Analyzer:
A Framework for File Analysis

Martin Svoboda
Jakub Starka
Jan Sochna
Jiri Schejbal
Irena Mlynkova

analyzer.contact@gmail.com
http://urtax.ms.mff.cuni.cz/anaxml/

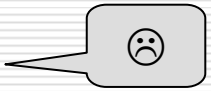
Department of Software Engineering
Faculty of Mathematics and Physics
Charles University
Prague, Czech Republic

Motivation

- Classical optimization strategy:
  - Exploitation of results of statistical analyses of real-world data
  - Efficient implementation of constructs that are used in real-world data most often
    - Observation: real-world data are usually simple
- Problems of real-world data:
  - Often change → many versions, findings are soon obsolete
  - Imprecise – do not fully follow recommendations, ..
  - Number of errors (typos, well-formedness, validity, …)
  - …

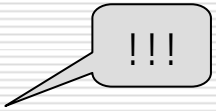
State of the Art

- Current analyses of real-world XML data:
  - Structure and complexity of DTDs (3 papers from 2001 – 2002)
  - Structure of XML documents (1 paper from 2003)
  - Comparison of XSDs and DTDs (2 papers from 2004)
  - Comparison of XML documents and XSDs (1 paper 2006)
- Usage:
  - XML data indexing
    - Average depth of XML documents < 8
  - Inference of XML schemas
    - DTD regular expressions are simple, form identifiable classes of languages
  - …
- Problem: No SW, just results



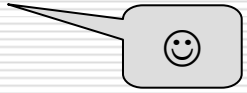
What do we need?

1. To gather a representative set of real-world data easily and automatically
   - Crawler + filters
2. To cope with errors
   A. Discard the incorrect data
      - Lost of large portion of data
   B. Provide a kind of corrector
3. To perform the analyses
   - Extensible, comparable, repeatable
4. To visualize and analyze the results
   - Huge amount of information → categorization, statistics, comparison, …



Analyzer

- http://urtax.ms.mff.cuni.cz/anaxml/
- General framework
  - Easy management of files
  - Configuration and execution of selected analyses
  - Advanced GUI for browsing generated reports
  - Extensibility



- Plugins
  - The basis of architecture
  - Users can create own plugins designed for a particular research intents
    - Initial motivation: Analysis of XML data
    - Current status: XML data analysis is a sample use case
      - XML documents, XML schemas, XPath/XQuery queries

Life Cycle of an Analysis

1. New project + configuration
   - Project = documents + analyzes + results
2. Selection of applied analyses
   - Subset of supported plugins
3. Insertion of data/documents to the project
   - Support for multiple versions
4. Computation of analyses over documents
5. Selection and configuration of collections
6. Assignment of data into collections
   - Clustering, classification
7. Computation of reports over collections

Data Insertion

- Import
  - A user has the data e.g. on hard or optical drives
- Download
  - Exploitation of a crawler
    - Download of a specified set of files
    - Initial address to start crawling from + parameters
      - Depth, file types, …
- Link searching
  - Current research interest
  - Motivation: XML schemas can refer other schemas (include, import), XML documents refer schemas now located anywhere else (typical error), XQuery queries refer documents now located anywhere else, …
    - Advanced crawler for XML data

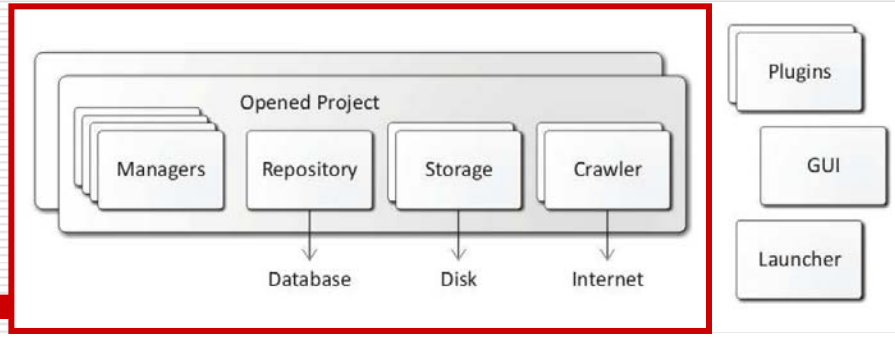
XML Data Insertion

- Errors in documents, schemas
  - Well-formedness (HTML → XHTML)
  - Validity
    - Should we re-validate?
    - Should we re-validate data or schemas?
    - Isn't it a kind of statistics?
- Crawling of XML operations
  - XSLT style sheets, XPath queries, XQuery queries, …
  - Currently no analysis of real-world XML operations
    - Problem:
      - Hard to find the operations
      - Hard to find the related data
        - Link searching

