

Scripting Documents with XQuery – Virtual Documents in TNTBase

Vyacheslav Zholudev, Michael Kohlhase

Jacobs University Bremen

KWARC – Knowledge Adaptation and Reasoning for Content

DFKI, Bremen

August 6, 2010

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

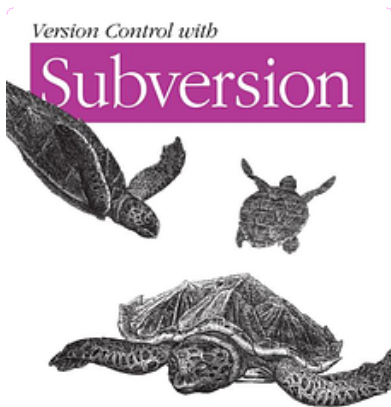
Future work

Summary

- Why **Versioning** ?
- Why **XML**?

- SVN Book sources:

<http://svnbook.googlecode.com/svn/>



Our Goal: Support your Workflows

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

Future work

Summary

- Provide a **basis** for XML-based (e.g. DocBook-based) management systems
- XML-fragment access techniques
- Make possible **workflows** and globally distributed project teams as we know them from Open Source community

What is TNTBase?

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

Future work

Summary

- Client-server architecture
- TNTBase Core = Subversion + Berkeley DB XML
= **xSVN**
- Web application based on xSVN

What do we inherit from SVN?

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

Future work

Summary

- Communication
 - Any SVN client (Command-line, SVNKit Java Library, Tortoise SVN)
- Sophisticated versioning
 - History support
 - Collaborative editing
 - Branching, tagging
 - Storing non-XML files

Facilitates migrating from SVN-based systems/applications

What do we get from DB XML?

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

Future work

Summary

- Communication
 - Via RESTful interface build on top of DB XML storage
- Sophisticated XML database
 - XQuery/XQuery Update
 - Indexing
 - XQuery external functions
 - ACID Transactions
 - Data integrity

Perfect basis for managing huge collections of XML data

What is on top of xSVN?

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications
VDoc FS Entity
Querying VDocs
VDoc Editing
Conclusion

Demo

Use Cases

Future work

Summary

- Communication: *RESTful* interface
- TNTBase XQuery functions
 - addressing path-based collections with *revision axis*
 - repository information
 - ... more
- *XML-diff*
- Your own validation and presentational workflows managed by SVN properties
- Plugin architecture for validation, presentation, interface extraction and XML-diff
- Number of built-in plugins, e.g.
 - Extracting RDF from XML files
 - Content MathML → Presentation MathML
 - Number of XML differs
- and ...

Virtual Documents

- No in-document macros in XML
- No analogues of views in XML databases
- “Views” should be first-class citizens in XML databases

- Virtual Documents (**VDocs**) are first class citizens in TNTBase
- “XML-database” views that are:
 - Queryable
 - Parametrizable
 - Editable
 - Materializable
 - Comprise our repository’s document format \Rightarrow can reuse our workflows

OR

- Like JSP is for HTML/Java, VDocs are for XML/XQuery

- Have a collection of mathematical documents in **OMDoc**
- Want to get a “view” (VDoc) on all exercises together with authors → two simple XQueries
- VDoc should comprise the same XML format ⇒ Need XML wrappers
- Add new documents → VDoc is automatically updated

The result should be like ...

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

Future work

Summary

```

<omdoc xmlns:dc="http://purl.org/dc/elements/1.1/">
  < dc:title >Exercises for Computer Science lectures</dc:title>
  <dc:creator>Vyacheslav Zholudev</dc:creator>
  <section>
    < dc:title >Acknowledgements</dc:title>
    <omtext>
      The following individuals have contributed material to this document:
      <author>Vyacheslav Zholudev</author>
      <author>Michael Kohlhase</author>
      ...
    </omtext>
  </section>
  <section>
    < dc:title >Exercises</ dc:title >
    <exercise title="Graph properties">...</exercise>
    ...
  </section>
</omdoc>

```

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

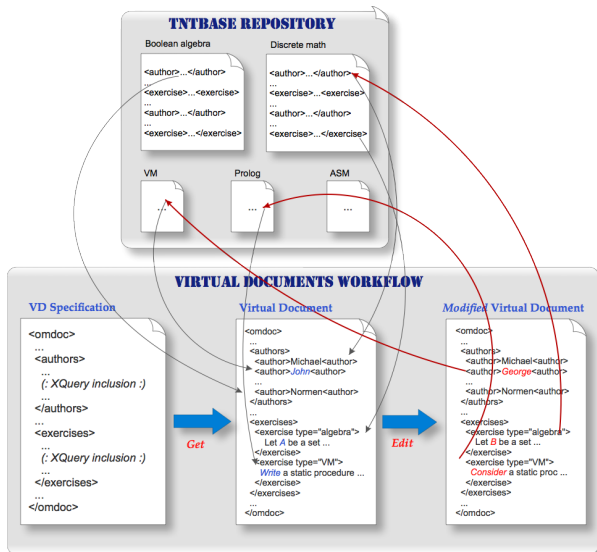
VDoc Specifications
VDoc FS Entity
Querying VDocs
VDoc Editing
Conclusion

