, new features for the storage of XML (Oracle's XML DB) [22]. In [25] a study of different XML DB solutions is performed.

For most organizations, management, security and confidentiality of information are critical topics [6]. Moreover, as some authors remark, information security is a serious requirement which must be carefully considered, not as an isolated aspect, but as an element that is present in all stages of the development life cycle [5,11,13]. A body as important as the Information Systems Audit and Control Foundation insists on the fact that security should be considered explicitly and as an integral item in all the development stages of an information system [15]. In the case of the XML DBs, security is also a key aspect that must be explicitly considered. It has to be taken into account in an orthogonal way for the complete development process of this kind of DB. Access control models have been widely investigated and several access control systems, specifically tailored to XML documents, have been developed [2,3,4,12,14,18]. However, all of them define security criteria directly over the XML documents or DTDs.

Our approach is based on the Model Driven Architecture (MDA) proposed by the Object Management Group (OMG) and allows us to define the security specifications on the conceptual data model, independently of the target logical data model (DB schema). Starting from this secure conceptual data model we transform it semi-automatically into a secure XML DB, as a logical data model.

Although there are different ideas for integrating security into the information systems development process, information security within the scope of DBs tends to be considered only from a cryptographic point of view. Recently, we have proposed a methodology for relational DB which integrates security aspects at all stages of the development process [7]. However, to the best of our knowledge, there are no works that deal with security when developing an XML DB.

In this paper, we will integrate the security aspect into the methodological approach for XML DB development [24] framed in MIDAS [16], a model driven methodology for the development of Web Information Systems (WIS). MIDAS proposes the use of standards in the development process, as well as the use of UML in modelling the WIS, irrespective of the abstraction level and the aspect of the system to be modelled. As UML does not allow us to represent all the necessary models, MIDAS incorporates some existing UML extensions and defines or adapts some new ones, whenever necessary [8,17].

In the next section, we will introduce the secure XML DB development process in the framework of MIDAS, where the Platform Independent Model (PIM) is the conceptual data model. It will be represented with an extended UML class diagram that includes the security aspect at this level. This profile will be summed up in section 3. As data Platform Specific Model (PSM) in MIDAS, it is proposed to use the OR model or the XML Schema model, depending on the technology used. In this paper we will show the part corresponding to secure XML DB development. The PSM employed will therefore be the XML Schema model. In section 4, we will present an adaptation of the previously-defined profile for XML DBs for the incorporation of specific security aspects into this kind of DBs. In section 5, we will show the mappings from the secure data PIM to the secure data PSM which will be the schema of the secure XML DB. These mappings are based on those defined in [24], where the rules to obtain the data PSM are described, but without taking into consideration security aspects. In this paper, we will adapt such rules so as to obtain the schema of an XML DB which includes the necessary constraints for security. Finally, in section 0, we will put forward our main conclusions and present our future work.

2 Secure XML DB Development Process

MIDAS proposes a model driven architecture based on MDA and, when modelling the system, considers, the aspects of *content*, *hypertext* and *behaviour* at the levels of Computation Independent Models (CIMs), common to all the system, PIMs and PSMs. In Figure 1 we can see the simplified MIDAS MDA.

In this paper, we will focus on the **content** aspect, which corresponds to the traditional concept of a DB, for the ***PIM*** and ***PSM*** levels. The development of a DB depends on several aspects; on the one hand, on whether there is already a DB within the organization or not, and, on the other hand, on the technology to be used: in other words, if we aim to use an OR DB [17] or an XML DB [24].