XML Data Analyses

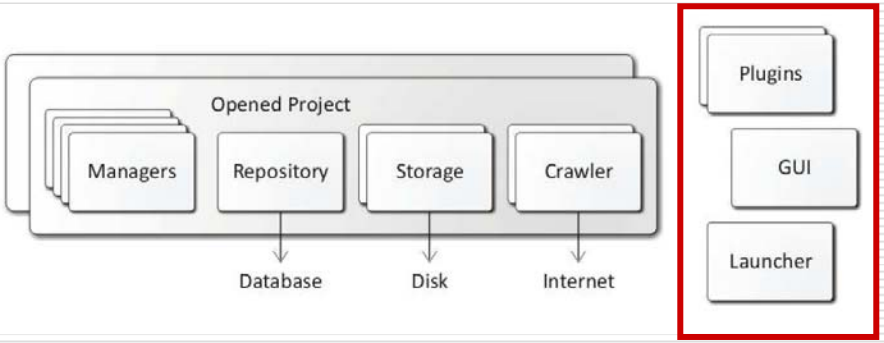
- XML data + schemas
  - Basic – usage/number/position/complexity of constructs
    - Elements, attributes, text nodes, user-defined types, substitution, …
  - Advanced – mutual comparison of results (data vs. schemas, DTD vs. XML Schema, XML Schema vs. Relax NG, …), pattern matching, similarity matching, …
- XML data + operations
  - Basic – usage/number/position/complexity of constructs
    - XPath axes, predicates, sub-queries, user-defined functions, operators, …
  - Advanced – selectivity, mutual comparison (XPath vs. XQuery, XQuery vs. XSLT, …)

Implementation



- Java 6 + NetBeans 6.8
- Project components – exclusive for each project
  - Storages – analyzed data
    - Native file system
  - Repositories – computed results, metadata, …
    - MySQL, Apache Derby, H2
  - Crawlers
    - Egothor
  - Managers – creating/editing/processing of documents, collections, reports, …
  - Can easily be replaced/extended
    - Not necessary

Implementation



- Shared components – common for all opened projects
  - Launcher – executes tasks over projects
    - Task = download of a document, computation of statistics over a document, aggregation of reports over documents, …
  - GUI
  - Plugins = Java classes with particular interface and conditions
    - Detector (checking of structure), tracer (searching of links), corrector (error correction), analyzer (analysis), collector (categorization), provider (providing results), performer (reports), viewer (visualization)

Current Plugins

- Current status:
  - Fully implemented framework
  - Support for user-defined plugins of any kind
  - Support for basic XML data analysis
- Plugins
  - Data + schemas: Number/distribution/complexity
    - Elements, attributes, text nodes, ...
  - DTDs, XSDs: Number/distribution/complexity of XML Schema constructs
  - XQuery/XPath: Occurrence of constructs
    - FLWOR, path expressions, constructors, …