Demo

Use Cases

Future work

Summary



```
<tnt:virtualdocument xmlns:tnt="http://tntbase.mathweb.org/ns">
  <tnt:skeleton xml:id="exercises">
    <omdoc xmlns:dc="http://purl.org/dc/elements/1.1/">
      <dc:title>Exercises for Computer Science lectures</dc:title>
      <dc:creator>Vyacheslav Zholudev</dc:creator>
      ... <!-- XQueries go in between -->
    </omdoc>
  </tnt:skeleton>

  <tnt:query name="exercises.xq"> for $t in $topics return
    tnt:collection (concat('/exercises/', $t, '/*.omdoc'))//
      exercise[position () le $max]
  </tnt:query>

  <tnt:params>
    <tnt:param name="max">
      <tnt:value>10</tnt:value>
    </tnt:param>
    ...
  </tnt:params>
</tnt:virtualdocument>
```

- Skeleton
- XQueries (*optional*)
- Parameters (*optional*)

There is a schema for VDoc specifications

```
https://svn.tntbase.org/repos/tntbase/trunk/  
DbXmlAccessLib/resources/tnt-vd-spec.rnc
```

```

<tnt:skeleton xml:id="exercises">
  <omdoc xmlns:dc="http://purl.org/dc/elements/1.1/">
    < dc:title >Exercises for Computer Science lectures</dc:title>
    <dc:creator>Vyacheslav Zholudev</dc:creator>
    <section>
      < dc:title >Acknowledgements</dc:title>
      <omtext>
        <tnt:xqinclude query="tnt:collection('/exercises/* .omdoc')//dc:creator/*">
          < tnt:return >
            <author><tnt:result/><author>
          </ tnt:return >
        </ tnt:xqinclude >
      </omtext>
    </section>
    ...
  </omdoc>
</tnt:skeleton>

```

```

<tnt:skeleton xml:id="exercises">
  <omdoc xmlns:dc="http://purl.org/dc/elements/1.1/">
    < dc:title >Exercises for Computer Science lectures</dc:title>
    <dc:creator>Vyacheslav Zholudev</dc:creator>
    ...
    <section>
      < dc:title >Exercises</ dc:title >
      <tnt:xqinclude>
        <tnt:query name="exercises.xq"/>
        <tnt:return><tnt:result/></ tnt:return >
      </tnt:xqinclude>
    </section>
  </omdoc>
</tnt:skeleton>

```

```
<tnt:query name="exercises.xq">
  for $t in $topics return
    tnt:collection (concat('/exercises/', $t, '/*.omdoc'))//
      exercise[position() le $max]
</tnt:query>
```

Queries can be defined as

- an attribute in a skeleton
- a child element of *tnt:xqinclude*
- a reference to a *query* element
- a reference to a repository file

```
tnt:collection(concat('/exercises/', $t,  
'/*.omdoc'))//exercise[position() le $max]
```

```
<tnt:params>  
  <tnt:param name="max">  
    <tnt:value>10</tnt:value>  
  </tnt:param>  
  <tnt:param name="topics">  
    <tnt:value>search</tnt:value>  
    <tnt:value>graphs</tnt:value>  
  </tnt:param>  
</tnt:params>
```

Other Features of VDoc Specs

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

Future work

Summary

- VDoc Specs can inherit skeletons from other VDoc Specs
- XQueries can reside in a separate repository file
- You can override XQueries and parameters
- VDoc Specs can be “abstract”

VDoc File System entity

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

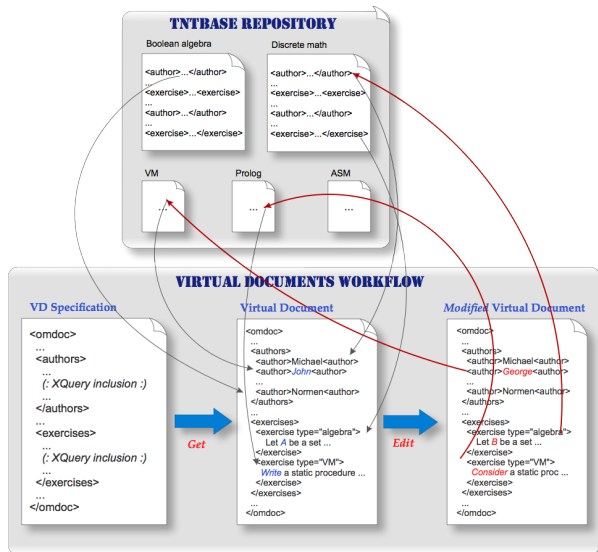
Conclusion

Demo

Use Cases

Future work

Summary



VDoc File System entity and Materialization

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

Future work

Summary

Q: How do I create a VDoc?