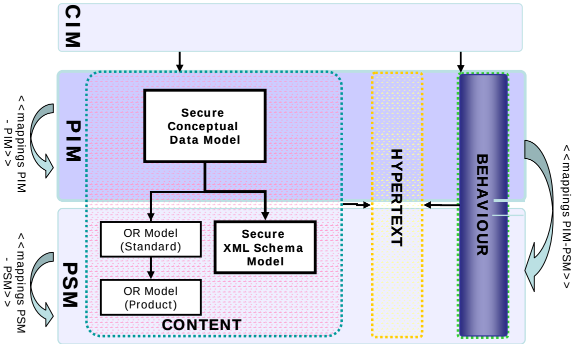

Figure 1. Simplified MIDAS Architecture

Moreover, a third dimension is considered in MIDAS, and it includes all aspects to be taken into account when developing a WIS, such as the system architecture or security. This third dimension is orthogonal to the ones presented in Figure 1.

In the cases in which the DB that we want to develop includes information to be protected, the security aspect will have to be taken into account from the first stages of the DB development. So, for the model driven development of a secure XML DB we have to perform the following tasks:

- At the ***PIM*** level, the secure data conceptual model is carried out without considering the selected technology, since this model is platform independent. This secure data PIM is represented through an extended UML class diagram, so as to be able to represent *security* aspects together with a set of security constraints that have been expressed through OSCL language [8], as we will see in the next section.
- At the ***PSM*** level, the data logical design is performed, taking into account the selected technology. In our case, this is an XML DB. We will start from the secure data PIM obtained at the previous level and will apply the mappings summarized in section 5. The secure data PSM will be represented through an XML schema in extended UML (see section 4). In this case, the DB schema will be the XML schema, which takes into account the necessary security aspects.

3 Secure Data PIM

To develop a secure data PIM, a secure UML profile has been developed (for more details, see [8]). The defined UML profile allows us to classify both data and users according to different classification criteria. These criteria are the following ones:

- **Security levels**: to define a hierarchy of levels such as those traditionally employed in the army: unclassified, confidential, secret and top secret.
- **User roles**: to define a hierarchical set of user roles that represents the hierarchical functions within an enterprise.
- **User categories**: to define a horizontal organization or classification (non hierarchical) of user groups.

In addition to this classification information, the profile allows us to define three kinds of constraints:

- **Data dynamic classification rules**: to define the classification data of different instances, depending on the value of one or several attributes of the instances.
- **Audit rules**: They specify situations in which it is interesting to us to register an audit trace to analyze which users have accessed (or have tried to access) information. To do so, conditions expressed in OCL are defined.
- **Authorization rules**: to define which users will be allowed to access to which data and to perform which actions depending on a condition expressed in OCL.

Our security model is general, and the classification criteria, together with the *data dynamic classification rules* and the authorization rules, allow us to integrate several access control models, such as the mandatory access control, a simplified role based access control, discretionary access control and access control based on rules. The coexistence of these rules frequently provokes conflicts, that we solve by applying a set of conflict resolution rules defined in [8, 9, 10].

For the definition of all these elements, we consider the UML profile known as *Conceptual Secure DB* (extension of UML and OCL to design secure DBs), which is composed of a set of data types, tagged values and stereotypes, together with the definition of a set of well-formedness rules. The package containing all the stereotypes defined within this UML profile can be analyzed in Figure 2. These stereotypes can be classified into three categories:

- The stereotypes necessary for representing security information in the *model elements.*
- The stereotypes needed to model the *security constraints* when defining: a) the dynamic classification of any element, b) audit rules expressed in OCL and c) authorization rules.
- The *UserProfile* stereotype that is necessary to specify security constraints on what might be seen as a property of a user or a group of users, for instance; citizenship, age, etc.

A detailed description of all these stereotypes, as well as the tagged values that have been defined for them, can be found in [8].

