PyXDiff
Model Differences

How to specify and detect changes in graph-like models

Ulf Schwekendiek (ulf.schwekendiek@gmail.com)

01. May 2008

Outline

Outline

- Challenges

Outline

- Challenges
- XML-Differences Computation

Outline

- Challenges
- XML-Differences Computation
- Project Description

Outline

- Challenges
- XML-Differences Computation
- Project Description
- Implementation

Outline

- Challenges
- XML-Differences Computation
- Project Description
- Implementation
- Demo

Outline

- Challenges
- XML-Differences Computation
- Project Description
- Implementation
- Demo
- Future Work

Challenges

Challenges

Challenges

- How to detect differences in models?

Challenges

- How to detect differences in models?
  - Idea 1: Take a textual representation of the model and run *diff*

Challenges

- How to detect differences in models?
  - Idea 1: Take a textual representation of the model and run *diff*
  - Idea 2: Since models can be represented in XML use the X-Diff algorithm in order to detect changes

Challenges

- How to detect differences in models?
  - Idea 1: Take a textual representation of the model and run *diff*
  - Idea 2: Since models can be represented in XML use the X-Diff algorithm in order to detect changes
  - Idea 3: MDE approach: Everything is a model! Use known difference algorithms and represent the difference as a model.

Challenges

- How to detect differences in models?
  - Idea 1: Take a textual representation of the model and run *diff*
  - Idea 2: Since models can be represented in XML use the X-Diff algorithm in order to detect changes
  - Idea 3: MDE approach: Everything is a model! Use known difference algorithms and represent the difference as a model.

Challenges

- How to detect differences in models?
  - Idea 1: Take a textual representation of the model and run *diff*
  - Idea 2: Since models can be represented in XML use the X-Diff algorithm in order to detect changes
  - Idea 3: MDE approach: Everything is a model! Use known diff algorithms and represent the difference as a model.

XML Differences

X-Diff

GNU diff on XML files

GNU diff on XML files

book1.xml

```
1  <Books>
2    <Book>
3      <Title>Harry Potter and the Sorcerer's Stone</Title>
4      <Price>$10.00</Price>
5    </Book>
6    <Book>
7      <Title>The adventures of Tom Sawyer</Title>
8      <Price>$5.00</Price>
9    </Book>
10 </Books>
```

vs.

book2.xml

```
1  <Books>
2    <Book>
3      <Title>The adventures of Tom Sawyer</Title>
4      <Price>$29.00</Price>
5    </Book>
6    <Book>
7      <Title>Harry Potter and the Sorcerer's Stone</Title>
8      <Price>$10.00</Price>
9    </Book>
10 </Books>
```

GNU diff on XML files

book1.xml

```
1  <Books>
2    <Book>
3      <Title>Harry Potter and the Sorcerer's Stone</Title>
4      <Price>$10.00</Price>
5    </Book>
6    <Book>
7      <Title>The adventures of Tom Sawyer</Title>
8      <Price>$5.00</Price>
9    </Book>
10 </Books>
```

vs.

book2.xml

```
1  <Books>
2    <Book>
3      <Title>The adventures of Tom Sawyer</Title>
4      <Price>$29.00</Price>
5    </Book>
6    <Book>
7      <Title>Harry Potter and the Sorcerer's Stone</Title>
8      <Price>$10.00</Price>
9    </Book>
10 </Books>
```

GNU diff on XML files

book1.xml

```
1  <Books>
2    <Book>
3      <Title>Harry Potter and the Sorcerer's Stone</Title>
4      <Price>$10.00</Price>
5    </Book>
6    <Book>
7      <Title>The adventures of Tom Sawyer</Title>
8      <Price>$5.00</Price>
9    </Book>
10 </Books>
```

vs.

book2.xml

```
1  <Books>
2    <Book>
3      <Title>The adventures of Tom Sawyer</Title>
4      <Price>$29.00</Price>
5    </Book>
6    <Book>
7      <Title>Harry Potter and the Sorcerer's Stone</Title>
8      <Price>$10.00</Price>
9    </Book>
10 </Books>
```

