

The MMiSS Repository

Christoph Lüth

April 2004



Introducing: the Repository

- What is a **repository**?
 - central database of **all documents** and their **history**

Introducing: the Repository

- What is a **repository**?
 - central database of **all documents** and their **history**
- What does it **do**?
 - **Version control**
 - **Configuration management**
 - **Change management**

Introducing: the Repository

- What is a **repository**?
 - central database of **all documents** and their **history**
- What does it **do**?
 - **Version control**
 - **Configuration management**
 - **Change management**
- **Well-known** software engineering concept.
 - CVS, Subversion, arch, ClearCase, SourceSafe, . . .
- **New**: application to **structured documents**.

Introducing: the Repository

- What is a **repository**?
 - central database of **all documents** and their **history**
- What does it **do**?
 - **Version control**
 - **Configuration management**
 - **Change management**
- **Well-known** software engineering concept.
 - CVS, Subversion, arch, ClearCase, SourceSafe, . . .
- **New**: application to **structured documents**.
 - **Now featuring** distributed repositories, offline authoring.

The MMiSS Repository

- **Fine-grained** approach:
 - Objects: **Packages**, **Units** and **Atoms**;
 - **Relations** between objects (explicit and implicit links).
- **Folders** for grouping of objects.
- **Distributed**: system of communicating repositories.
- Aims: support **Reuse** and **Change Management**:
 - keep track of **dependencies**
 - maintain **consistency** and **completeness**

