Conceptual Modeling for XML: A Survey[*]

Martin Necasky

Charles University, Faculty of Mathematics and Physics,
Malostranske nam. 25, 118 00 Praha 1, Czech Republic
martin.necasky@mff.cuni.cz

Abstract. Recently XML is the standard format used for the exchange of data between information systems and is also frequently applied as a logical database model. If we use XML as a logical database model we need a conceptual model for the description of its semantics. However, XML as a logical database model has some special characteristics which makes existing conceptual models as E-R or UML unsuitable. In this paper, the current approaches to the conceptual modeling of XML data are described in an uniform style. A list of requirements for XML conceptual models is presented and described approaches are compared on the base of the requirements.

Keywords: conceptual modeling, XML, XML Schema

1 Introduction

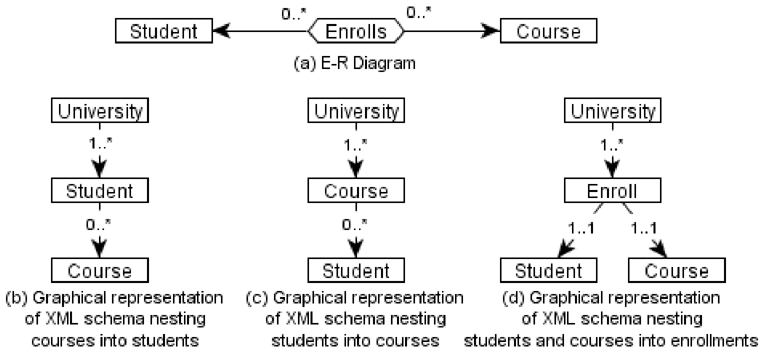
Today XML is used for the exchange of data between information systems and it is frequently used as a logical database model for storing data into databases. If we use XML as a logical database model we need a conceptual model for modeling XML data. There is the Entity-Relationship (E-R) [19] model for the conceptual modeling of relational data. However, XML as a logical database model has some special differences which makes the E-R model unsuitable for the conceptual modeling of XML data. The main differences are the following:

- hierarchical structure
- irregular structure
- ordering on siblings
- mixed content

These features cannot be properly modeled in the E-R model. There are some approaches, for example Extended E-R [1], EReX [11], EER [12], XER [17], ERX [16], and C-XML [5], trying to extend the E-R model to be suitable for the conceptual modeling of XML data. It is possible to extend the E-R model to model ordering, mixed content and irregular structure of XML data. However, there is a problem with the modeling of a hierarchical structure of XML data.

[*] This paper was supported by the National programme of research (Information society project 1ET100300419)

Suppose an E-R diagram with a relationship type *Enroll* between two entity types *Student* and *Course* representing courses enrolled by students. Each student may enroll zero or more courses and each course may be enrolled by zero or more students. The diagram is shown in Figure 1(a).



(a) E-R Diagram

(b) Graphical representation of XML schema nesting courses into students

(c) Graphical representation of XML schema nesting students into courses

(d) Graphical representation of XML schema nesting students and courses into enrollments

Fig. 1. Representation of E-R relationship type in a hierarchical structure

Figures 1(b), (c), and (d) show possible representations of the relationship type in a hierarchical structure. Oriented arrows denote a nesting. There is not the best nesting of the concepts. The nesting of courses into students illustrated by Figure 1(b) is suitable when we need to see students and the courses they enrolled. The nesting of students into courses illustrated by Figure 1(c) is suitable when we need to see courses and the students enrolled in them.

The previous example shows another difference between the conceptual level of XML and the E-R model. This difference is not in the structure but it is in the usage of XML. It is shown that there may be many ways of how to use entity types connected together by a relationship type. If we represent data in the form of XML, each of these ways may require another hierarchical ordering of the entities. However, this feature cannot be effectively modeled by the E-R model.

Another possibility of how to model XML data is to start from a hierarchical structure. This approach may be called the hierarchical approach. There are conceptual models based on the hierarchical approach, for example X-Entity [10], ORA-SS [4], and Semantic Networks for XML [6]. The base of a schema in the hierarchical approach is a tree, whose nodes are entity types and edges are relationship types between entity types. Figures 1(b), (c), and (d) show examples of a basic hierarchical schemata.

