Evolution of XML Applications

NICTA, Sydney, Australia

Irena Mlynkova

XML and Web Engineering Research Group

Department of Software Engineering
Faculty of Mathematics and Physics
Charles University in Prague
Czech Republic

mlynkova@ksi.mff.cuni.cz

23.10. 2011
XML and Web Engineering Research Group

- 4 researchers + 5 PhD students
- Current areas:
  - Modelling of XML applications and systems
  - Evolution and change management of applications
  - Inference of XML schemas
  - Similarity of XML data
  - Linked Data
  - Data provenance
  - Statistical analysis of real-world data
- Older topics:
  - XML benchmarking
  - Storage strategies for XML data

http://www.ksi.mff.cuni.cz/xrg/
Evolution of XML Applications

- **XML application** = an application that exploits XML technologies
  - XML data, XML schemas (DTD, XSD, Schematron, RELAX NG), XML queries (XPath, XQuery), XSLT scripts, ...

- **Evolution of application**
  - User requirements or surrounding environment changes => the application needs to adapt accordingly
  - Key problem: changes in data structure
    - Influence further processing
    - Our focus
    - Many related issues to be solved...
Preliminary Approaches

- They solve separate issues
- An XML application can be much complex
XML System of Applications

- An XML application usually involves a family of XML schemas
  - Each involved in particular execution part
  => XML system of applications

- Schemas are related, overlap, exploit each other, ...

- Problems:
  - The amount of schemas can be high
    - Tens, hundreds, ...
  - A change in a single schema may need to be propagated to numerous other schemas
  => Difficult, error-prone, ...
Example