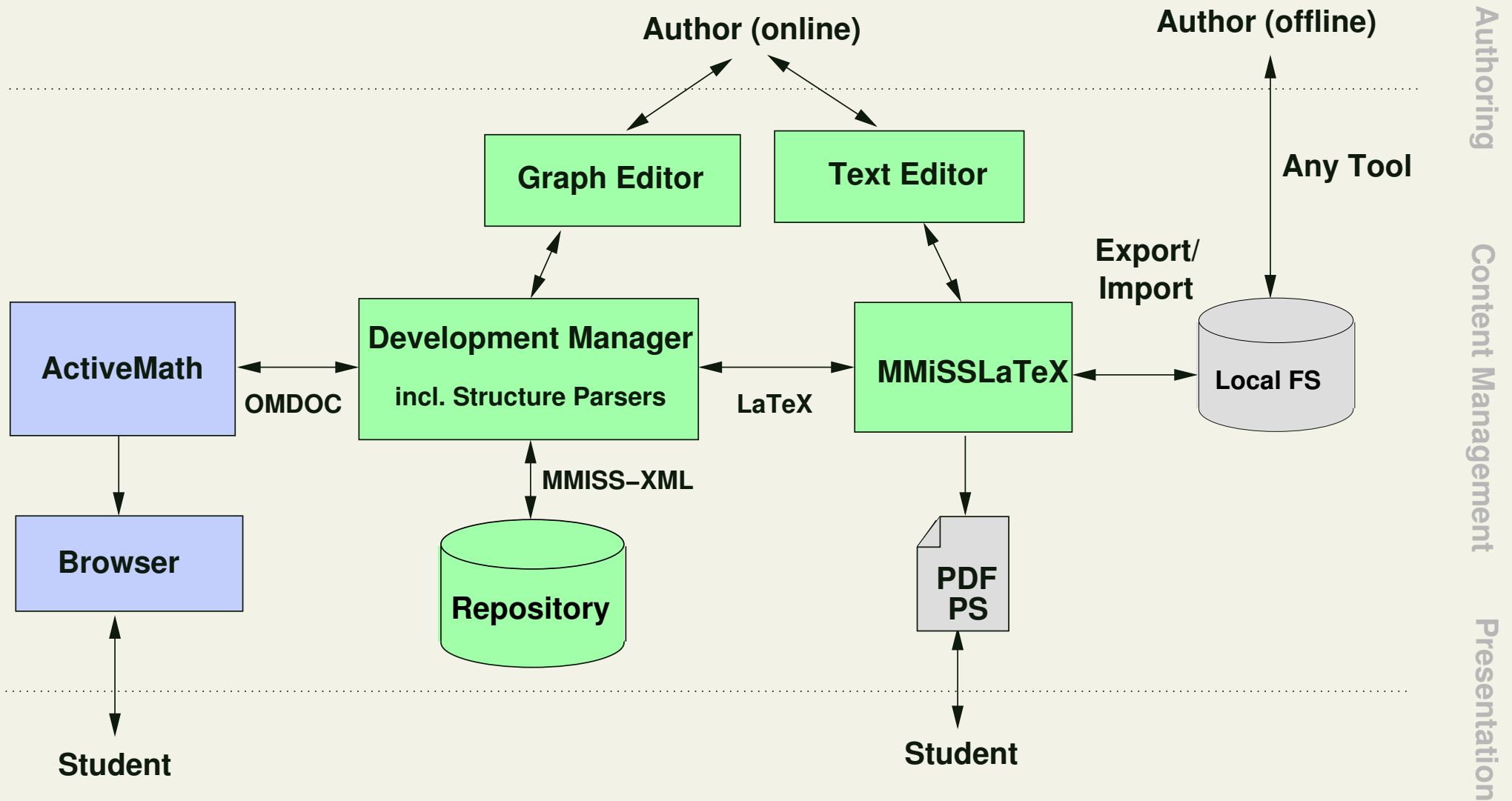
User Interaction with the Repository

- Different versions \rightsquigarrow **version graph**
 - Here: **global** versions.
- Dependencies between objects \rightsquigarrow **structure graph**
 - contains, includes, references

User Interaction with the Repository

- Different versions \rightsquigarrow **version graph**
 - Here: **global** versions.
- Dependencies between objects \rightsquigarrow **structure graph**
 - contains, includes, references
- **Visualisation** of repository as a **graph**
 - dag, starting with **root folder**
- Editing **online** or **offline**.

