Cost-Effective Integration of MKM Semantic Services into Editing Environments
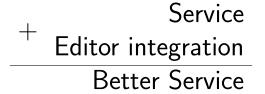
CICM 2012

Constantin Jucovschi

Jacobs University

July 10, 2012

Call for Action

$$\frac{\begin{array}{r} \text{Service} \\ + \quad \text{Editor integration} \end{array}}{\text{Better Service}}$$

Call for Action

$$\frac{\begin{array}{r} \text{Service} \\ + \quad \text{Editor integration} \end{array}}{\text{Better Service}}$$

- new users – moderate learning curve
- moderate & advanced users – increase productivity

Java example

- Namespaces

    import info.kwarc.sissi.doc.spreadsheet.AbstractSpreadsheetDocument

  → info/kwarc/sissi/doc/spreadsheet/AbstractSpreadsheetDocument.java

- Generics

    import java.util.HashMap;
    import java.util.Map;
    import java.util.Map.Entry;

    Map<Entry<Integer, String>, Integer> t =
                new HashMap<Map.Entry<Integer,String>, Integer>();

Is it just boring ...

to create better tools for MKM authors?

- handful of editors to support
    - MKM service authors $\neq$ editor developers
    - editor plugin developers $\neq$ MKM enthusiasts
- version upgrades (both editor and service side)
- multiple operating systems
- communication with services (Java editor with python services)
- integrate with existing extensions (e.g. extend LaTeX mode of TeXentric)

What are your options?

- Editor extensions
- Editor generation frameworks
- Real-Time Document Synchronization and Service Broker (ReDSyS)

Editor extensions

Pro:

- Most editors have one
- Simple things should be simple, complex things should be possible. [Alan Kay]
- Direct power and control

Editor extensions

Pro:

- Most editors have one
- Simple things should be simple, complex things should be possible. [Alan Kay]
- Direct power and control

Contra:

- write once use once
- "complex things" – need to go low level
  color text in line 2 in blue with yellow background in jEdit
  - text color - rewriting line tokens of current mode
  - background color - low level Painter object
- low level – no guaranty of stability
- write Java or go to hell

Editor generation frameworks

Examples: xText, Proxima
Pro:

- Create editors in 30 min
    - syntax highlighting
    - autocompletion
    - validation

Editor generation frameworks
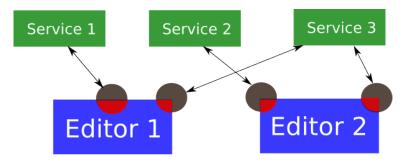
Examples: xText, Proxima

Pro:

- Create editors in 30 min
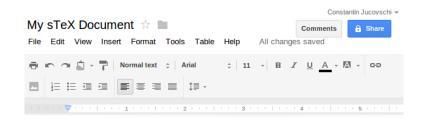 - syntax highlighting
  - autocompletion
  - validation

Contra:

- write once use once
- No CFG – no editor
- few types of customizations possible
- don't go low level

Current state of Editor-Service integrations

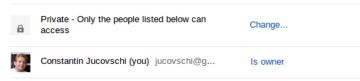Most of existing integrations are 1-to-1 integrations

Proposed Model

My sTeX Document ☆ 📁

Constantin Jucovschi ▾

File Edit View Insert Format Tools Table Help   All changes saved

Comments   🔒 Share

🖨 ↶ ↷ 📋 ▾ 🖌   Normal text ▾   Arial ▾   11 ▾   **B** *I* U̲ A̲ ▾ 🅰 ▾   🔗

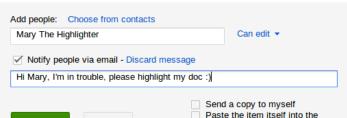🖼   📝 📝 ⇥ ⇤   ▤ ▤ ▤ ▤   ↕☰ ▾
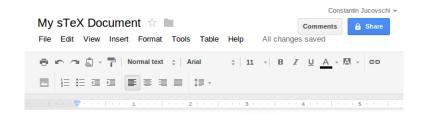
\documentclass{article}

\begin{document}
The \termref{cd=physics-energy, name=grav-potential}{gravitational
    potential energy} of a system of masses $\STRlabel[m1]{m_1}$
and $\STRlabel[m2]{M_2}$ at a distance