GNU diff on XML files

book1.xml

```
1  <Books>
2    <Book>
3      <Title>Harry Potter and the Sorcerer's Stone</Title>
4      <Price>$10.00</Price>
5    </Book>
6    <Book>
7      <Title>The adventures of Tom Sawyer</Title>
8      <Price>$5.00</Price>
9    </Book>
10 </Books>
```

vs.

book2.xml

```
1  <Books>
2    <Book>
3      <Title>The adventures of Tom Sawyer</Title>
4      <Price>$29.00</Price>
5    </Book>
6    <Book>
7      <Title>Harry Potter and the Sorcerer's Stone</Title>
8      <Price>$10.00</Price>
9    </Book>
10 </Books>
```

GNU diff on XML files

book1.xml

```
1  <Books>
2    <Book>
3      <Title>Harry Potter and the Sorcerer's Stone</Title>
4      <Price>$10.00</Price>
5    </Book>
6    <Book>
7      <Title>The adventures of Tom Sawyer</Title>
8      <Price>$5.00</Price>
9    </Book>
10 </Books>
```

vs.

book2.xml

```
1  <Books>
2    <Book>
3      <Title>The adventures of Tom Sawyer</Title>
4      <Price>$29.00</Price>
5    </Book>
6    <Book>
7      <Title>Harry Potter and the Sorcerer's Stone</Title>
8      <Price>$10.00</Price>
9    </Book>
10 </Books>
```

book1-book2.diff

```
1  3,4c3,4
2  <     <Title>Harry Potter and the Sorcerer's Stone</Title>
3  <   <Price>$10.00</Price>
4  ---
5  >     <Title>The adventures of Tom Sawyer</Title>
6  >   <Price>$29.00</Price>
7  7,8c7,8
8  <     <Title>The adventures of Tom Sawyer</Title>
9  <   <Price>$5.00</Price>
10 ---
11 >     <Title>Harry Potter and the Sorcerer's Stone</Title>
12 >   <Price>$10.00</Price>
13
```

GNU Diff upgrade script

GNU diff on XML files

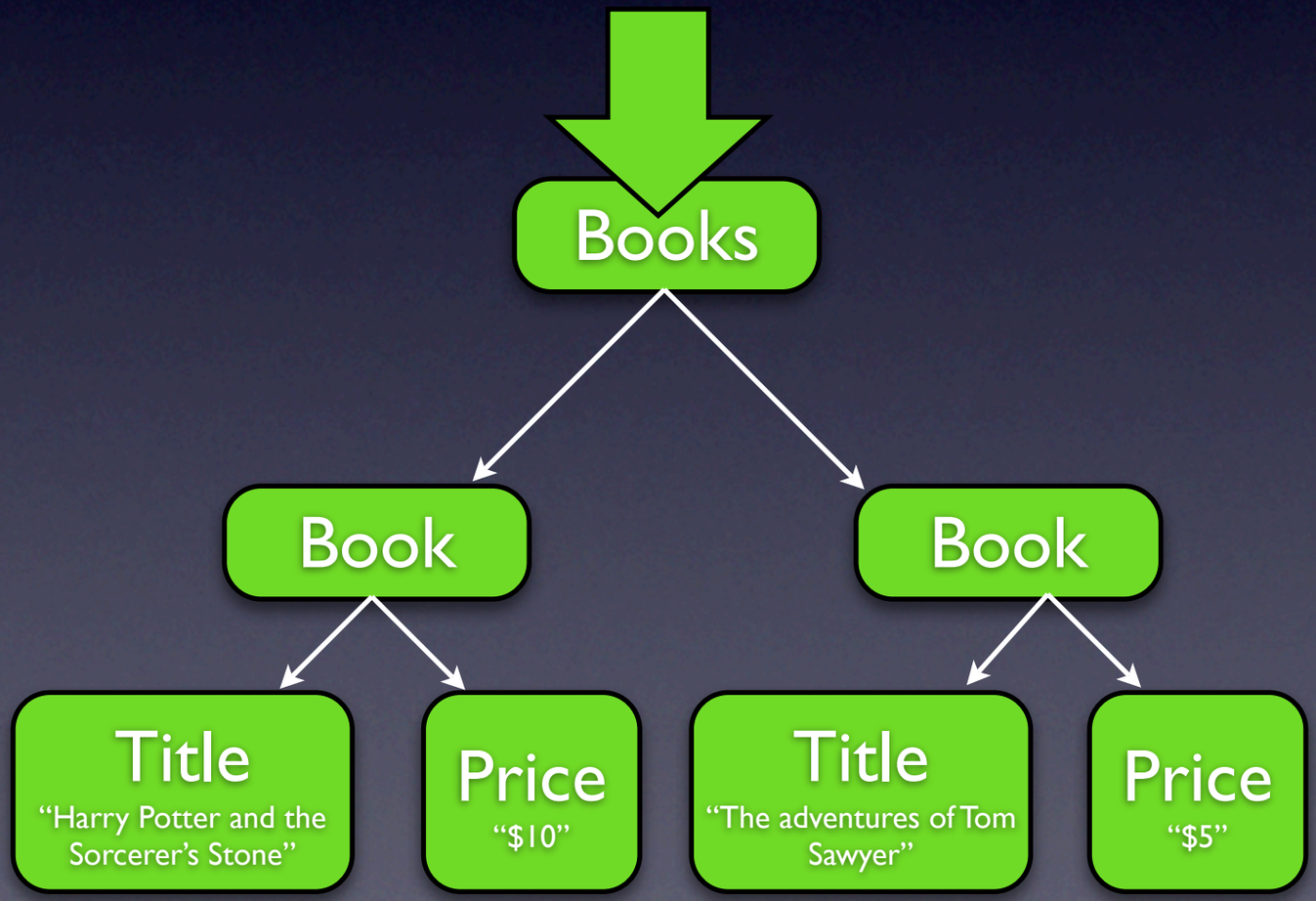
book1.xml

```
1  <Books>
2    <Book>
3      <Title>Harry Potter and the Sorcerer's Stone</Title>
4      <Price>$10.00</Price>
5    </Book>
6    <Book>
7      <Title>The adventures of Tom Sawyer</Title>
8      <Price>$5.00</Price>
9    </Book>
10 </Books>
```