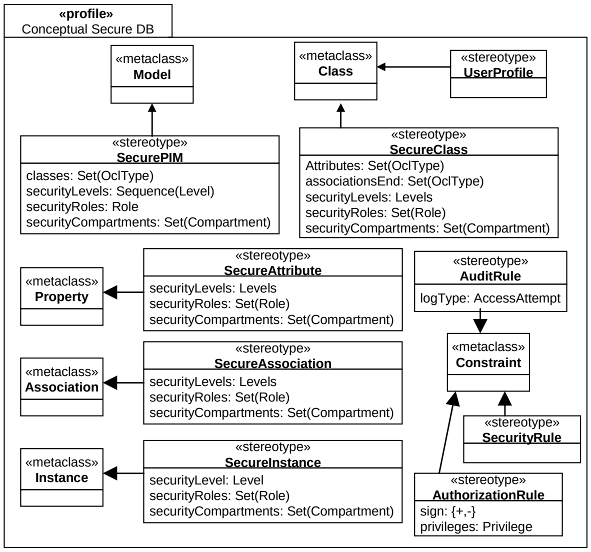

Figure 2. *Conceptual Secure DB* profile

4 Secure Data PSM

In MIDAS, the XML schema model is proposed as data PSM. It is represented in extended UML, using the profile defined in [24]. To include the security aspects in the model, in this paper we have adapted such a profile by adding the elements that are needed to be able to consider the aspect of *security*.

In Figure 3, we will show the elements that have been added, with the goal of adapting the profile so that it is able to represent secure XML schemas through a UML class diagram. The extension defines a set of new stereotypes. The aim is for it to be able to consider all the components of a secure XML in a graphical notation of UML, maintaining the associations, the order and the links between the different elements.

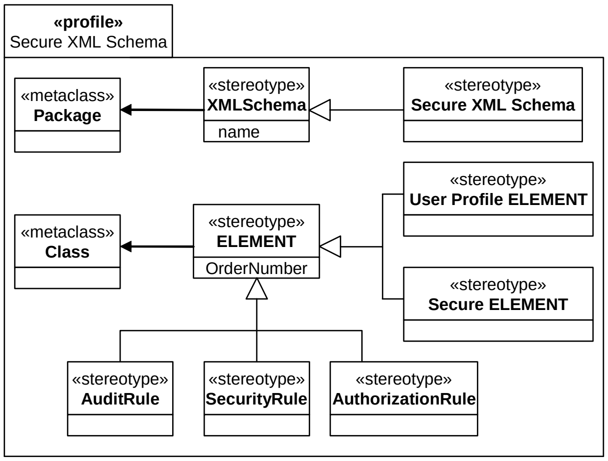

Figure 3. *Secure XML Schema* profile

5 Mappings from PIM to PSM

In the same way that methodologies for relational or OR DBs propose some rules for the transformation of a conceptual schema into a standard logical one, in MIDAS, mappings from the data PIM to the data PSM are proposed. In this work, we have defined the transformation rules needed to obtain a secure data PSM from the secure data PIM. The work of [24] is taken as a basis, where the different mappings to obtain the schema of an XML DB were defined (but where security was not yet taken into account).

- **Transformation of the secure data PIM:** The data conceptual model, that is, the *secure PIM*, is transformed, at the PSM level, into an XML schema named '*Secure Data PSM*'. It will be represented with a UML package stereotyped with <<Secure XML SCHEMA>> and will be called as the XML schema. It will include all components of the secure XML schema (PSM). Furthermore, it will contain the security attributes (*securityLevel*, *securityRoles* and *securityCompartments*) of the secure PIM. These attributes will be defined within the XML schema as global elements. They could have been included as schema attributes but if they were represented in such a way, they would not be considered first order elements and the fact that they could have a multiple maximum cardinality could not be collected either.
- **Transformation of the *User Profile* class:** This class includes the information that we want to record for each user. It will be transformed by including a global element stereotyped with <<User Profile ELEMENT>>, which will contain a sequence complexType with all class attributes as subelements.
- **Transformation of secure classes:** In a generic way, a UML class is transformed into an element of the XML schema with the same name as the class it comes from [24]. To transform secure UML classes, stereotyped with <<SecureClass>>, we have to include the secure characteristics that they have, too. Secure classes can have three specific attributes: *securityLevel*, *securityRoles* and *SecurityCompartments*. They will be transformed into secure elements stereotyped with <<Secure ELEMENT>>. Each secure element will contain a complexType of sequence type, which will contain as subelements, among others, the secure attributes, indicating, with the subelements attribute *maxOccurs*, the number of possible instances of the security attributes.
- **Transformation of secure attributes:** Due to the fact that the attributes of a class, according to the proposal of [24], are transformed as subelements of the element that represents the UML class to which those attributes belong, if an attribute has its own security attributes associated with it, these attributes will be represented as subelements of the element that represents the corresponding attribute. Thus, the security attributes defined within an attribute will be transformed into <<Secure ELEMENT>> subelements.
- **Transformation of secure associations:** Regarding the transformation of associations, a detailed study of the most appropriate way to map them at the PSM level was carried out in [24]. The associations between two classes are transformed, in a generic way, by including a subelement in one of the elements, corresponding to one of the classes implied in the relationship with one or several references to the other element implicated in the association. If it were a secure association, this subelement would have subelements to represent the corresponding security attributes (*securityLevel*, *securityRoles*, *securityCompartment*) stereotyped as <<Secure ELEMENT>>.
- **Transformation of security constraints:** When transforming the security constraints that had been defined at the PIM level, these can be defined for any element (model or class), although it is normal to define them at the class level. If they are defined at the model level, global elements to collect this fact will be created. In the rest of the cases, subelements of the elements they depend on will be created. There are three types of constraints:

a) **Audit Rules**: They will be transformed by creating a subelement stereotyped with <<AuditRule>> with the name of "AuditRule_" plus the number of the rule. This element will be of the complexType and it will contain a sequence formed by two elements: One *AuditRuleType* element of simple Type of the *string* base type with a constraint of enumeration type with the values *all*, *frustratedAttempt*, *successfullAccess*; and another element *AuditRuleCondition* that will be an element of *string* type, that will contain the XPATH expression associated with the OCL expression.

```
<complexType>
  <sequence>
    <element name= “AuditRuleType”>
      <simpleType>
              <restriction base= ”string”>
                 <enumeration value= ”all”/>
                 <enumeration value= ”frustatedAttempt”/>
                <enumeration value= ”successfullAccess”/>
              </restriction>
      </simpleType>
    </element>
    <element name= “AuditRuleCondition” type=”string”/>
  </sequence>
</complexType>
```

b) **Security Rule:** The dynamic classification of any PIM element will be transformed by creating a subelement stereotyped with <SecurityRule>>, with the name “SecurityRule_” plus the number of the rule. This element will be of complexType and it will contain one element of string type with the XPATH expression associated with the OCL expression.

```
<complexType>
   <sequence>
      <element name= “SecurityRuleCondition” type=”string”/>
   </sequence>
</complexType>
```

c) **Authorization Rules:** These will be transformed by creating a subelement stereotyped with <<AuthorizationRule>> with the name “AuthorizationRule_”, plus the number of the rule. This element will be of a complexType and it will contain a sequence formed by three elements: An *AuthorizationRuleSign* element of simpleType of *string* base type with a constraint of enumeration type with the values: + or - ; another *AuthorizationRulePrivileges* element of simpleType of *string* base type with a constraint of enumeration type with the values: *read*, *insert*, *delete*, *update* and *all*; and an *AuthorizationRuleCondition* element of string type that will contain the XPATH expression associated with the expression in OCL.

```
<complexType>
  <sequence>
    <element name= “AuthorizationRuleSign”>
      <simpleType>
            <restriction base=”string”>
               <enumeration value=”+/> <enumeration value=”-”/>
            </restriction>
      </simpleType>
    </element>
    <element name= “AuthorizationRulePrivileges”>
      <simpleType>
            <restriction base=”string”>
              <enumeration value=”read”/>
              <enumeration value=”insert”/>
              <enumeration value=”delete”/>
              <enumeration value=”update”/>
              <enumeration value=”all”/>
            </restriction>
      </simpleType>
    </element>
    <element name= “AuthorizationRuleCondition” type=”string”/>
  </sequence>
</complexType>
```

According to MDA, once we have applied these rules, the next step is the mapping from PSM to Code of specific DBMSs. These DBMSs usually do not provide security solutions for solving the security issues we consider in our approach, but they support most of XML standards (DOM, XSL, XSLT, XPath, etc.), which allow us to easily implement all these security specifications.

