

Differences between Versions of UML Diagrams

Dirk Ohst

Michael Welle

Udo Kelter

Presented by: Oleksii Kononenko

Feb 27, 2012

Outline

- Problem
- Differences between UML diagrams
- Presentation of differences
- Computing of differences
- Conclusion

Problem

- We want to use Software configuration management (SCM) during early phases, but:
 - Existing SCM systems work with text files; however, documents in these phases usually are diagrams
 - Cannot use usual methods for displaying differences
 - Large number of differences between two versions

Characteristics of UML Diagrams

- Semantics of the Layout
 - Relevant (sequence diagrams)
 - Irrelevant (all other)
- Structure of Diagrams
 - Node types
 - Structure of node
- Complex Nodes

Characteristics of UML Diagrams (2)

- Kinds of Attributes
 - Simple attributes
 - Multi-valued attributes
 - List-valued attributes
 - Reference to an attribute
 - Derived attributes

Classification of Differences

- Classes of differences:
 - Intra-node differences
 - Differences within a node
 - Structure differences
 - Deletion
 - Creation
 - Shifting

Intra-node Differences

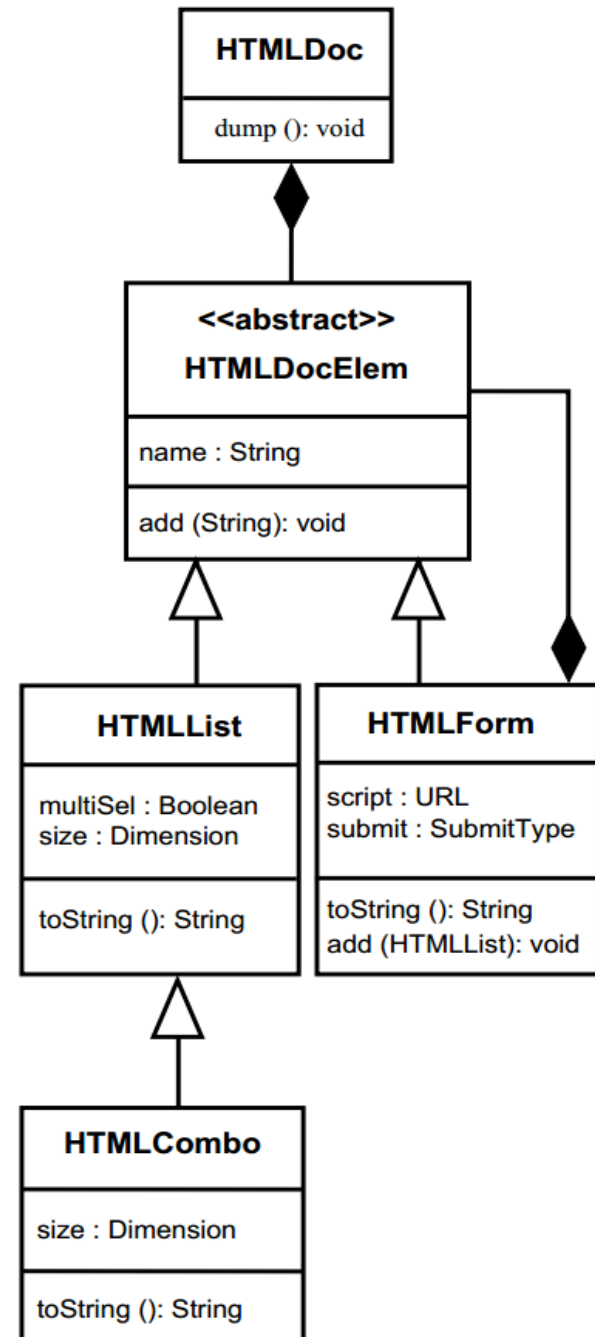
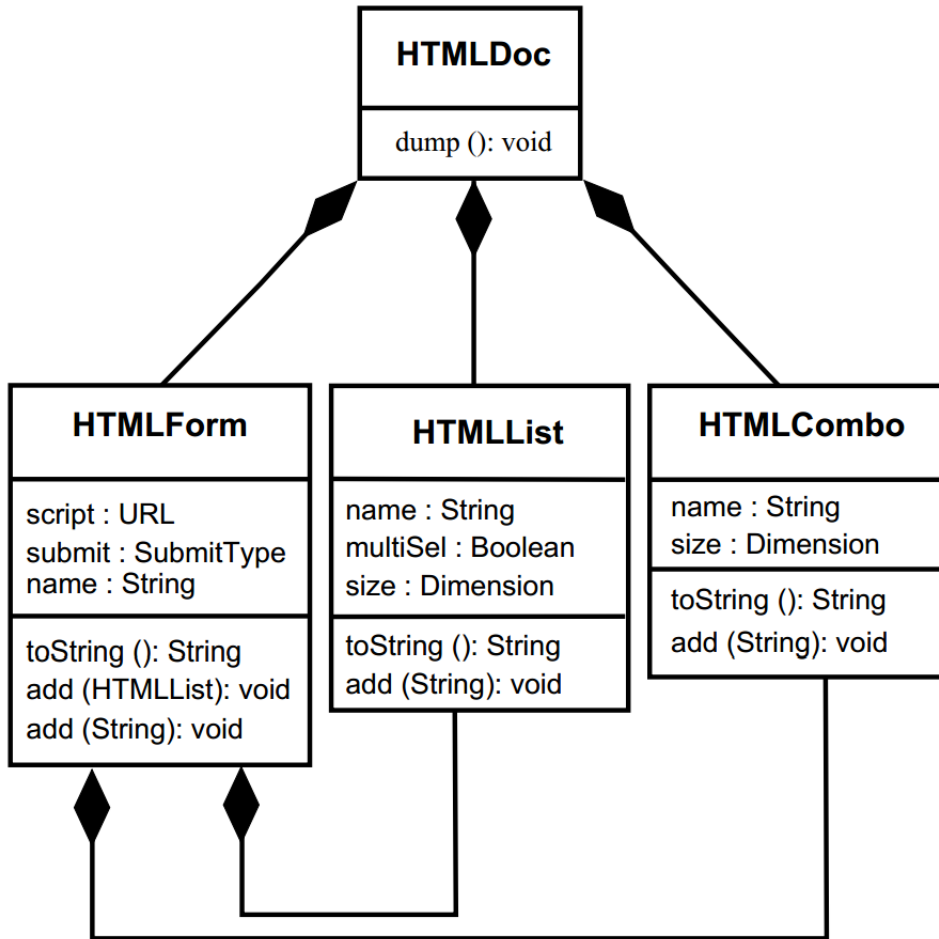
- Differences between attributes of two corresponding nodes
- Possible differences:
 - Simple attributes → Changed values
 - Multi valued attributes → Created or deleted items
 - List-valued attributes → Created or deleted items
 - Reference attributes → Modification of the reference

