Current Trends in Testing XMLMSs

Irena Mlynkova

irena.mlynkova@mff.cuni.cz

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

Introduction (1)

- **XML = a standard for data representation and manipulation**
  - **Huge amount of XML processing tools / XML management systems (XMLMS)**
    - **Parsing, validating, storing, querying, transforming, updating, exchanging, compressing, …**

⇒ **Questions / tasks arise:**

  - **User: the most sufficient XMLMS for my application**
  - **Vendor: testing correctness / performance of my XMLMS, comparison with competing SW**
  - **Analyst: comparison of various aspects of existing XMLMSs from different points of view**

Introduction (2)

- Solution: to find results of appropriate analysis
  - A huge amount of related papers exists
- Problems:
  - Development of XMLMSs is fast ⇒ results soon become obsolete
  - We find reasonably up-to-date analytical results ⇒ however the testing scenarios usually do not fit well to all our use cases

⇒ Mostly we need to prepare own testing scenarios that represent our particular application

Goals of this presentation

- **Overview of possibilities how to acquire / prepare XML testing scenarios**
  - **Conformance test suites, repositories of real-world XML data, XML benchmarking projects, data generators, …**
- **Limitations of the current approaches**
- **Overview of the key findings and related recommendations**

- **Purpose?**
  - **Useful source of information**
  - **Based on throughout analysis done for current research**
    - **Proposal of a comprehensive benchmark**

Content

1. Overview and classification of existing approaches
2. Key findings and recommendations
3. Conclusion

What is a Benchmark?

- **A set of testing scenarios / test cases = data + related operations**
  - **Empty (apparent) set of operations: compression**
- **Enables to compare versatility / efficiency / behavior of system(s) under test (SUT)**
- **Our case:**
  - **Data = XML documents**
    - **With / without XML schemes**
  - **Operations = any kind of XML-related data operations**

Classification of Existing Methods

- Type of data
  - Real-world vs. synthetic
    - Realistic, but too simple, contain errors
  - Fixed vs. dynamic data sets/operations
- Type of operation
  - Parsing, validating, querying, updating, transforming, …
- Tested technology
  - DTD vs. XML Schema, XPath vs. XQuery, XPath 1.0 vs. XPath 2.0, …

XML Data Sets

- Typical approach: fixed sets of (real-world) XML data
  - Rather interesting than useful
    - The Bible in XML, Shakespeare's plays, …
  - XML exports of databases – most common
    - *IMDb* (movies and actors), *DBLP* (scientific papers), *Medical Subject Headings* (medical terms), …
  - Repositories of real-world XML – some not originally in XML format
    - *INEX*, *Ibiblio, …*
  - Special real-world XML data – uncommon structure
    - Protein sequences, RNAs, astronomical NASA data, linguistic trees, …
- Problem: Mostly simple, without respective operations

Data Generators

- Solution: to generate synthetic testing data sets
- Schema-unaware generators
  - General structural parameters
    - Depth of XML tree, numbers of subelements, …
- Template-based generators
  - Input: annotated XML schema
    - Schema = precise description of structure
    - Annotations = more specific information
      - e.g. distributions of occurrences of attributes/lengths of texts, …
- Aim: to generate as realistic structure as possible
  - Zip’s law, Markov chains, statistical distributions, …
- (Dis)advantage: huge amount of parameters

Parsing and Validating (1)

- Primary application for XML data processing
- W3C: XML Conformance Test Suites
  - XML 1.0, XML 1.1 and Namespaces in XML 1.1
  - 2.000 XML documents
  - Binary tests:
    - Parser must accept/reject the document correctly
  - Output tests:
    - Parser must report information as required

Parsing and Validating (2)

- Types of parsers
  - Event-driven – while reading they return data fragments
    - Push – reading cannot be influenced
    - Pull – read the next data only if they are “asked” to
  - Object-model – read the document and built it completely in memory
  - Various combinations