6 Conclusions and Future Work

At the present time, there are different solutions for the storage of XML data but there is no methodology for the systematic design of XML DBs that incorporates security in the development process from its early phases.

In this work, we have integrated the security aspect into the methodological approach for the development of an XML DB in the framework of MIDAS, a model-driven methodology for the development of WIS based on MDA. In the case of the specified development process for secure XML DB, for the secure data PIM, a UML extension to incorporate security aspects at the conceptual level is used. For the secure data PSM we have modified the previously-defined XML DB profile. The incorporation of security aspects has been our main goal. Moreover, we have defined mappings from secure data PIM to secure data PSM that will be the secure XML DB schema. From this logical model of the secure XML DB (PSM), we will obtain the code for the specific XML DB product that we want to use, in a semi-automatic way. Up to now, we have studied the security aspects for the Oracle 10g product, but in future work, we will study other XML DBMSs in detail, in order to analyze which of them take into account security aspects, and how.

A case study for the management of hospital information has been developed, to validate our proposal; we have left this out for the sake of space.

We are now working along several different lines, in an attempt to extend the proposal of this paper. One of these, on which we have already started to work, is the automation of the transformations of the constraints expressed in OCL at the PIM level, to convert them into XPATH language. Moreover, our intention is to automate the transformations between the metamodels and the corresponding models using the incipient Query View Transformation (QVT) proposal [20], which aims to become the standard for defining transformations.

We are also studying the possibility of using XACML [19] as a PSM security rules specification language that could complement the current PSM model (XML Schema). In fact, XACML is a powerful standard language that specifies schemas for authorization policies and for authorization decision requests and response, which is applicable to a wide range of applications, and which can integrate many security policies into a complete security model.

In addition, we want to define queries using the XQuery language, in order to obtain information about the security aspects of the XML DB.

We have a further goal, which is to perform several case studies to detect new needs. These would also analyze the advantages of incorporating security aspects provided by the different XML DB administrators, not only native ones, but also the XML extensions that DBMSs have. At the same time, we are going to include the security aspect in the subsystem for the semi-automatic development of XML DBs of the tool CASE that we are developing.

Acknowledgements

We would like to thank the reviewers for their valuable comments, which have helped us to improve this paper.

This research has been carried out in the framework of the following projects: GOLD financed by the Spanish Ministry of Education and Science (TIN2005-00010/), DIMENSIONS (PBC-05-012-2) financed by the FEDER and by the "Consejería de Ciencia y Tecnología of the Junta de Comunidades de Castilla-La Mancha" and the RETISTIC network (TIC2002-12487-E) financed by the "Dirección General de Investigación" of the Spanish Ministry of Science and Technology.