Proposed Model

Proposed Model

Proposed Model

```
\documentclass{article}

\begin{document}
The \termref{cd=physics-energy, name=grav-potential}{gravitational
 potential energy} of a system of masses $\STRlabel[m1]{m_1}$
and $\STRlabel[m2]{M_2}$ at a distance $\STRlabel[r]{r}$ using
\termref{cd=physics-constants, name=grav-constant}{gravitational constant}
$\STRlabel[G]{G}$ is $\STRlabel[U]{U}$
\end{document}
```
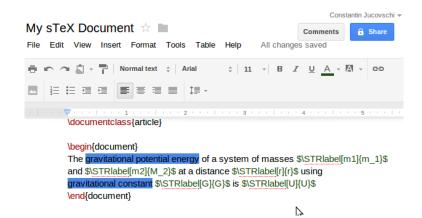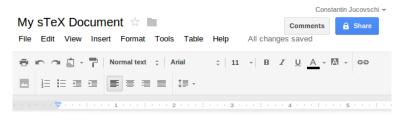
```
\termref{cd=physics-constants,
        name=grav-constant}
   {gravitational constant}
```

Proposed Model

Proposed Model

Proposed Model

```
\documentclass{article}

\begin{document}
The gravitational potential energy of a system of masses $\STRlabel[m1]{m_1}$
and $\STRlabel[m2]{M_2}$ at a distance $\STRlabel[r]{r}$ using
gravitational constant $\STRlabel[G]{G}$ is $\STRlabel[U]{U}$
\begin{equation}
    \STRcopy{U} = -\STRcopy{G}\frac{\STRcopy{m1}\STRcopy{m2}}
                {\STRcopy{r}}+\STRlabel[K]{K}
\end{equation}
\end{document}
```
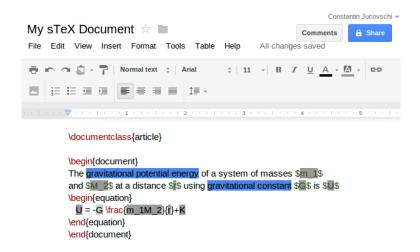
$$\STRlabel[m1]{m\_1}$$

$$\STRcopy{m1}$$

Proposed Model

DEMO

One Document, Many Experts

Experts

- independent, self-contained entities
- provide services editors hardly support
- located anyware in the world
- can go for 5 hours to think
- undo changes of bad experts

Document

- edited in any environment

Integration through the Document

- Direct Editor-Service integration breaks separation of concerns
- Ideally

Editors	Services
change document/properties	
menu/shortcut/toolbar integration	
show properties	configure themselves
fire events	interact with user

Real-Time Document Synchronization and Service Broker

API

- allows multiple entities to change document in real-time (broadcast)
    - insert/delete text/properties
- allows entities to exchange private messages (1-1 messages)
    - autocomplete
    - configure

- limited $\Rightarrow$ stable API
- service/editor independent
- conventions
    - property ("bold",true), means bold
    - event "autocomplete.stex", means autocomplete for sTeX

What promised / What is there / Current version

New repository: https://github.com/jucovschi/ShareJSServices

- RedSyS switched from Etherpad $\rightarrow$ ShareJS
- sTeX Semantic Syntax Highlighting & Term Spotting implemented in Etherpad
- Termref Hider & Transclusion
- Integrated Web Editor (AceEditor)
- jEdit integration worked with Etherpad

Stable/Experimental

Stable parts

- RedSyS
- JavaScript services
- Java document type (jeasysync2 library)

Experiments/in development

- 4 services
- Java $\rightarrow$ ReDSyS connection

How do I write a service

Implement

- changeset onInit(doc)
- changeset onChange(oldDoc, changeset, currentdoc)
- void onEvent(Event)

Changesets are programs

- skip 146 characters
- apply ("underline", true) to next 10 characters
- delete next 4 characters

Conclusion

- $4 \star \rightarrow 5 \star$ service – editor integration
- several ways to do that ...
- document centric approach
    - pays off in the long run
    - better separation of concerns
    - distribution
    - long processing tasks
    - I do editors, you do services
- clone it on github
  https://github.com/jucovschi/ShareJSServices