vs.

book2.xml

```
1  <Books>
2    <Book>
3      <Title>The adventures of Tom Sawyer</Title>
4      <Price>$29.00</Price>
5    </Book>
6    <Book>
7      <Title>Harry Potter and the Sorcerer's Stone</Title>
8      <Price>$10.00</Price>
9    </Book>
10 </Books>
```



GNU diff on XML files

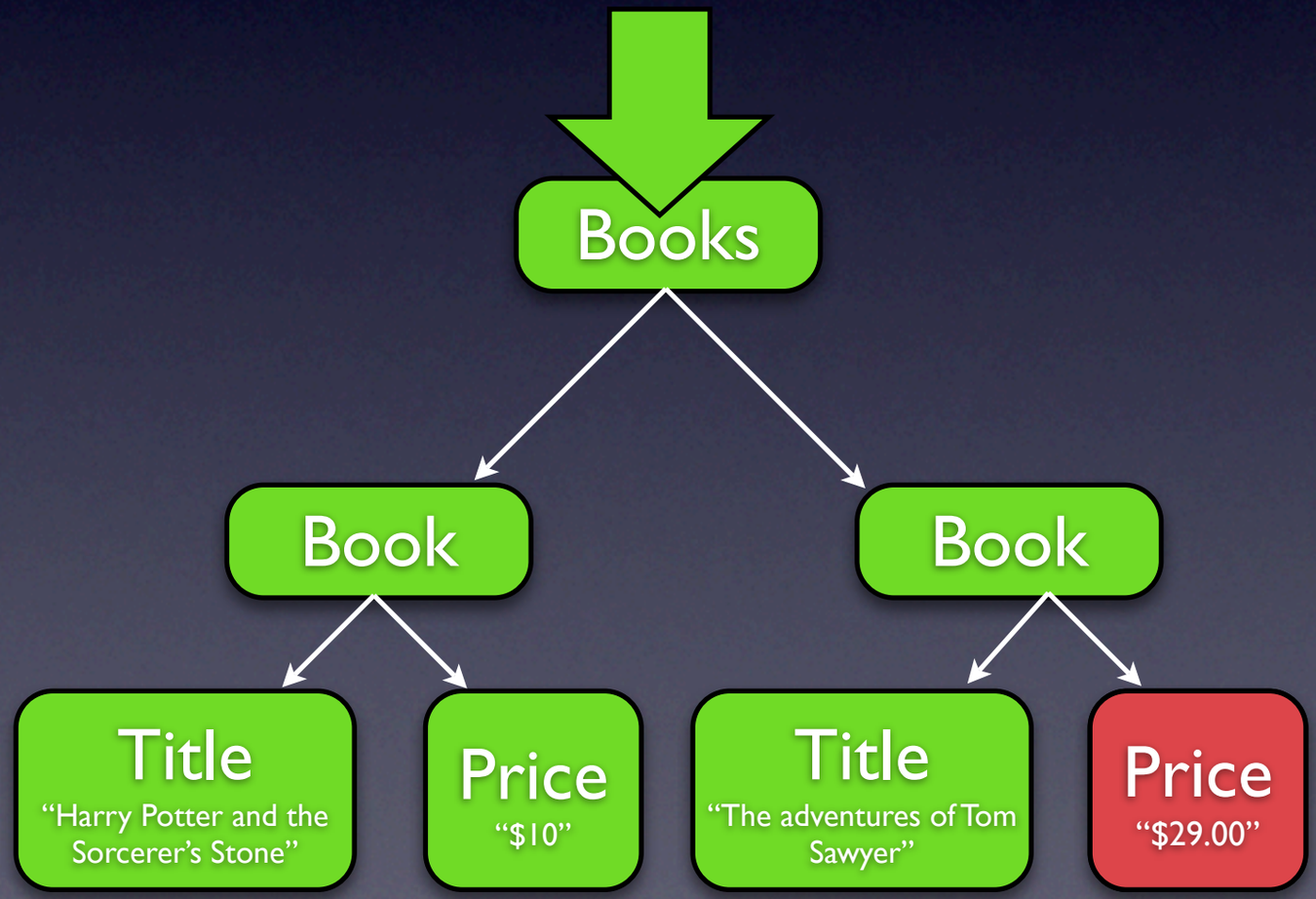
book1.xml

```
1  <Books>
2    <Book>
3      <Title>Harry Potter and the Sorcerer's Stone</Title>
4      <Price>$10.00</Price>
5    </Book>
6    <Book>
7      <Title>The adventures of Tom Sawyer</Title>
8      <Price>$5.00</Price>
9    </Book>
10 </Books>
```

vs.

book2.xml

```
1  <Books>
2    <Book>
3      <Title>The adventures of Tom Sawyer</Title>
4      <Price>$29.00</Price>
5    </Book>
6    <Book>
7      <Title>Harry Potter and the Sorcerer's Stone</Title>
8      <Price>$10.00</Price>
9    </Book>
10 </Books>
```



X-Diff

Y. Wang, D.J. DeWitt and J. Cai

"An Effective Change Detection Algorithm for XML Documents"