The hierarchical approach is able to solve the mentioned problem with different views of the same data. For each of the views there is a separate tree. However, a problem with the modeling of attributes of relationship types or with the modeling of n-ary relationship types, effectivelly solved in the E-R model, arises. Another problem arises when deciding which of hierarchical organizations of the same data is the best to select as the basic organization used for the data storage.

The goal of this paper is to describe the existing conceptual models for XML based on the E-R model and on the hierarchical approach. There are approaches based on the UML (Unified Modeling Language) [15] and ORM (Object Role Modeling) [8] models, too. However, we do not describe them in this paper. We propose a list of requirements for conceptual models for XML and compare the described models against the requirements. The main contributions of this paper are the unified descriptions of the conceptual models and the comparison of the models against the list of requirements. This paper is an abbreviated version of the full paper [14] where all the conceptual models mentioned in this paper are compared in detail and described in an unified formalism.

Section 2 introduces the list of requirements for conceptual models for XML. Section 3 describes representatives of the conceptual models for XML based on the well-known E-R model. Section 4 describes representatives of the hierarchical conceptual models for XML. Section 5 compares the described conceptual models against the requirements introduced in Section 2.

2 Requirements for Conceptual Models for XML

Requirements for conceptual models for XML are summarized in this section. There are two groups of the requirements described. The first group consists of *general requirements* covering general goals of the XML conceptual modeling. The second group consists of *modeling constructs requirements* covering requirements on what kinds of modeling constructs should XML conceptual models support.

2.1 General Requirements

Independence on XML schema languages The conceptual model should be independent on a certain XML schema language (XML Schema [7], DTD, ...). The constraints given by a certain XML schema language should not be propagated to the conceptual level.

Formal foundations The modeling constructs of the conceptual model should be described formally, which allows to compare the model with other conceptual models or to describe the operations on the model structures and modeled data (for example, data transformation between two conceptual schemata or their integration).

Graphical notation A user-friendly graphical notation for the formal modeling constructs should be offered by the conceptual model.

Logical level mapping There should be algorithms for mapping of the conceptual modeling constructs to the XML logical level. The logical schema should implement as many integrity constraints arised from the conceptual schema as possible. It may require the usage of more than one XML schema language for the logical level description (XML Schema and Schematron [9], for example). The hierarchical structure of the XML data should be utilized as much as possible on the logical level.

Different structures on the logical level The XML logical level is hierarchical. However, there are different users with different requirements accessing the modeled data on the logical level. Hence, there can be different hierarchical views of the same data. Each of the views suits to different requirements. It should be possible to model the different hierarchical views on the conceptual level and translate them to the corresponding views on the logical level. Moreover, there should be algorithms allowing automatic translation of data from one logical view to another logical view (using XSLT [2], for example).

Semantic web mapping With the increasing usage of the semantic web technologies the problem of publishing data in the form of RDF [13] triples described by RDF Schema [13] or OWL [18] arises. One possible solution is to have the data internally represented in the form of XML and translate them to the RDF triples represented in the form of RDF/XML [13] utilizing XSLT. The conceptual model for XML should consider this problem. It would be useful to have algorithms for the translation from the conceptual level to the semantic web level where the structures from the conceptual level are described using OWL. It would allow companies to publish their internally represented data on the semantic web and, backwards, to obtain data from the semantic web and integrate them to the internal representation automatically.

2.2 Modeling Constructs Requirements

Hierarchical structure Although it can be useful to keep a document designer out of the hierarchical structure of XML data on the conceptual level, the conceptual model should offer modeling constructs for modeling nesting explicitly. For example, aggregation relationship types can be used. However, non-hierarchical relationship types (for example, association relationship types or references) should be offered too. The conceptual model should introduce contructs for modeling a recursive structure.

Cardinality for all participants The hierarchical structure of XML data restricts the specification of cardinality constraints only to the nested participants of the relationship type. However, it should be possible to specify cardinality constraints for the all participants on the conceptual level.

***N*-ary relationship types** For the same reason, the modeling of n-ary relationship types and their translation to the XML logical level is problematic. However, it should be possible to model n-ary relationship types on the conceptual level.