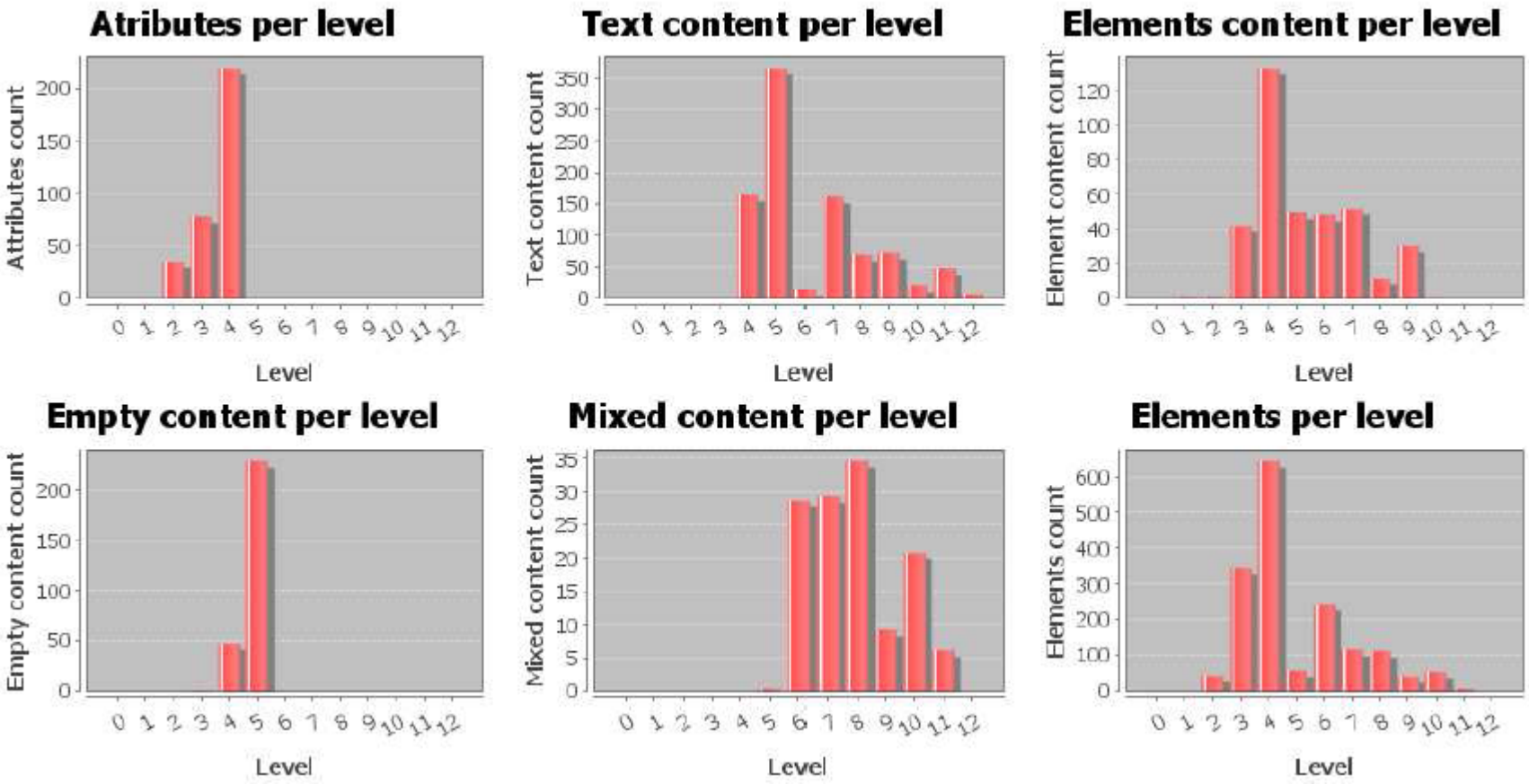
Preliminary Experiments with Analyzer

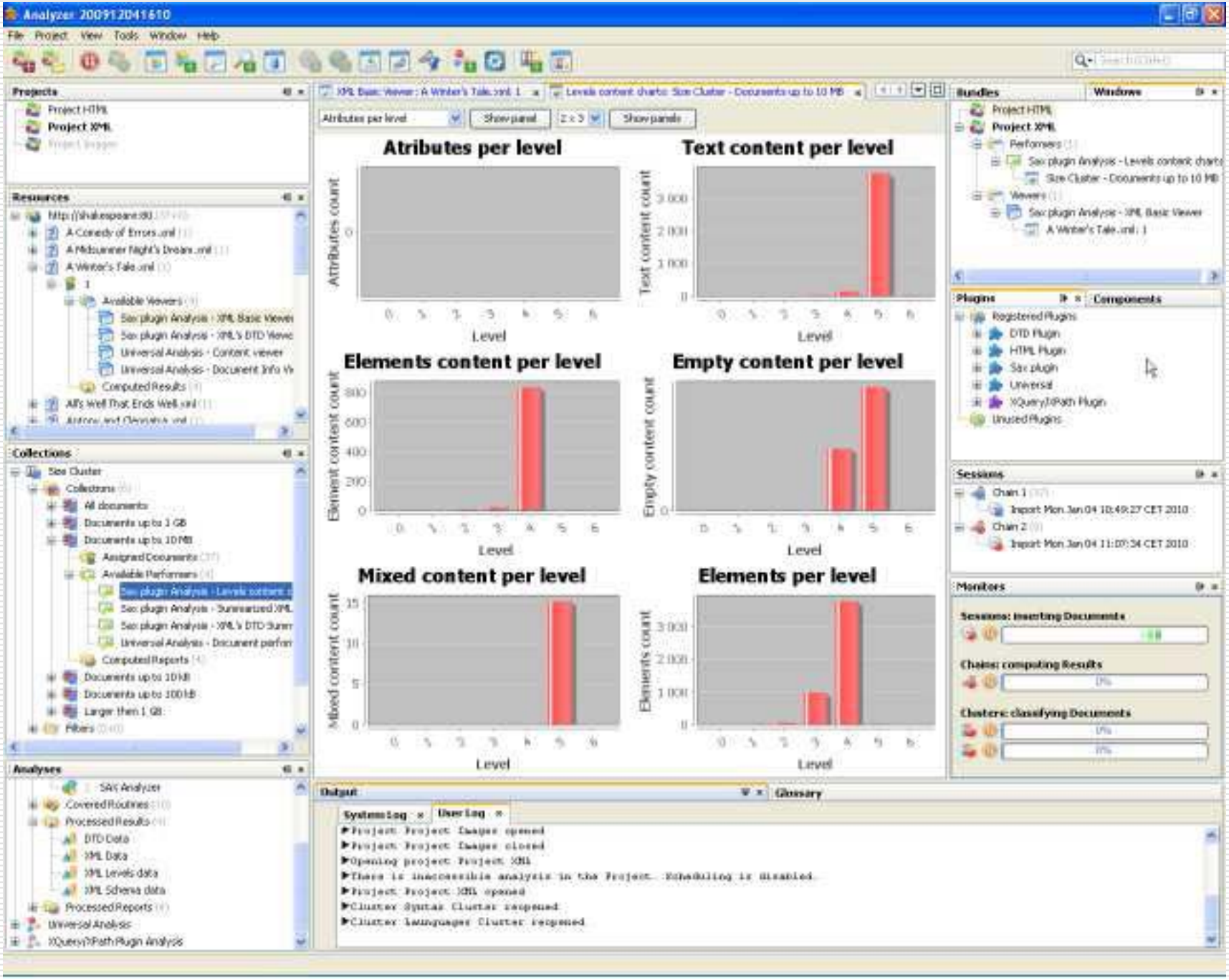
- PC with Intel Core 2 Quad Q9550, 2.83 Ghz processor, 4 GB RAM, Gentoo Linux 10.1
- Computation speed
  - Size of documents vs. types of repositories

Set name	Document count and size	Repository database	Document import	Result computation	Collection filling	Report computation
A	1,000 x 100 kB	Derby	7 s	60 s	14 s	12 s
		H2 DB	2 s	12 s	6 s	1 s
		MySQL	3 s	19 s	9 s	< 1 s
B	10,000 x 10 kB	Derby	45 s	13 min	5 min	11 min
		H2 DB	10 s	100 s	90 s	60 s
		MySQL	15 s	135 s	70 s	10 s
C	100,000 x 1 kB	H2 DB	1 min	150 min	150 min	16 h
		MySQL	3 min	22 min	14 min	1 min

XMark Data Analysis

maximum depth = 12
average depth = 9.4
minimum depth = 5
average element count per a document = 1,455
average attribute count per a document = 333




Analyzer 200912041610
File Project View Tools Window Help
Projects
Project HTML
Project XML
Resources
Collections
Analyses
Atributes per level
Show panel
Show panels
Atributes per level
Attributes count
Level
Text content per level
Text content count
Level
Elements content per level
Element content count
Level
Empty content per level
Empty content count
Level
Mixed content per level
Mixed content count
Level
Elements per level
Elements count
Level
Bundles
Windows
Project HTML
Project XML
Plugins
Components
Registered Plugins
DTD Plugin
HTML Plugin
Sax pAgin
Universal
XQuery/XPath Plugin
Unused Plugins
Sessions
Monitors
Sessions: Inserting Documents
Chains: computing Results
Clusters: classifying Documents
Output
Glossary
System Log
User Log
Project Project Images opened
Project Project Images closed
Opening project Project XML
There is inaccessible analysis in the Project. Scheduling is disabled
Project Project XML opened
Cluster Syntax Cluster reopened
Cluster Languages Cluster reopened

Conclusion

- Original idea: to implement a tool for statistical analysis of real-world XML data
- Current status:
  - A general and extensible framework for analysis of data
    - Plugins
  - Basic plugins for XML data/schema/query analysis
- Future work:
  - Complex plugins for XML data analyses
    - Link searching
    - Sophisticated corrector of errors
    - Up-to-date XML data analysis
    - Analysis of XML operations

Further Information

- http://urtax.ms.mff.cuni.cz/anaxml/
  - Installation package + installation manual
  - Resource files + development documentation
  - User manual
  - Detailed description of plugin support and possible extensions
- analyzer.contact@gmail.com
  - Questions, comments, …

Thank you