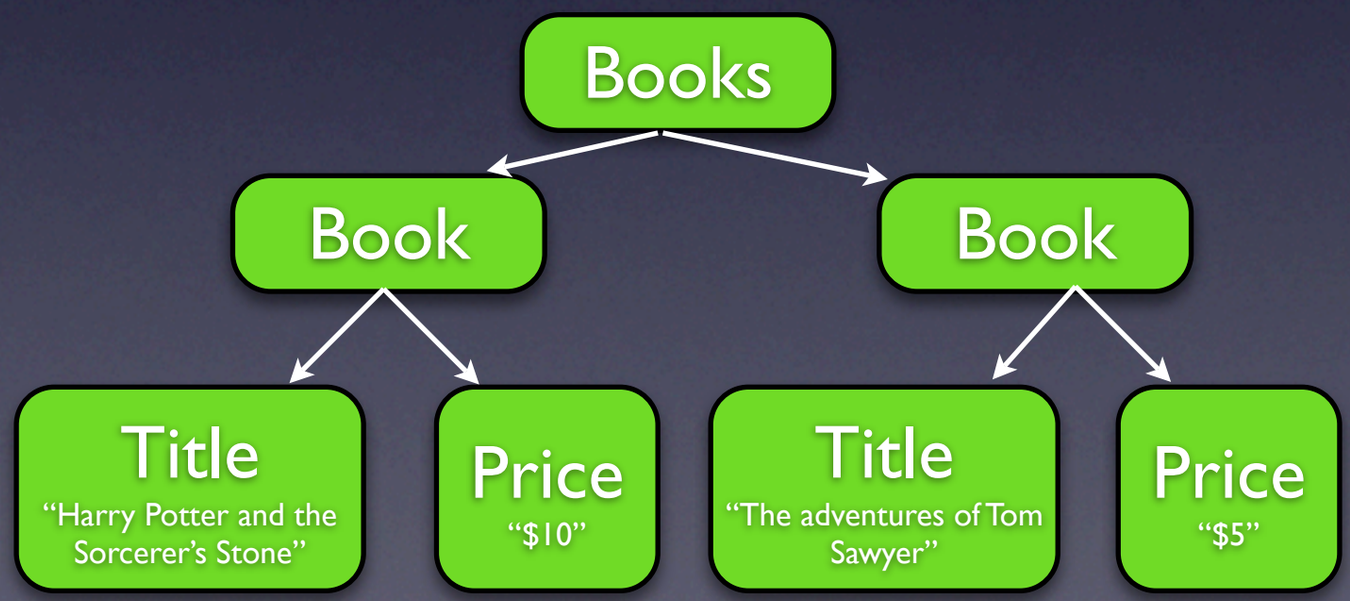
University of Wisconsin (2003)

- XML structure is hierarchical and can be represented as a tree
- Unordered trees
  - only ancestor relationship is important
  - optimal change detection of unordered trees is NP complete[1]
- Algorithm generates near optimal edit scripts
- Two trees are *isomorphic* if they are identical except for the ordering of siblings. X-Diff considers two trees as equivalent if they are *isomorphic*.

[1] K. Zhang, R. Statman, D. Shasha, "On the Editing Distance betweem Unordered Labeled Trees", *Information Processing Letters*, 42: 133-139, 1992

Tree Definition

- A tree has three kinds of nodes:
  - **Element** nodes - non-leaf nodes with one label, *name*
  - **Text** nodes - leaf nodes with one label, *value*
  - **Attribute** nodes - leaf nodes with two labels, *name* and *value*



Algorithm

Algorithm

- In order to compare edit scripts a cost model is introduced, which also affect the algorithm design.

Algorithm

- In order to compare edit scripts a cost model is introduced, which also affect the algorithm design.
  - X-Diff uses a simple cost model

Algorithm

- In order to compare edit scripts a cost model is introduced, which also affect the algorithm design.
  - X-Diff uses a simple cost model
- Unnecessary to compare each source node with any other target node

Algorithm

- In order to compare edit scripts a cost model is introduced, which also affect the algorithm design.
  - X-Diff uses a simple cost model
- Unnecessary to compare each source node with any other target node
  - only nodes of the same type are matched

Algorithm

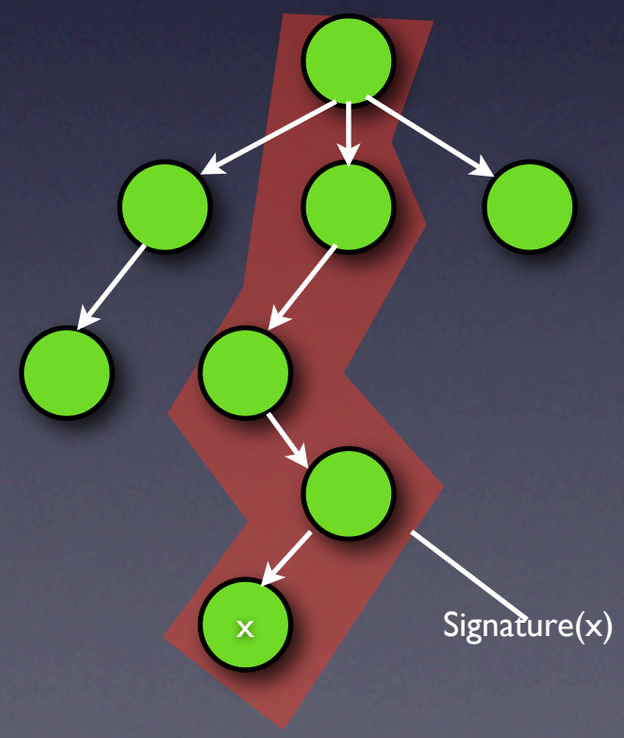
- In order to compare edit scripts a cost model is introduced, which also affect the algorithm design.
  - X-Diff uses a simple cost model
- Unnecessary to compare each source node with any other target node
  - only nodes of the same type are matched
  - text nodes should not be matched with element nodes or attribute nodes