- Common problem domain: purchasing goods

```xml
<custList version="1.3">
  <cust>
    <name>Martin Necasky</name>
    <address>Vaclavske nam. 123, Prague</address>
    <phone>123 456 789</phone>
  </cust>
  <cust>
    <name>Department of Software Engineering, Charles University</name>
    <hq>Malostranske nam. 25, Prague</hq>
    <storage>Ke Karlovu 3, Prague</storage>
    <secretary>Ke Karlovu 5, Prague</secretary>
    <phone>111 222 333</phone>
  </cust>
</custList>

<purchaseRQ version="1.0">
  <bill-to>Malostranske nam. 25, Prague</bill-to>
  <ship-to>Ke Karlovu 3, Prague</ship-to>
  <cust>
    <name>Department of Software Engineering, Charles University</name>
    <email>ksi@mff.cuni.cz</email>
  </cust>
  <items>
    <item>
      <code>P045</code>
    </item>
    <item>
      <code>P332</code>
    </item>
  </items>
</purchaseRQ>

- And other schemas
  - Customer details, purchase responses, purchase transport details,...
New user requirement: Address is not a simple string, but should be divided into street, city, zip code, ...

We need to:
- Find all the addresses in all the schemas
- Correctly modify their structure
- Correctly propagate the modification to all respective documents

The situation can be even more complicated...
- E.g. We may want to change only addresses that represent a place to ship the goods, not all addresses
  - Not, e.g., the address of the headquarters
- And so far we consider only data...
Our Solution

Five-Level XML Evolution Framework

- Based on the MDA principle
  - Idea: modelling of problem domain at different levels of abstraction

- Levels:
  - Extensional level – XML documents
  - Operational level – XML queries
  - Schema level – XML schemas
    - DTD, XSD, Schematron, RELAX NG
  - Platform-independent level – conceptual model of the whole problem domain
    - e.g. purchasing goods
  - Platform-specific level – mapping of the problem domain to particular XML formats
    - e.g. list of customers, purchase requests, ...
Platform-Independent Model (PIM)

- Models real-world concepts + relationships among them
- Simplified UML class diagram
  - Classes, attributes + data types
  - Binary associations + cardinalities
- Simplification: easier and shorter description of ideas
  - Can be further extended – future work
Platform-Specific Model (PSM)

- How part of reality modelled by PIM is represented in an XML schema
  - “Glue” between PIM and XML schemas
- UML class diagrams extended for the purposes of XML modelling
  - Hierarchical structure
  - XML elements vs. attributes
  - Order
  - Specific XML schema constructs...

Two Perspectives of PSM

- **Conceptual:**
  - Classes, attributes and associations of PSM are mapped to classes, attributes and associations of PIM
  - The mapping specifies the semantics of PSM schema in terms of PIM schema

- **Grammatical:**
  - A PSM schema models an XML format = an XML schema = a regular tree grammar
  - A regular tree grammar can be translated to a PSM schema and vice versa
    - Multiple possible translations from PSM to XML schema
How the Evolution Process Works?

But there are many related problems to be solved...
“Background” Issues

- Definition of PIM
  - Constructs
  - Operations
    - Atomic vs. composite

- Definition of PSM
  - Constructs
  - Operations
    - Atomic vs. composite

- Definition of mapping between PIM and PSM, PSM and XML schema

- Definition of propagation of each operation

In current literature mostly omitted...
Formal Definition of Models and Mapping

- Algorithm for translating PSM to regular tree grammars and vice versa
  - Proof of correctness and expressive power

- Definition of interpretation = formal definition of mapping between conceptual levels
  - Proof of correctness

Formal Definition of Models and Mapping

- Note:
  - DTD <=> local-tree grammars
  - XSD <=> single-type tree grammars
  - RELAX NG <=> regular tree grammars

- Note:
  - PSM schema can describe multiple XML schemas
    - Inheritance, references, ...
    - Lots of “syntactic sugar”

- Side results:
  - Algorithm for PSM schema normalization and optimization
Edit Operations and Their Propagation

- Definition of a set of **atomic operations** at PSM and PIM levels
  - Proof of minimality, correctness

- Specification of **propagation** mechanism
  - Correctness

- Definition of **composition** of edit operations
  - Atomic ones are too simple, just for formal reasons

---

Our Edit Operations

- Creation, update, removal – classical operations
- New atomic operation: synchronization
  - Two sets of schema components are semantically equivalent
- Idea:
  - Requirement: Address is not a simple string, but should be divided into street, city, zip code, ...
  - Existing approaches: creating new attributes + removing the old one
    => loss of information
  - Our approach: synchronization
    => the propagation mechanism knows where to “get” data
Re-Validation of XML Documents

- PSM diagram = XML schema changes => XML documents need to be adapted
  - Re-validation (XSLT scripts)
  - User interaction

---


Propagation to Operational Level

- Schema evolves => query inconsistency
- Related work: a single paper describing rules for backward-compatible queries
- We need: a model of queries, a mapping between data and queries, edit operations, propagation mechanism, ...


Modelling Strategies

- **Top-Down** (forward engineering) – modelling a new system (or its modification)
  1. Creating PIM schema
  2. Creating a PSM schema from PIM
     - Or its modification
  3. Translating PSM schema to XML schema
  4. Instantiating the schema

- **Bottom-Up** (reverse engineering) – an existing system + integration of an existing XML schema
  1. Translating XML schema to PSM
  2. Mapping of PSM schema to and PIM schema
     - We need a suitable similarity measure

  If we do not have a schema, we need an *inference* method
Reverse Engineering: Inference of XML Schemas

Currently:
- Heuristic vs. grammar-based approaches
- Exploit purely XML documents

Our proposals:
- Other input data
  - Old schema, queries, user interaction
  - Other output information
  - XSD, ICs
- Optimization of the inference process


And many other...
jInfer

- A framework for XML schema inference
- Based on the idea of plug-ins
  - Modules
  - Can be added, removed, replaced, compared, ...
- “Playground” for proposing new inference methods
  - Not interesting features are implemented
    - Transforming XML data to grammar rules
    - Transforming automata to XML schemas
    - ...
  - Current know approaches are implemented
    - New ones can be compared

http://jinfer.sourceforge.net/
Reverse Engineering: Mapping of PSM to PIM

- Current approaches: Analyze structure, context, semantics, operations, ...

- Our aim:
  - To adapt verified techniques
  - To minimize user interaction

Other Data Formats

An XML system can contain other than XML data => the ideas can be generalized
- Relational models, UML class models, ...
- Business process models?

The idea can be extended
Implementations

- **XCase** ([http://xcase.codeplex.com/](http://xcase.codeplex.com/))
  - Implementation of first ideas
  - Desktop application
  - XML data levels and propagation among them

- **eXolutio** ([http://exolutio.com/](http://exolutio.com/))
  - Web application of XCase
  - Automatic generation of mappings between PIM and PSM

  - A general framework for any kind of data (not only XML)
    - New data format => new plug-in
  - Currently: XML, relational, UML
eXolutio
Future Work

- More complex conceptual-modelling constructs
  - e.g. inheritance, n-ary relationships
- Queries, integrity constraints
  - Modelling, change propagation, ...
- Other data formats
  - Relational data, objects, ...
  - Formal specification, edit operations, propagation, ...
- Storage strategies
  - New dimension: schema evolves => storage strategy should be optimized
- Extension to business process models
  - New dimension: not data structures, but “what happens with them”
  - Question: Is it possible to integrate it? How?