A: Associate a file system entity with a specification

Q: Can I influence VDoc Spec parameters?

A: Yes!

Q: Are VDocs versioned?

A: No, but VDoc Specs are!

Q: How do I get the content?

A: Use XQuery or RESTful interface

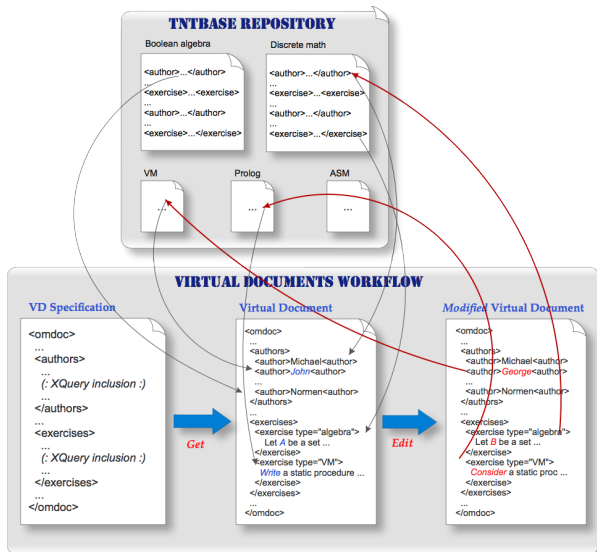
```
tnt:vdoc('/path/to/exercises.vdoc')
```

Q: What to do if I want to version the content?

A: Materialize them!

It's easy:

- `tnt:vdoc($path as xs:string) as document-node()`
- Can use VDocs in XQueries of other VDocs



- VDocs can be edited and “committed” back
- Changes are propagated back to the source documents
- History is preserved
- Abstraction from a notion of a file
- “Committing” via RESTful interface

Prerequisites for Editing

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

Future work

Summary

- Only XQuery result nodes can be edited
- Only XML *elements* that came from *physical files* are editable
- The XML element can be present only once in a VDoc
- Only HEAD revision results are editable

Why not a single XQuery?

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

Future work

Summary

- Think of JSP
- Modularization
- Distribution of responsibilities
- Inheriting of VDoc Specs
- Overriding XQueries/parameters
- One VDoc Spec for multiple VDocs that differ in parameters

There is nothing as *practical*
as a good *demo*

Real-world Use Cases

- Corpus of more than 1000 exercises for students in the OMDoc format
- Automated exam generation by means of VDocs
- Different parameters for different type of exams
- Editing facilities allow to easily fix mistakes in underlying documents

Write a VDoc Spec once and reuse it over the years!

Multiple Versions of Documents

Scripting
Documents
with XQuery

Vyacheslav
Zholudev,
Michael
Kohlhase

TNTBase
Recap

Virtual
Documents

VDoc Specifications

VDoc FS Entity

Querying VDocs

VDoc Editing

Conclusion

Demo

Use Cases

Future work

Summary

Example:

- XML 1.0/1.1
- XQuery 1.0/1.1
- MathML1.0/1.0.1/2.0/2.0(2e)/3.0
- ...

Experimenting with XML specifications

- in XMLSpec format
- What formal parts have been changed in XML 1.1?
- Provide a *Diff* VDoc that summarizes formal changes
- Fixing bugs without navigating to the source documents

Get the up-to-date content via VDoc “file” in a repository

Preview refactored ontologies before saving them!

- renaming entities
- factoring out or merging modules
- rewriting axioms
- lowering expressivity
- stripping axiom annotations

Single VDoc Spec is applicable for different ontologies for the same type of a refactoring!

Demo video:

<http://tntbase.org/wiki/screencasts>

- Editing of a VDoc presentation (like XHTML)
- More checks for validity of a VDoc Spec
- Make XQuery errors more informative
- Web interface for managing VDocs

- ① Analogue of JSP/PHP or $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ in the **XML world**
- ② **Enabling** technology
- ③ VDocs **are not** tied to TNTBase
- ④ Write **once** - use for **different** needs
- ⑤ **Many applications** like generation of *news* or *guided tours*

Questions???



Project page: <http://tntbase.org>

E-mail: v.zholudev@jacobs-university.de