⇒ Number of papers which evaluate efficiency of subsets of known implementations

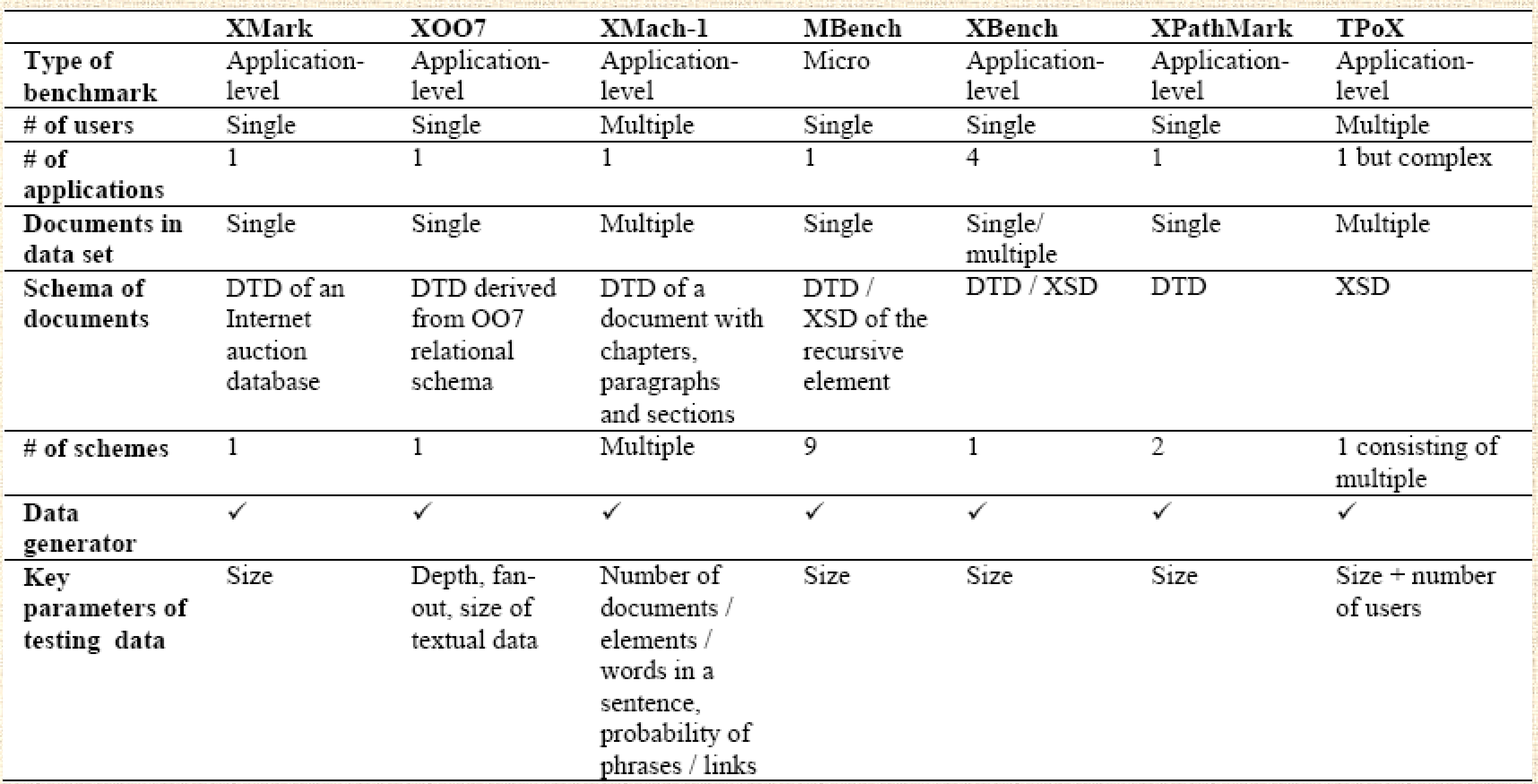
  - Compare same / different types of parsers
  - All the related data are available

- Problem: No true benchmarking project for parsers / validators

Querying (1)