Algorithm

- In order to compare edit scripts a cost model is introduced, which also affect the algorithm design.
  - X-Diff uses a simple cost model
- Unnecessary to compare each source node with any other target node
  - only nodes of the same type are matched
  - text nodes should not be matched with element nodes or attribute nodes
  - Signature is introduced

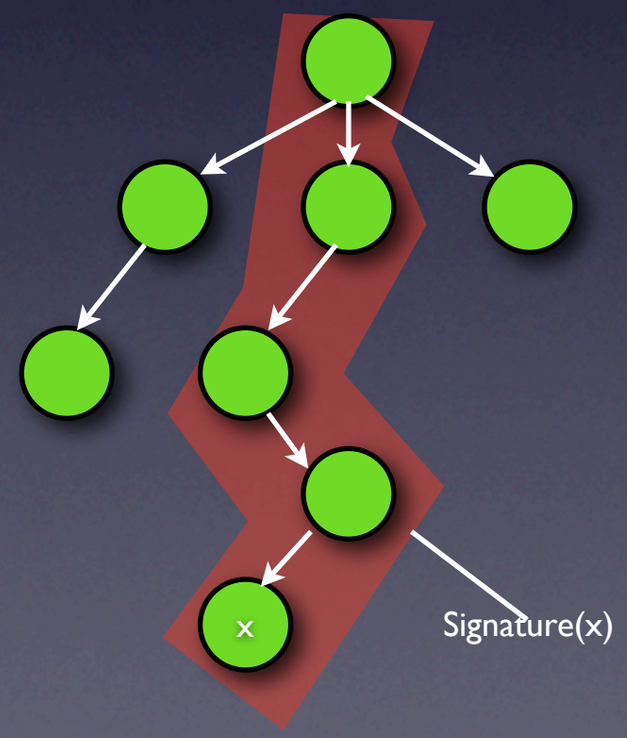
Algorithm

- In order to compare edit scripts a cost model is introduced, which also affect the algorithm design.
  - X-Diff uses a simple cost model
- Unnecessary to compare each source node with any other target node
  - only nodes of the same type are matched
  - text nodes should not be matched with element nodes or attribute nodes
  - Signature is introduced



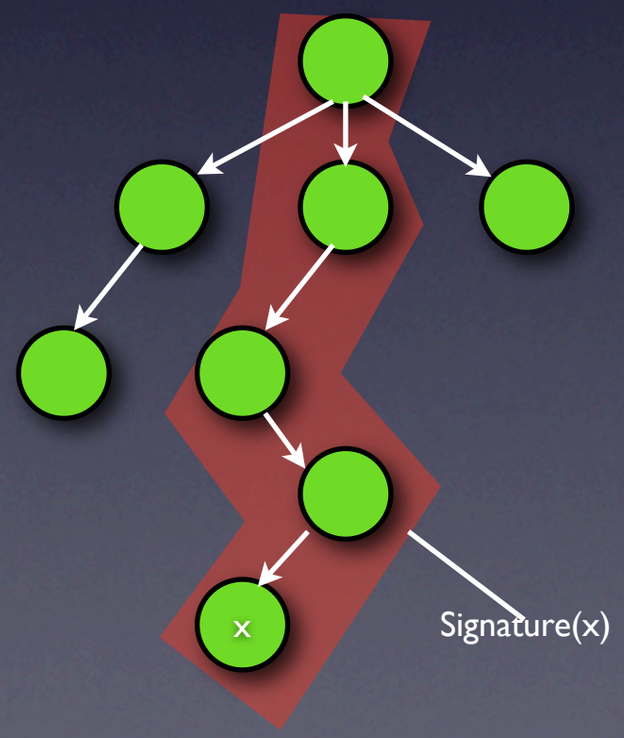
Algorithm

- In order to compare edit scripts a cost model is introduced, which also affect the algorithm design.
  - X-Diff uses a simple cost model
- Unnecessary to compare each source node with any other target node
  - only nodes of the same type are matched
  - text nodes should not be matched with element nodes or attribute nodes
  - Signature is introduced
- Based on minimum-Cost Matching an upgrade script is generated