Attributes of relationship types For the same reason again, the modeling of attributes of relationship types is problematic. Nor the nesting nor the concept of referential integrity on the XML logical level do not allow to directly express attributes of relationship types. However, the conceptual model should allow to model attributes of relationship types.

Ordering XML is ordered and this property should be propagated to the conceptual level. It should be possible to express the ordering on values of attributes, the ordering on concepts connected with another concept (for example, a book has a title page first, followed by an abstract, chapters, appendixes and a bibliography in this order) and the ordering on a participant of a relationship type (for example, the list of authors of a book or the list of chapters of a book are ordered).

Irregular and heterogeneous structure XML data may have irregular and heterogeneous structure. The conceptual model should introduce constructs for modeling such a structure. For example, variant-valued constructors for constructing attributes or disjunctive relationship types should be introduced.

Document-centric data The difference between the conceptual models for XML and the other conceptual models is that the conceptual models for XML must allow to model document-centric data. It means that not only the real-world objects with attributes and relationships but also the certain parts of documents are modeled on the conceptual level. Hence, there should be corresponding modeling constructs offered by the conceptual model. It means to allow attributes and relationships of a given concept to be mixed with a text when represented in a document content. However, the mixed content should not be restricted as it is restricted by XML Schema. Some form of generalized mixed content should be introduced allowing to specify where the text values may appear exactly (as it is possible in Relax NG [3] schemata, for example).

Reuse of content The reuse of content should be supported by the conceptual model. For example, the concept inheritance (modeled by IS-A relationship types in E-R, for example) supports the reuse of content. However, the conceptual model may be inspired in the XML Schema language and may support named types and named groups of concepts on the conceptual level.

Integration of conceptual schemata XML data are often used for the data integration. However, it can not be done effectivelly and automatically without the support on the conceptual level. A conceptual model for XML should offer modeling constructs to support an integration of schemata on the conceptual level and it should allow to merge different conceptual schemata to an overall conceptual schema. Further, it would be useful to generate XSLT transformation scripts to translate data corresponding to one conceptual schema to data corresponding to another conceptual schema.

3 E-R Based Conceptual Models for XML

In this section, we describe two representatives of the conceptual models for XML based on the E-R model. The first representative is the Extended E-R model [1] and the second representative is the EReX model [11].

3.1 Extended E-R Model (by Antonio Badia)

Extended E-R model proposed by Badia in [1] is a minimalistic extension to the E-R model. The extension is based on the idea of integration of structured and semistructured data where an overall conceptual schema is needed. Moreover, the author proposes algorithms for the translation of E-R schemata to relational schemata and to DTD schemata. Further, he studies the utilization of combination of relational schemata and DTD schemata for a data representation. The author proposes the following DTD based extensions to the E-R model.

- Each attribute is marked as *optional* or *required*. If an entity type has an optional attribute its entities may or may not have a value of the attribute. If an entity type has a required attribute its entities must have a value of the attribute.
- A choice between two or more attributes called *choice attribute* can be modeled. A choice attribute can be *inclusive* or *exclusive*. If an entity type has an inclusive choice attribute its entities may have values of one or more of the attributes in the choice. If an entity type has an exclusive choice attribute its entities may have only a value of one of the attributes in the choice.

In a graphical representation, an optional attribute is connected to the corresponding entity type by a solid line with two dashes crossing it. A choice of attributes is expressed by marking the choice with an upward triangle, with the choices in the opposed side of the triangle.

Figure 2 displays the entity type *Student* having the optional attribute *phone* and the exclusive choice attribute involving the attributes *hosteladdr* and *homeaddr*, i.e. each student has a hostel address or a home address but not both.

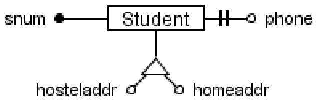


Fig. 2. Extended E-R Diagram

3.2 EReX

EReX is an extension to the E-R model proposed by Mani in [11]. The author introduces the following extensions to the E-R model:

- *Categorization* of entity types can be modeled using *category relationship types*. Category relationship types are a special kind of binary relationship types similar to IS-A relationship types from the well-known E-R model. A category relationship type is displayed by an arrow with the label CAT going from its category entity type to its categorizied entity type. The difference between the IS-A relationship types and the category relationship types is that a categorized entity type may have an empty key (i.e. an entity type with an empty key must be categorized). Moreover the integrity constraints called *coverage constraints* can be specified on categorizied entity types.