MMiSS Workbench System Architecture

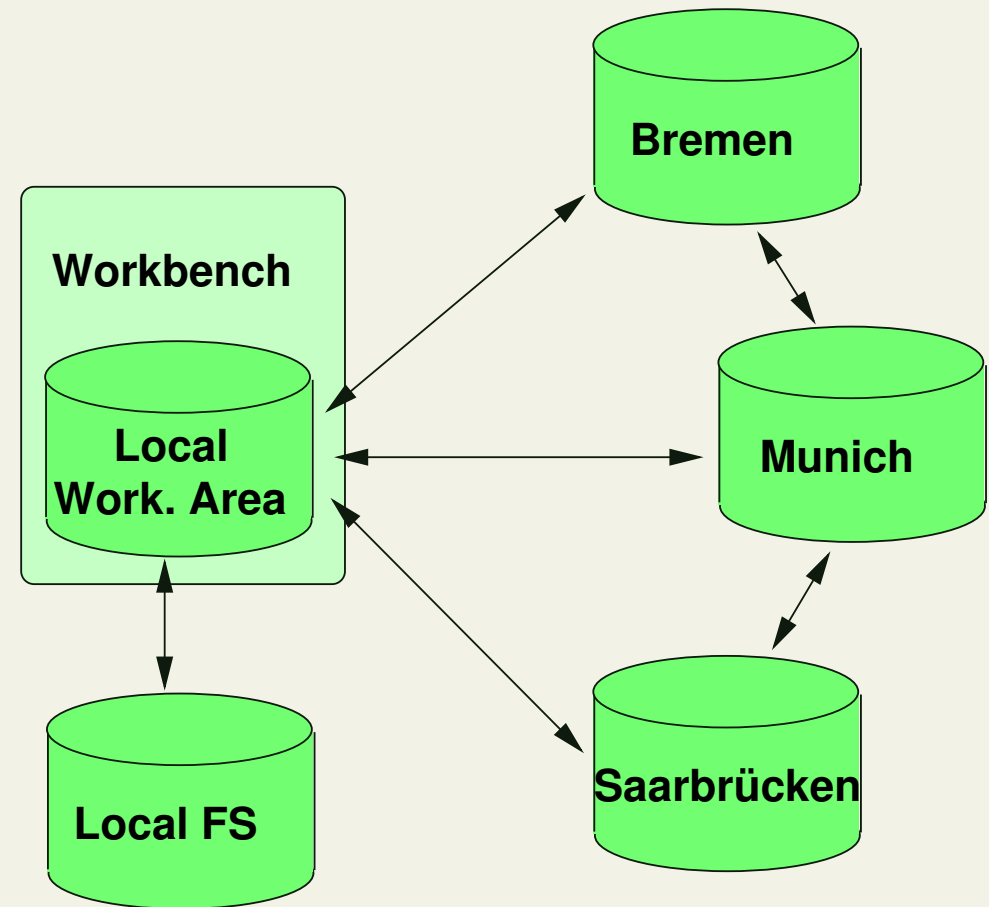


The MMiSS Repository at Work

Start Workbench

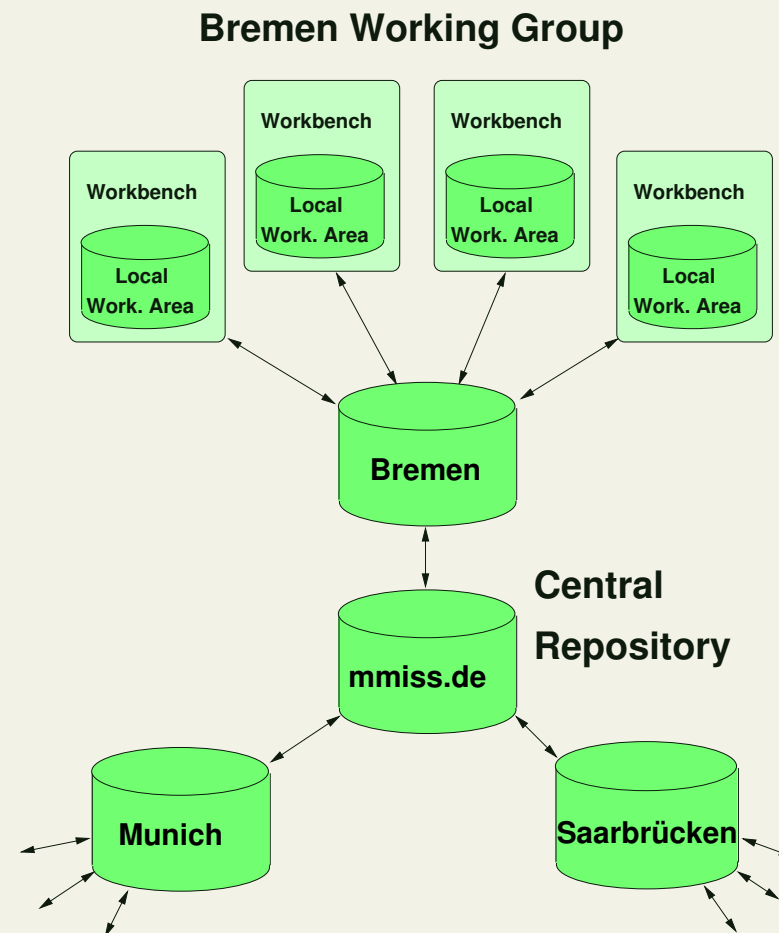
Distributed Development

- User **client**:
 - **offline** via export/import to local file space,
 - **online** via local working area,
 - connects to **remote servers**.
- **Copy versions** between
 - local fs and local working area,
 - local working area and servers,
 - servers
- Flexible setup.



Distributed Development

- User **client**:
 - **offline** via export/import to local file space,
 - **online** via local working area,
 - connects to **remote servers**.
- **Copy versions** between
 - local fs and local working area,
 - local working area and servers,
 - servers
- Flexible setup.



The Implementation

- Implemented in **Haskell** (80 Kloc)
- Based on **events** as central notion of interaction
- **Generic** content model (other XML formats)
- Uses
 - **Tcl/Tk** for user interface (**HTk**)
 - **BerkeleyDB** (<http://www.sleepycat.com>) as database
 - **HaXML** for generic content model
 - **daVinci** for graph visualisation
- Runs on **Linux**, **MacOS X** and **Solaris**
 - Java client for platform independence (in development).

Summary

- Repository holds all **documents** and their **history**.
- Features
 - configuration management and version control,
 - support for **MMiSS \LaTeX** and **OMDoc**,
 - **distributed** development and **offline** editing.
- Aims to support
 - **collaborative** content development
 - **reuse** of content
 - **change management** for content
- Please **download** it at <http://www.mmiss.de>.

Roll Of Honour

Serge Autexier
Ahsan Fayyaz
Achim Mahnke
George Gogvadze
Arne Lindow
George Russell