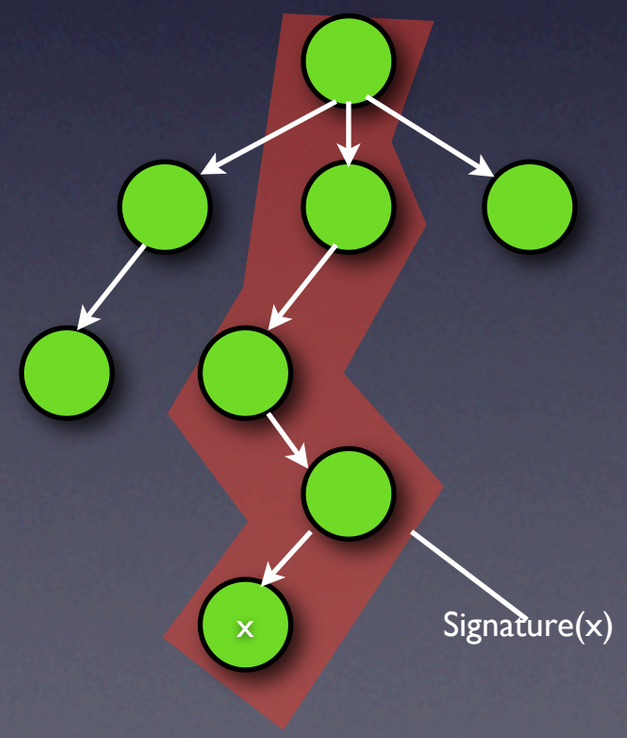
Algorithm

- In order to compare edit scripts a cost model is introduced, which also affect the algorithm design.
  - X-Diff uses a simple cost model
- Unnecessary to compare each source node with any other target node
  - only nodes of the same type are matched
  - text nodes should not be matched with element nodes or attribute nodes
  - Signature is introduced
- Based on minimum-Cost Matching an upgrade script is generated
- Algorithm only computes editing distances between nodes that have the same signature



Algorithm

- In order to compare edit scripts a cost model is introduced, which also affect the algorithm design.
  - X-Diff uses a simple cost model
- Unnecessary to compare each source node with any other target node
  - only nodes of the same type are matched
  - text nodes should not be matched with element nodes or attribute nodes
  - Signature is introduced
- Based on minimum-Cost Matching an upgrade script is generated
- Algorithm only computes editing distances between nodes that have the same signature
  - Important to archive polynomial time complexity



Project Implementation

Implementing X-Diff in Python

Implementation

Implementation

- Since ATOM^3 is implemented in Python it would be useful to extend it at some point with model difference computation and representation

Implementation

- Since ATOM^3 is implemented in Python it would be useful to extend it at some point with model difference computation and representation
- In my project I implemented the base, the XML difference computation algorithm

Implementation

- Since ATOM^3 is implemented in Python it would be useful to extend it at some point with model difference computation and representation
- In my project I implemented the base, the XML difference computation algorithm
- Algorithm is released under GPL3 and will be in future maintained and extended as a private project

Implementation - Input

XDiff.py

```
274 class XDiff(object):
275     """
276     This class is a wrapper for the XDIFF methods.
277     It takes two xml files and creates the editscript in I{self.editscript}.
278     """
279
280     def __init__(self, xmlfile1, xmlfile2):
281         # create Reader object
282         reader = Sax2.Reader()
283         doc1 = reader.fromStream(xmlfile1)
284         doc2 = reader.fromStream(xmlfile2)
285
286         root1 = doc1.documentElement
287         root2 = doc2.documentElement
288
289         matching = findMatching(root1, root2)
290         #: The computed editscript
291         self.editscript = editScript(root1, root2, matching, set([]))
```

Line: 274 Column: 1 Python Tab Size: 4 XDiff(object)

- The input for the X-Diff algorithm are two Strings linking to XML files
- After that two DOM-trees are parsed using the libraries:
  `xml.dom.ext.reader`
  `xml.dom`

Implementation - Help classes

Implementation - Help classes

- Help class:

Implementation - Help classes

- Help class:
  - **Action:** defines a change-action. An editscript is a python set (`set([])`) that contains action objects.