- *Total* and *exclusive coverage constraints* can be specified for categories and for roles of entity types in relationship types. A total coverage constraint specifies that the union of sets of instances of all included categories or roles must be the same as a set of instances of the categorizied entity type or the entity type with the included roles. An exclusive coverage constraint specifies the disjunction between the sets of instances of the included categories or roles. We do not formally define the coverage constraints here. We show them only in a form of the examples illustrated in Figure 3.
- *Order constraints* can be specified for participants of a relationship type. An ordering on a participant E of a relationship type R is displayed by a thick solid line between R and E. If an ordering on E in R is specified, then for a given entity e of the entity type E the set of relationships of the relationship type R with e as a participant is ordered.

The extending modeling constructs of the EReX model are demonstrated by the schema in Figure 3. It displays the categorized entity type $Person$ and its categories $Student$ and $Professor$. The key of $Person$ is empty. Further, there are the entity types $Book$ and $Paper$ connected with $Professor$ by the relationship types $AuthorOfB$ and $AuthorOfP$, respectively. Attributes of $Book$ and $Paper$ are not displayed. There is an ordering specified on the entity types $Book$ and $Paper$ in the relationship types $AuthorOfB$ and $AuthorOfP$, respectively. It means, that the authors of a given paper or a given book are ordered.

The total coverage constraint $Student + Professor = Person$ specifies, that each person is a student or a professor and there are no other persons. The exclusive coverage constraint $Student|Professor$ specifies that students and professors are disjoint. The total coverage constraint $AuthorOfB.pbook + AuthorOfP.ppaper = Professor$ specifies that each professor is an author of some paper or book.

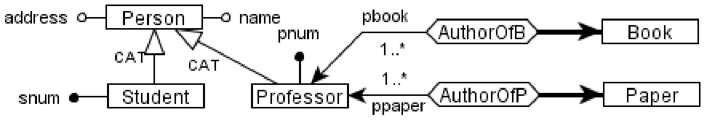


Fig. 3. EReX Diagram

4 Hierarchical Conceptual Models for XML

The extensions of the E-R model allow to model conceptual schemata with a graph structure. However, XML schema languages allow to express relationship types only by nesting and references. It is possible to express all the relationship types from an E-R schema by references, but it leads to flat schemata and the advantages of the hierarchical structure of XML are not utilized. On the other hand, if the hierarchical structure is used to express relationship types in a conceptual schema the problem with the decision about what to nest arises. Another problem is how to represent n-ary relationship types and attributes of relationship types.

In this section we describe a basic hierarchical conceptual model for XML first. In the next subsections, we describe two representatives of the conceptual models for XML based on the hierarchical approach. The first representative is the ORA-SS model [4] and the second representative is the Semantic Networks for XML model [6].

4.1 Basic hierarchical conceptual model for XML

The basic hierarchical conceptual model for XML can be easily defined as a restriction of the E-R model where only the binary relationship types with cardinality types $(1,1) : 1$ or $(1,1) : N$ and without attributes are allowed. Each relationship type is oriented from the entity type with the arbitrary cardinality called the *parent participant* to the entity type with the cardinality $(1,1)$ in the relationship type called the *child participant*. We say that the child participant is *nested* in the parent participant. This kind of relationship types may be called *nesting binary relationship types*. When modeling XML data, the nesting binary relationship types are represented by a nesting of elements on the XML logical level. They express a hierarchical structure on the XML logical level explicitly on the conceptual level. However, the semantics of nesting relationship types do not have to be only "part-of". It may be a general association too.

Such restrictions are too strong and do not allow to model conceptual schemata with richer semantics. Nor n-ary relationship types, nor attributes of relationship types can be modeled. Moreover, lots of redundancies may appear in schemata. There are some approaches extending this basic hierarchical model described in the following subsections.

