

# **XCASE - A Case Tool for Modeling XML data**

## **Motivation**

Recently, XML has been applied as a format for data representation on different levels of information systems. As the amount of data represented in XML grows, a need for methods as well as tools for modeling XML data grows as well. Recently, XML schema languages like XML Schema [5] are applied to describe XML data on the logical level. However, modeling XML data directly with these languages is a time-consuming and error-prone task as we discuss in detail in [1]. XML schemes are usually interrelated with each other and they are also related to other components in the system (such as relational databases). However, these interrelations are difficult to capture and maintain on the XML schema level.

As we show in [1], designing XML data first on the conceptual level could help to solve the problems of XML schema languages. For this, existing conceptual models such as the Entity-Relationship model (ER) or the model of UML class diagrams (UML) can be applied. However, several extensions to these basic models are required to cover the special features of XML data. There are fairly recent surveys of this area, concretely [2] (ER-based approaches) and [3] (UML-based approaches).

## **Project Goals**

In this software project, we will create a case tool for conceptual modeling of XML data. We will utilize a conceptual model for XML data called XSEM [1] which was developed at KSI MFF UK. XSEM follows Model-Driven Architecture (MDA) [4]. Recently, MDA has been successfully applied for software development. It is based on application of models to raise the level of abstraction at which developers create and evolve software and can also be applied to data modeling.

In XSEM, two types of models called Platform-Independent Model (PIM) and Platform-Specific Model (PSM) are considered. First, a schema in PIM is designed. It is a conceptual schema that abstracts from any implementation details specific for XML (XML elements vs. attributes, hierarchical and irregular structure, ordering, ...). These details are specified by one or more PSM schemes. A PSM schema specifies how the concepts modeled by the PIM schema are represented in a given type of XML documents. For each intended type of XML documents, a separate PSM schema is designed. PSM schemes can share the same concepts from the PIM schema and can represent them in an arbitrary XML structures.

The main goal of the project is to examine possibilities of XSEM and conceptual modeling for XML in general. The tool is original, because neither a conceptual model for XML nor a tool for conceptual modeling of XML has been developed neither in commercial nor academic conceptual modeling community to our best knowledge. There are some efforts such as Sparx Systems Enterprise Architect [6] that provide constructs for modeling XML data in UML class diagrams. However, the offered possibilities are limited as we discuss in [1].

## Tool Description in General

The tool will enable to model XML data on PIM and PSM level as specified by XSEM. It will offer a standard environment commonly offered by other tools for conceptual modeling such as ER or UML class diagram editors. The tool will enable to model the data on the following two levels:

- Platform-Independent Model Level
  - data is modeled independently of its representations in different types of XML documents (i.e. a conceptual schema of the problem domain is created)
  - UML class diagrams are applied for modeling
    - packages, classes, class attributes, stereotypes, (n-ary) associations
- Platform-Specific Model Level
  - modeling how the data is represented in different types of XML documents
  - each PSM diagram models one type of XML documents
  - working with components of PIM diagrams
    - organizing them to hierarchical structure
    - using extended constructs proposed in [1] to model how the data is represented in more complex XML structures (XML elements vs. XML attributes, irregular structure, mixing structured and unstructured data, ordering, ...)
  - automatic translation of PSM diagrams to representations in XML Schema

## Tool features

In addition to common features provided by UML class diagram editors, the tool will concentrate on XSEM specific features. This covers mainly the transition between PIM and PSM diagrams. This transition can not be performed automatically as we show in [1]. The following will be supported by the tool:

- semiautomatic guided transition from PIM diagrams to PSM diagrams
- more PSM diagrams can be derived from the same part of one or more PIM diagrams
  - user specifies which components of one or more PIM diagrams should be translated to a PSM diagram
  - default PSM diagram is created by the tool automatically
  - user edits the PSM diagram to model the required XML representation of the selected components (a set of operations will be provided for editing)
- changes in PIM diagrams are propagated to the corresponding PSM diagrams and, conversely, changes in PSM diagrams are propagated back to appropriate PIM diagrams dynamically to ensure the consistency between PIM and PSM diagrams
- organization of diagrams and related files within projects

# Platform and Third-Party Libraries

Windows Vista / XP  
Microsoft C#/.NET

A library implementing OMG UML class diagrams specification will be utilized.

## Team

Supervisor: Mgr. Martin Nečaský

Team members:

Jakub Klímek  
Lukáš Kopenec  
Lucie Kučerová  
Jakub Malý  
Kateřina Opočenská

## References

- [1] M. Necasky: Conceptual Modeling for XML. Ph.D. thesis. Faculty of Mathematics and Physics, Charles University, Prague. May 2008.  
<http://www.necasky.net/thesis.pdf>
- [2] M. Necasky: Conceptual Modeling for XML: A Survey. TR 2006-3, Dpt. of Software Engineering, Faculty of Mathematics and Physics, Charles University, Prague. January 2006.  
<http://www.necasky.net/papers/tr2006.pdf>
- [3] E. Dominguez, J. Lloret, B. Perez, A. Rodriguez, A. L. Rubio, M A. Zapata: A Survey of UML Models to XML Transformations. In Proceeding of Web Information Systems Engineering (WISE 2007). LNCS, Vol. 4831, pp. 184-195, Springer, 2007.
- [4] J. Miller, J. Mukerji. MDA Guide Version 1.0.1. OMG. 2003.
- [5] W3C, XML Schema Part 0: Primer Second Edition, October 2004,  
<http://www.w3.org/TR/xmlschema-0/>
- [6] <http://www.sparxsystems.com.au/>