Implementation - Help classes

- Help class:
  - **Action:** defines a change-action. An editscript is a python set (`set([])`) that contains action objects.
    - contains a source DOM node

Implementation - Help classes

- Help class:
  - **Action:** defines a change-action. An editscript is a python set (`set([])`) that contains action objects.
    - contains a source DOM node
    - can contains a target DOM node

Implementation - Help classes

- Help class:
  - **Action:** defines a change-action. An editscript is a python set (`set([])`) that contains action objects.
    - contains a source DOM node
    - can contains a target DOM node
    - contains a action type: *ADD*, *REMOVE* or *UPGRADE*

Implementation - Methods

Implementation - Methods

- The PyXDiff Package contains the following methods:

Implementation - Methods

- The PyXDiff Package contains the following methods:
  - collectLeafNodes(root)

Implementation - Methods

- The PyXDiff Package contains the following methods:
  - collectLeafNodes(root)
  - collectNodes(root, list)

Implementation - Methods

- The PyXDiff Package contains the following methods:
  - collectLeafNodes(root)
  - collectNodes(root, list)
  - removeNodes(nodeList)

Implementation - Methods

- The PyXDiff Package contains the following methods:
  - collectLeafNodes(root)
  - collectNodes(root, list)
  - removeNodes(nodeList)
  - signature(node)

Implementation - Methods

- The PyXDiff Package contains the following methods:
  - collectLeafNodes(root)
  - collectNodes(root, list)
  - removeNodes(nodeList)
  - signature(node)
  - dist(x, y)

Implementation - Methods

- The PyXDiff Package contains the following methods:
  - collectLeafNodes(root)
  - collectNodes(root, list)
  - removeNodes(nodeList)
  - signature(node)
  - dist(x, y)
  - isRoot(x)

Implementation - Main Methods

Implementation - Main Methods

- The two main methods are:

Implementation - Main Methods

- The two main methods are:
  - **findMatching(t1, t2)**
    This method returns from two tree nodes, that are treated as root nodes a minimum cost matching set.

Implementation - Main Methods

- The two main methods are:
  - **findMatching(t1, t2)**
    This method returns from two tree nodes, that are treated as root nodes a minimum cost matching set.
  - **editScript(t1, t2, M_min, script)**
    This method creates an edit script. It takes two root nodes from two DOM trees, the previous computed minimum cost matching set and returns a valid edit script consisting of Action objects.

Demo

Future Work

Future Work

Future Work

- To speedup the algorithm the paper proposes the use of XHash, which calculates a hash value for a node in an isomorphic tree. I was not able to find such algorithm yet, so this needs to be also integrated in PyXDiff.

Future Work

- To speedup the algorithm the paper proposes the use of XHash, which calculates a hash value for a node in an isomorphic tree. I was not able to find such algorithm yet, so this needs to be also integrated in PyXDiff.
- Further work on Model Differences

Future Work

- To speedup the algorithm the paper proposes the use of XHash, which calculates a hash value for a node in an isomorphic tree. I was not able to find such algorithm yet, so this needs to be also integrated in PyXDiff.
- Further work on Model Differences
  - Do not compute difference on models itself, rather map the model to its semantic domain and try to differentiate there

Future Work

- To speedup the algorithm the paper proposes the use of XHash, which calculates a hash value for a node in an isomorphic tree. I was not able to find such algorithm yet, so this needs to be also integrated in PyXDiff.
- Further work on Model Differences
  - Do not compute difference on models itself, rather map the model to its semantic domain and try to differentiate there
  - Note: It has to kept track of backlinks.

Questions?