4.2 ORA-SS

ORA-SS is a rich hierarchical conceptual model for XML proposed by Dobbie et al. in [4]. ORA-SS has three basic modeling constructs: object types, relationship types and attributes. The object type construct is similar to the entity type from the E-R model. Relationship types between object types represent hierarchical relationships. Non-hierarchical relationships can be modeled by the references. The authors introduce the concept of n-ary hierarchical relationship types and attributes of hierarchial relationship types. Moreover, the authors offer the following extending features:

- *Cardinality constraints* for the both participants of hierarchical relationship types.
- An *ordering* on different concepts. The first type is an ordering on values of a multivalued attribute of an object type. The second type is an ordering between the attributes of an object type and nesting relationship types going from the object type. The third type is an ordering on a relationship type going from an object type. It allows to specify ordering between the objects nested by the ordered relationship type in the parent object.
- A *disjunction* between two or more attributes or nesting relationship types. It allows to model irregular structure.

Figure 4 displays an ORA-SS schema representing professors as employees of departments and professors as members of projects. Each professor is employed by exactly one department and each department employes one or more professors. Each professor is a member of zero or more projects and each project has one or more members. There is a ternary relationship type *AuthorOf* between the object types *Project*, *Professor* and *Paper*. It represents papers written by a professor participating in a project. Each professor is an author of zero or more papers in a project and each paper has one or more authors. For each project there is a list of member professors and for each member professor there is a list of papers he wrote during his work in the project. Moreover, there is the attribute *pages* of the relationship type *AuthorOf*. For a professor being an author of a paper in a project the value of *pages* is the number of pages the professor wrote in the paper. However, the attribute *pages* cannot be directly assigned to the relationship type *AuthorOf*. It must be assigned to the nested object type *Paper*. For each *Professor* instance nested in a *Project* instance there must be a *Professor* instance nested in a *Department* instance containing the *name* and *address* values of the professor. This is modeled by the reference between the object types.

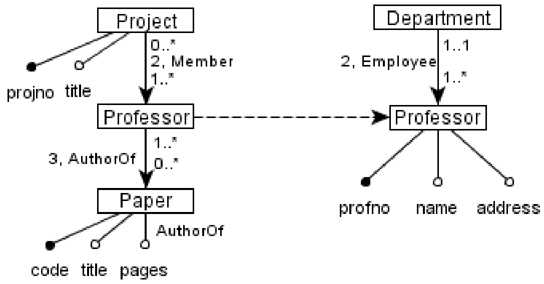


Fig. 4. ORA-SS Diagram

4.3 Semantic Networks for XML

The semantic network model for XML was introduced by Feng et al. in [6]. The model is a little extension to the basic hierarchical conceptual model described in Section 4.1. Schemata in the semantic network model for XML are called semantic networks. Nodes in semantic networks are used for modeling objects from the real world and their attributes, and edges are used for modeling relationships between the objects.

Only binary hierarchical relationship types without attributes can be modeled in the semantic network model. Moreover the parent participant cardinality constraint of a hierarchical relationship type must be equal to $(1, 1)$. Beside the hierarchical relationship types it is possible to use non-hierarchical relationship types for modeling associations.

Different constraints can be specified in the semantic network schema for XML. Constraints can be specified over a node, over an edge and over a set of edges. These constraints are a uniqueness, order, disjunction, etc.

Figure 5 displays a semantic network schema. There are departments represented by the node *Department* and professors in the departments represented by the node $Professor$. The content of the node $Professor$ is ordered. For each professor there are the papers he wrote represented by the node *Paper*. Each paper may be composed of chapters or sections, but not both (the exclusive constraint). The courses offered by a department are represented by the node *Course*. $Professor$ is associated with *Course*. It represents the relationships between a professor and the courses he teaches.

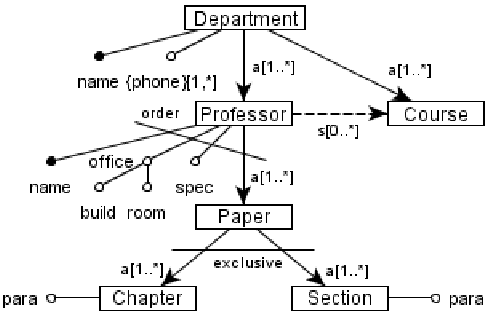


Fig. 5. Semantic Network Diagram

5 Comparison of Described Conceptual Models

In this section, we compare the conceptual models mentioned in this paper. The comparison is made against the general requirements and the modeling constructs requirements introduced at the beginning of the paper.