References

1. Bertino, E. and Sandhu, R. *Database Security – Concepts, Approaches, and Challenges.* IEEE Transactions on Dependable and Secure Computing. Vol. 2, Nº 1, January-March 2005, pp 2-19, 2005.
2. Bertino, E. and Ferrari, E. *Secure and Selective Dissemination of XML Documents.* ACM Transactions on Information and System Security. Vol. 5, Nº 3, pp. 290-331, 2002.
3. Bertino, E., Castano, S., Ferrari, E., and Mesiti, M. *Specifying and Enforcing Access Control Policies for XML Document Sources*. World Wide Web Journal. Vol. 3. Nº 3, Baltezer Science Publisher, pp. 139-151, 2000.
4. Damiani, E., De Capitani di Vimercati, S., Paraboschi, S. and Samarati, P. *Securing XML Documents.* Proceedings of the 2000 International Conference on Extending Database Technology (EDBT2000), Konstanz, Germany, pp.121-135, 2000.
5. Devanbu, P. and Stubblebine, S. *Software engineering for security: a roadmap*. In: A. Finkelstein (Ed.), The Future of Software Engineering, ACM Press pp. 227-239, 2000.
6. Dhillon, G. and Backhouse, J. *Information System Security Management in the new Millennium.* Communications of the ACM. 43, 7. pp. 125-128, 2000.
7. Fernández-Medina, E. and Piattini M. *Designing secure databases.* Information & Software Technology 47(7), pp. 463-477. 2005
8. Fernández-Medina, E. and Piattini, M. *Extending OCL for Secure Database Design*. In Int. Conference on the Unified Modeling Language (UML 2004). Lisbon (Portugal), October, 2004. Springer-Verlag, LNCS 3273, pp. 380-394. 2004.
9. Fernández-Medina, E., Trujillo, J., Villarroel, R. and Piattini, M. *Extending UML for Designing Secure Data Warehouses.* In Conceptual Modeling (ER 2004). Shanghai (China). November, 2004. Springer Verlag. LNCS 3273, pp. 217-230.
10. Fernández-Medina, E., Trujillo, J., Villarroel, R. and Piattini, M. *Access Control and audit Model for the Multidimensional Modeling of Data Warehouses.* Decision Support Systems. 2006 (In Press).
11. Ferrari E. and Thuraisingham B., *Secure Database Systems*, in: M. Piattini, O. Díaz (Ed.), Advanced Databases: Technology Design. Artech House, 2000.
12. Gabillon, A. and Bruno, E. *Regulating Access to XML Documents.* Proceedings of the 15th Annual IFIP WG 11.3 Working Conference on Database Security, pp. 299-314, 2001.
13. Ghosh, A., Howell C., Whittaker J., *Building software securely from the ground up*, IEEE Software 19 (1) (2002), pp. 14-17, 2002.
14. He, H. and Wong, R.K. *A Role-Based Access Control for XML Repositories.* Proceedings of the First International Conference on Web Information Systems Engineering (WISE'00), 2000.
15. ISACF, Information Security Governance. *Guidance for Boards of Directors and Executive Management, Information Systems Audit and Control Foundation,* USA, 2001.
16. Marcos, E., Vela, B., Cáceres, P. and Cavero, J.M. *MIDAS/DB: a Methodological Framework for Web Database Design.* DASWIS 2001. Yokohama (Japan), November, 2001. Springer-Verlag, LNCS 2465, pp. 227-238, 2002.
17. 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 and B. Rumpe. Vol. SoSyM 2, pp.59-72, 2003.
18. Murata, M., Tozawa, A., Kudo, M. and Hada, S. *XML Access Control Using Static Analysis.* Proceedings of the 10th ACM Conference on Computer and Communication Security, pp.73-84, 2003.
19. OASIS. *eXtensible Access Control Markup Language (XACML 2.0).* Retrieved from: http://www.oasis-open.org.
20. OMG. *MDA Guide Version 1.0.* Document number omg/2003-05-01. Ed.: Miller, J. and Mukerji, J. Retrieved from: http://www.omg.com/mda, 2003.
21. OMG, Query/Views/Transformation RFP. 2002. Retrieved from: http://omg.org/ad/2002-4-10.
22. Oracle Corporation. *Oracle XML DB. Technical White Paper.* Retrieved from: www.otn.com, 2003.
23. Software AG. *Tamino X-Query. System Documentation Version 3.1.1.* Software AG, Darmstadt, Germany. Retrieved from: www.softwareag.com, 2001.
24. Vela, B., Acuña, C. and Marcos, E. *A Model Driven Approach for XML Database Development*, 23rd. International Conference on Conceptual Modelling (ER2004). Shanghai (China), November, 2004. Springer Verlag, LNCS 3288, pp. 780-794. 2004.
25. Westermann, U. and Klas W. *An Analysis of XML Database Solutions for the Management of MPEG-7 Media Descriptions*. ACM Computing Surveys, Vol. 35 (4), pp. 331-373, 2003.