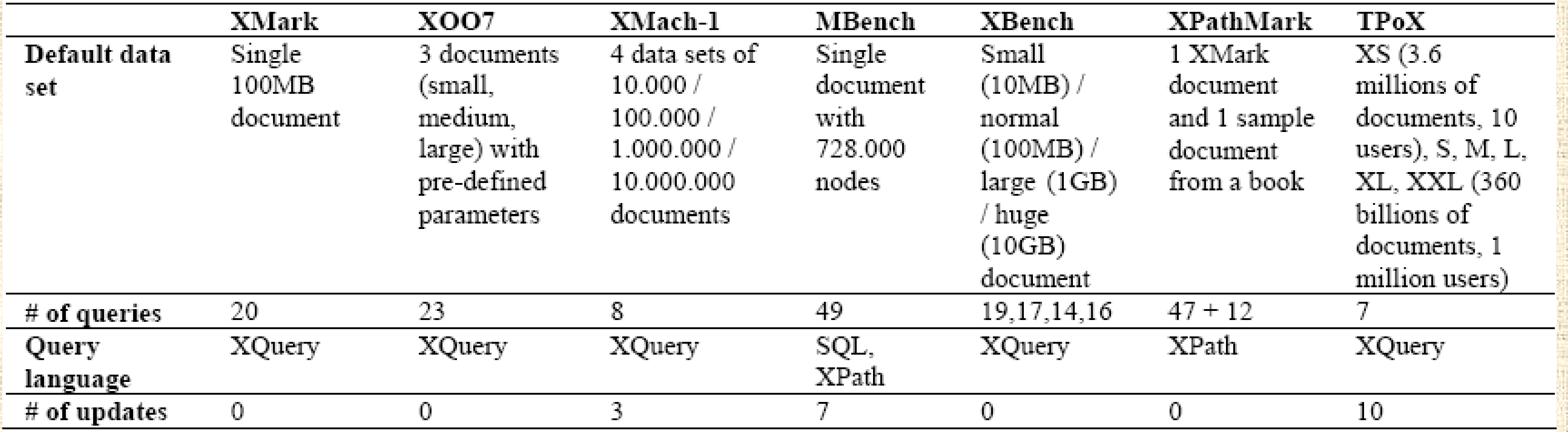
- W3C:
  - XML Query Use Cases – not a benchmark, a set of examples of XML query applications
  - XML Query Test Suite – 15.000 test cases (queries and expected results), tests support of XML Query constructs
- The biggest set of true benchmarks
- Test the amount of supported query constructs + efficiency of evaluation
  - Assumption: correct results ⇒ not tested
- Best known representatives: XMark, XOO7, XMach-1, MBench, XBench, XPathMark, TPoX

Querying (2)



	XMark	XOO7	XMach-1	MBench	XBench	XPathMark	TPoX
Type of benchmark	Application-level	Application-level	Application-level	Micro	Application-level	Application-level	Application-level
# of users	Single	Single	Multiple	Single	Single	Single	Multiple
# of applications	1	1	1	1	4	1	1 but complex
Documents in data set	Single	Single	Multiple	Single	Single/ multiple	Single	Multiple
Schema of documents	DTD of an Internet auction database	DTD derived from OO7 relational schema	DTD of a document with chapters, paragraphs and sections	DTD / XSD of the recursive element	DTD / XSD	DTD	XSD
# of schemes	1	1	Multiple	9	1	2	1 consisting of multiple
Data generator	✓	✓	✓	✓	✓	✓	✓
Key parameters of testing data	Size	Depth, fan-out, size of textual data	Number of documents / elements / words in a sentence, probability of phrases / links	Size	Size	Size	Size + number of users

Querying (3)