There are two comparative tables. Table 1 compares the models against the general requirements and Table 2 compares the models against the modeling constructs requirements. The well-known E-R model and the basic hierarchical model are compared too.

We are not able to decide, which of the previous two approaches (E-R extensions, hierarchical modeling) is better for the conceptual modeling of XML data. Conceptual models based on the E-R model allow user to create a schema with no metadata redundancy, but there is the problem with the modeling of the specific XML features. Hierarchical conceptual models solve the problem with a hierarchical structure of XML, but there arises problems such as data and metadata redundancy, modeling of attributes of relationship types, and modeling of n-ary relationship types.

There are requirements that are not met by the described models. The modeling of document centric data and the reuse of content is problematic. The important requirement on the integration of conceptual schemata is solved only by the ORA-SS model. None of the models solves the problem of the integration with the semantic web technologies.

Table 1. Comparison Against the General Requirements

ER	ER-B	EReX	EER	XER	ERX	C-XML	Hier	X-Entity	ORA-SS	Sem.net.
· independence on XML schema languages										
√	−	√	√	−	√	√	√	√	√	√
· formal foundations										
√	√	√	−	−	√	−	√	√	√	√
· graphical notation										
√	√	√	√	√	√	√	√	√	√	√
· logical level mapping										
− relational model										
√	√	−	−	−	−	−	−	−	−	−
− tree grammar based XML schema languages										
−	√	√	√	√ 1	−	√	√	√	√	√
− pattern based XML schema languages										
−	−	−	−	−	−	−	−	−	−	−
− utilization of hierarchical structure of XML										
−	−	√	√	√	−	√	√	√	√	√
· different structures on the logical level										
− conceptual hierarchical views										
−	−	−	−	−	−	−	−	−	√	−
− translation between hierarchical views										
−	−	−	−	−	−	−	−	−	√	−
· semantic web mapping										
−	−	−	−	−	−	−	−	−	−	−
1 formal description is missing										

Table 2. Comparison Against the Modeling Constructs Requirements

ER	ER-B	EReX	EER	XER	ERX	C-XML	Hier	X-Entity	ORA-SS	Sem.net.
· hierarchical relationship types										
–	–	–	√	–	√	–	√	√	√	√
– $M : N$ cardinality – N-ary – attributes										
–	–	–	–	–	–	–	–	–	√	–
· non-hierarchical relationship types										
√	√	√	√	√	√	√	–	–	√	√
– $M : N$ cardinality										
√	√	√	√	–	√	√	–	–	– 5	–
– N-ary – attributes										
√ 1	√	√	√	–	–	√	–	–	– 5	–
· ordering										
– on the values of an attribute										
√ 1	–	–	–	–	–	–	√	–	√	√
– on the content of a concept										
–	–	–	–	√ 2	–	√	–	–	√	√
– on the participant of a relationship type										
–	–	√	√	–	√ 3	–	–	–	√	–
· irregular and heterogeneous structure										
– variant-valued attribute constructor										
√ 1	√	–	–	–	–	–	√	–	√	√
– disjunctive constraints on relationship types										
–	√ 4	√ 4	–	–	√	–	–	√	√	√
· document-centric data										
– basic mixed content										
–	–	–	–	√	–	–	–	–	–	–
– generalized mixed content										
–	–	–	–	–	–	–	–	–	–	–
· reuse of content										
– IS-A or the category concept										
√	√	√	–	√	√	√	√	–	√	√
– named types and groups of concepts										
–	–	–	–	–	–	–	–	–	–	–
· integration of conceptual schemata										
– modeling constructs										
–	–	–	–	–	√	–	–	–	√	–
– algorithms for merging schemata										
–	–	–	–	–	–	–	–	–	√	–
– algorithms for the data translation between schemata										
–	–	–	–	–	–	–	–	–	√	–
1 with the complex attributes extension										
2 unordered content is restricted by the restrictions of xsd:all										
3 only by ordered attributes, native XML ordering is not utilized										
4 using the category concept										
5 indirect modeling using hierarchical relationship types is possible										

6 Conclusions

In this paper, we describe a state of the art of the conceptual modeling for XML. There has been several papers proposing new conceptual models for XML. We selected some representants of the models and describe them in this paper. We propose a list of requirements conceptual models for XML should satisfy and compare the mentioned conceptual models against the requirements.