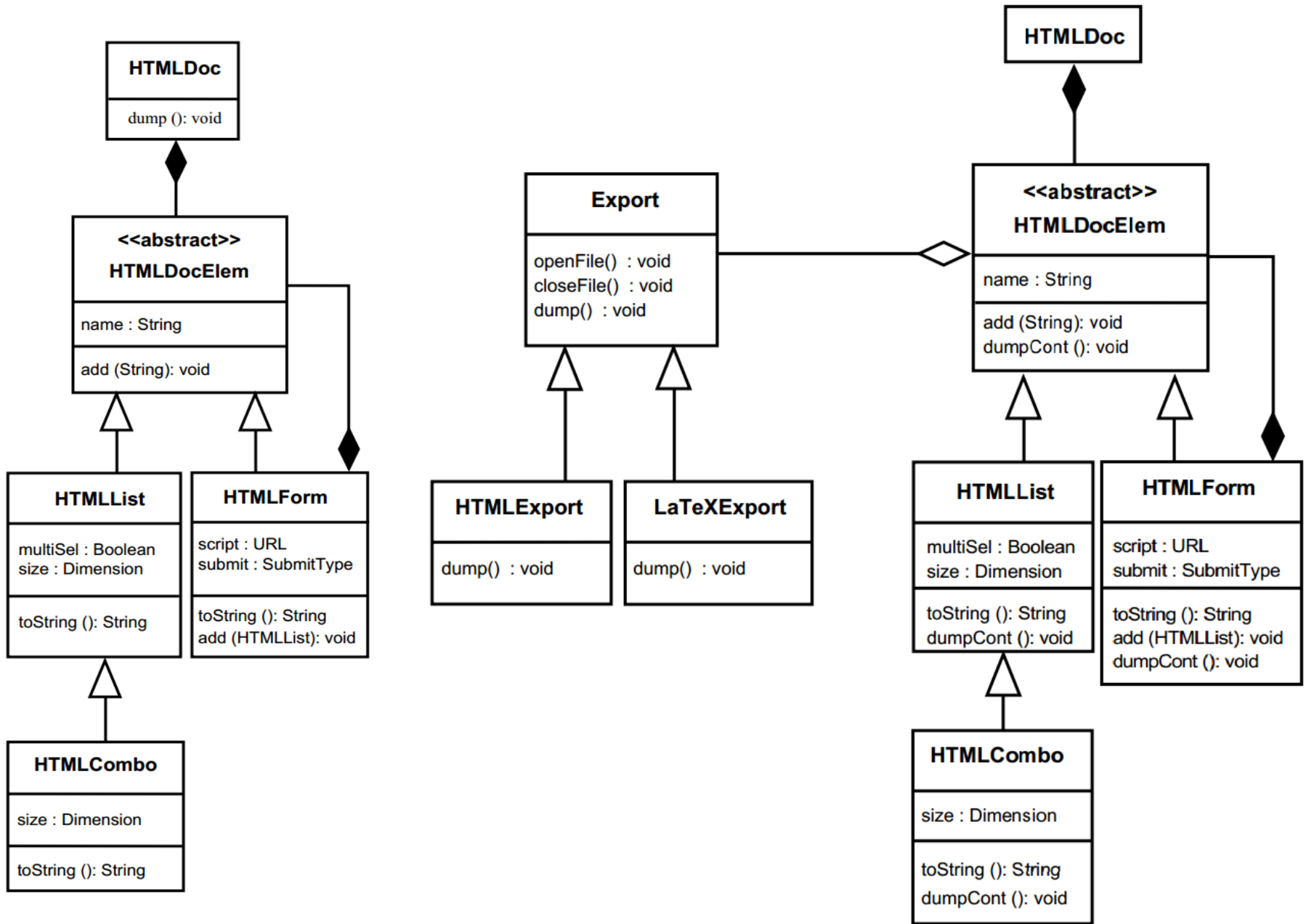
Shifting Elements within a Diagram

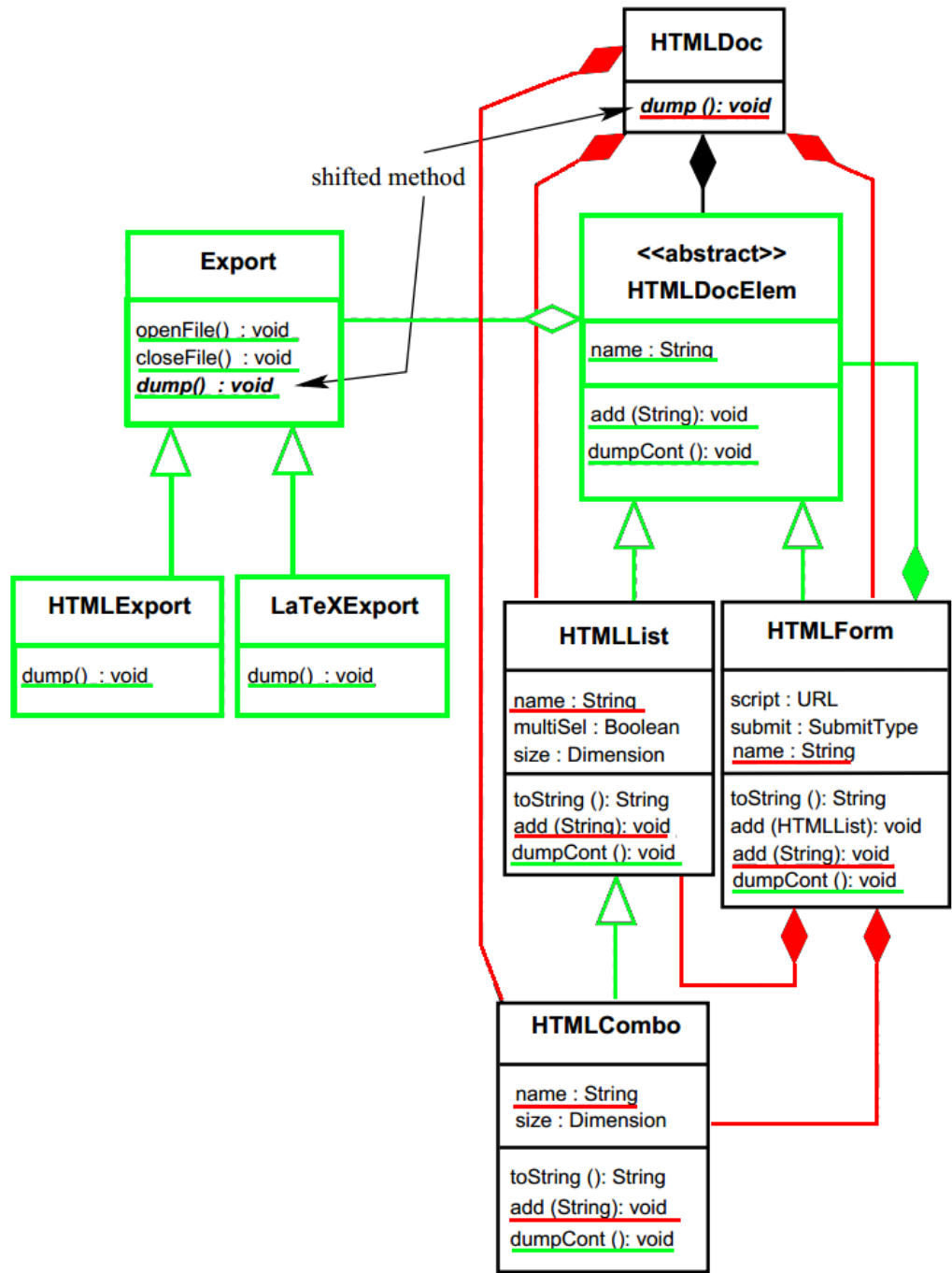
- Kinds of shifts:
 - Modifications to the layout
 - Structural shifts (e.g. shifting an edge)
 - Inter-node shifts (e.g. shifting an attribute)
 - Position shifts (e.g. reordering a list of attributes)

Presentation of differences

- “Layered” approach for data visualisation
- The unified document (UD)
 - Common parts
 - Specific parts
- 2-way and 3-way differences
 - 2-way only
- Layout of UD is similar to the layout of one of the base documents