	XMark	XOO7	XMach-1	MBench	XBench	XPathMark	TPoX
Default data set	Single 100MB document	3 documents (small, medium, large) with pre-defined parameters	4 data sets of 10.000 / 100.000 / 1.000.000 / 10.000.000 documents	Single document with 728.000 nodes	Small (10MB) / normal (100MB) / large (1GB) / huge (10GB) document	1 XMark document and 1 sample document from a book	XS (3.6 millions of documents, 10 users), S, M, L, XL, XXL (360 billions of documents, 1 million users)
# of queries	20	23	8	49	19,17,14,16	47 + 12	7
Query language	XQuery	XQuery	XQuery	SQL, XPath	XQuery	XPath	XQuery
# of updates	0	0	3	7	0	0	10

- **Type of benchmark:**
  - **Application-level – compare and contrast distinct applications ⇒ queries are highly different**
  - **Micro – evaluate performance of a single system in distinct situations ⇒ similar queries, differentiate, e.g., in selectivity**
    - **MBench**

Querying (4)

- **Purpose of benchmark:**
  - **Number of users, applications, documents**
  - **Most: single-user, single-application, with single document**
    - **XBench – 4 classes of XML applications**
      - **Text-centric/single document, data-centric/multiple documents, …**
    - **XMach-1, TPoX – multi-user, test other XML management aspects**
      - **Indexing, schema validation, concurrency control, transaction processing, network characteristics, …**
- **Data sets:**
  - **All projects involve DTD/XSD and a simple data generator**
    - **Typical parameter: size of data**
- **Operations:**
  - **All projects involve a set of XQuery queries**
  - **XMach-1, MBench, TPoX – involve update operations**
  - **XMach-1, TPoX (multi-user benchmarks) ⇒ additional, less XML-like operations**

Querying (5)

- Analysis of benchmarks
  - Only 1/3 of papers use a kind of benchmark
  - 38% of benchmark queries are incorrect/out-dated
  - XMark – most popular, simple ⇒ users do not want to bother with complex application
- Benchmark repository
  - Observation: A fixed set of queries ⇒ cannot test various aspects of applications

  ⇒ MemBeR repository of micro-benchmarks
    - New micro-benchmark/new result set must be specified as an XML document

Other XML Technologies

- Basic: parsing, validating, querying
- Advanced: transformations, compressing, … ⇒ need for special purpose benchmarks
  - Problem: low number, representatives are obsolete
- Example 1: XSLT
  - XSLTMark – from 2000, not maintained, constructs of version 1.0 (from 1999, obsolete)
  - Analyses of implementations use XSLTMark
- Example 2: XML update
  - New technology, not much supported
  - First proposal of a benchmark from 2008

Content

1. Overview and classification of existing approaches
2. **Key findings and recommendations**
3. Conclusion

Summary (1)

1. The most typical source of testing XML data: repositories of fixed, real-world XML data
   - Realistic, but too simple, without respective operations.
2. Solution of simplicity: generators of synthetic XML data
   - Precise specification of structure of data
   - Require a skilled user
3. Key XML operations: parsing and validating
   - W3C: conformance test suites
   - Well covered
4. Key users’ interest in parsing: efficiency / space overhead
   - Several analytical papers and projects dealing with it
   - No true test suite covering key aspects, bottlenecks, …

Summary (2)

5. Second key XML operation: querying
   - W3C: XML Query Test Suite, XML Query Use Cases
     - Enable to test the full support of language
     - Provide a set of typical application of XML querying
   - Several well-known and verified benchmarking projects
     - Different purposes, features, advantages, …
   - Area is wide, well covered
6. All query benchmarks involve a data generator, however the most popular are of simple usage
   - XMark: Only few parameters to specify ⇒ users do not want to bother with complex applications
7. Other XML technologies require special benchmark projects
   - The amount of respective benchmarks is low

Content

1. Overview and classification of existing approaches
2. Key findings and recommendations
3. **Conclusion**

Conclusion

- General observations:
  - Basic XML data operations are well covered with test suites and benchmarking projects
    - Parsing, validating, querying
  - Other XML technologies = problem
    - We can always exploit either real-world / synthetic XML data + hand-made operations
    - Advantage: wide research areas of proposing special-purpose benchmarking projects, performing respective analyses of existing implementations, …

Thank you