The comparison of the models shows that there is a poor support for some specific XML features as ordering or mixed content as described by the modeling constructs requirements proposed in Section 2. Moreover, the models poorly concentrates on the usage of conceptual schemata for the data integration and the integration with the semantic web technologies as described by the general requirements proposed in Section 2.

For these reasons, there is an open space for a research in the area of the conceptual modeling for XML. Not only new modeling constructs should be proposed to support specific XML features. The utilization of the conceptual models for the data integration between different data sources including semantic web resources should be explored too.

References

1. A. Badia. Conceptual Modeling for Semistructured Data. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering Workshops (WISE 2002 Workshops), p. 170-177.* Singapore, December 2002.
2. J. Clark. *XSL Transformations (XSLT) Version 1.0.* World Wide Web Consortium, Recommendation REC-xslt-19991116. November 1999.
3. J. Clark, M. Makoto. *RELAX NG Specification.* Oasis. December 2001.
4. G. Dobbie, W. Xiaoying, T.W. Ling, M.L. Lee. ORA-SS: An Object-Relationship-Attribute Model for Semi-Structured Data. *Technical Report, Department of Computer Science, National University of Singapore.* December 2000
5. D.W. Embley, S.W. Liddle, R. Al-Kamha. Enterprise Modeling with Conceptual XML. In *Proceedings of the 23rd International Conference on Conceptual Modeling (ER 2004), p. 150-165.* Shanghai, China, November 2004.
6. L. Feng, E. Chang, T. Dillon. A Semantic Network-Based Design Methodology for XML Documents. *ACM Transactions on Information Systems, Volume 20, Number 4, p. 390-421.* October 2002.
7. D. C. Fallside, P. Walmsley. *XML Schema Part 0: Primer Second Edition.* World Wide Web Consortium, Recommendation REC-xmlschema-0-20041028. October 2004.
8. T. Halpin. *Information Modeling and Relational Databases From Conceptual Analysis to Logical Design.* Morgan Kaufmann Publishers, 2001. ISBN: 1-55860-672-6
9. International Organization for Standardization. *Information Technology Document Schema Definition Languages (DSDL) Part 3: Rule-based Validation Schematron.* ISO/IEC 19757-3, February 2005.
10. B.R. Loscio, A.C. Salgado, L.R. Galvao. Conceptual Modeling of XML Schemas. In *Proceedings of the Fifth ACM CIKM International Workshop on Web Information and Data Management (WIDM 2003), p. 102-105.* New Orleans, Louisiana, USA, November 2003.

11. M. Mani. EReX: A Conceptual Model for XML. In *Proceedings of the Second International XML Database Symposium (XSym 2004), p. 128-142.* Toronto, Canada, August 2004.
12. M. Mani, D. Lee, R.R.Muntz. Semantic Data Modeling Using XML Schemas. In *Proceedings of the 20th International Conference on Conceptual Modeling (ER 2001), p. 149-163.* Yokohama, Japan, November 2001.
13. F. Manola, E. Miller. *RDF Primer.* World Wide Web Consortium, Recommendation REC-rdf-primer-20040210. February 2004.
14. M. Necasky: Conceptual Modeling for XML: A Survey. Technical Report No. 2006-3, Dep. of Software Engineering, Faculty of Mathematics and Physics, Charles University, Prague, 2006, 54 p.
15. Object Management Group. *UML 2.0 Superstructure Specification.* October 2004.
16. G. Psaila. ERX: A Conceptual Model for XML Documents. In *Proceedings of the 2000 ACM Symposium on Applied Computing, p. 898-903.* Como, Italy, March 2000.
17. A. Sengupta, S. Mohan, R. Doshi. XER - Extensible Entity Relationship Modeling. In *Proceedings of the XML 2003 Conference, p. 140-154.* Philadelphia, USA, December 2003.
18. M. K. Smith, Ch. Welty, D. L. McGuinness. *OWL Web Ontology Language Guide.* World Wide Web Consortium, Recommendation REC-owl-guide-20040210. February 2004.
19. B. Thalheim. *Entity-Relationship Modeling: Foundations of Database Technology.* Springer Verlag, 2000, Berlin, Germany. ISBN: 3-540-65470-4