Computing differences

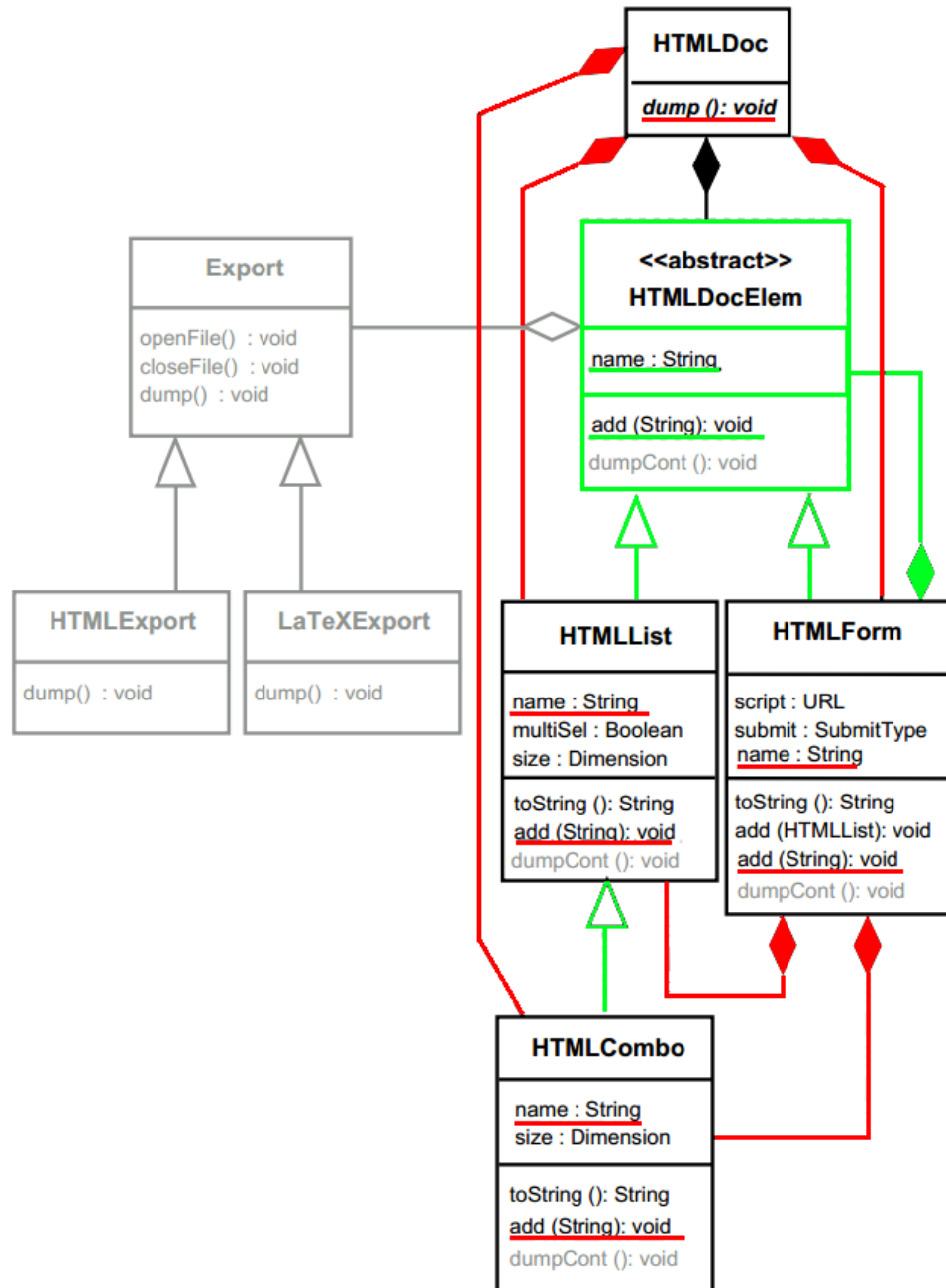
- Documents are in a fine-grained model
- All elements of a diagram are separate objects
- The prototype uses H-PCTE repository system
- Direct changing of the elements in the object graph

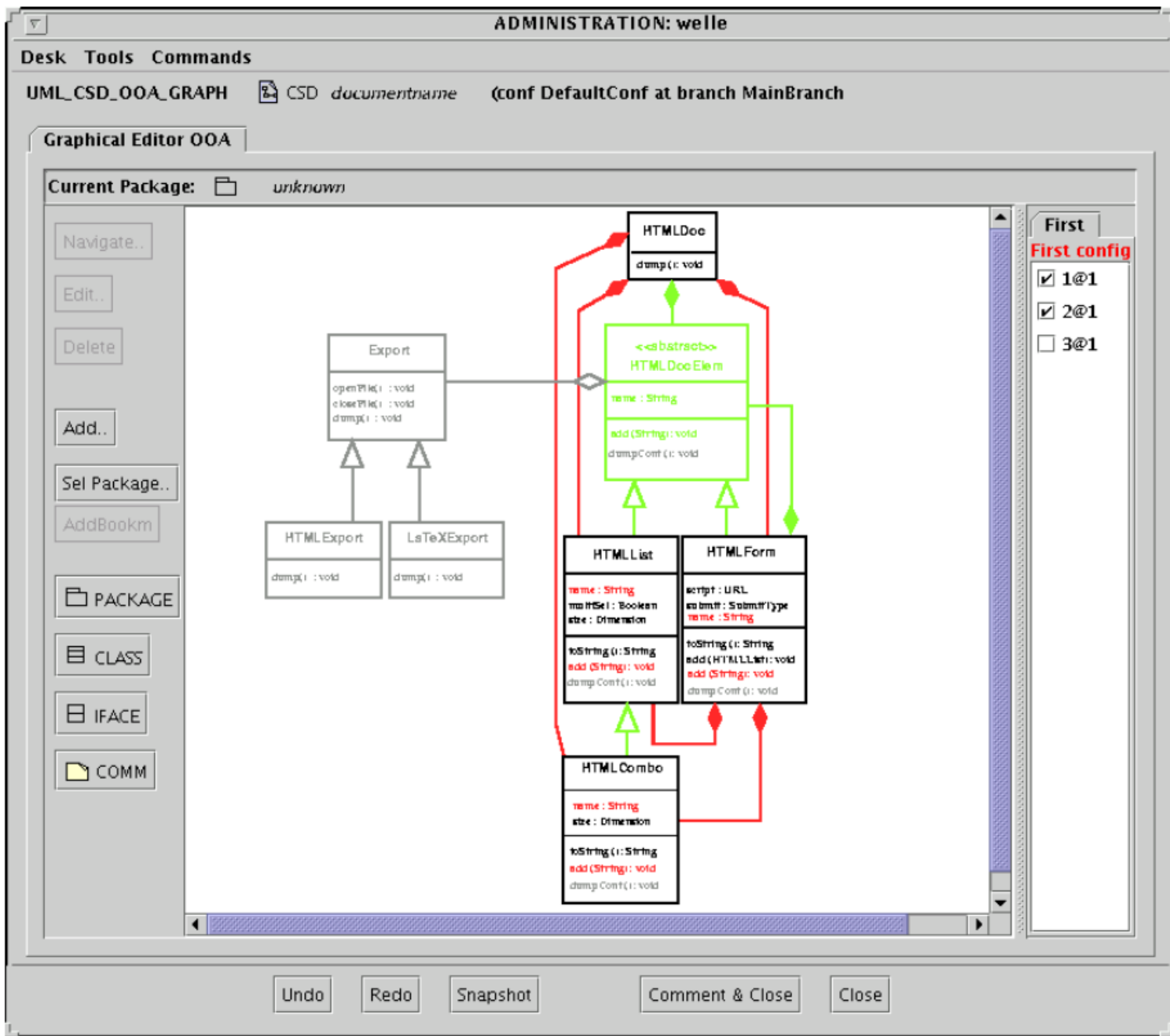
Computing differences (2)

- Assumption: the root is unchanged
 - Simultaneously goes through two spanning trees starting from the root
 - **foreach** level **in** spanningTree
 - find corresponding sub-trees
 - insert a new pair into the queue
 - compare attributes and relations of the root
 - create new object in a unified document
- end**

Number of colored differences

- Too many differences → too colorful → useless
- Color-code only a subset of all differences
- “Not-interesting” in grey
- Restrictions might be based on:
 - (1) Type of affected elements
 - (2) History of revisions
 - (3) Differences related to references
 - Combination of (1) – (3)





Conclusion

- Diagrams are represented as directed graphs
- Different kinds of differences and ways for their detection are presented
- “Layered” visualisation of graphical data
- New approach to shifts detecting

Discussion

- Even though their approach sounds quite promising, do you think it will work without features that their tools have?
- What are the limitations of the tool in terms of the number of differences on the diagram?