Freie Universität Berlin

Diplomvortrag

Iterative, prototype-driven development of a whiteboard feature

Michael Jurke
Institut für Informatik
FU Berlin
27. Januar 2011

Overview

Freie Universität Berlin

I. Introduction and motivation
II. Preliminary considerations
III. Prototyping
IV. First graphical prototype using batik
V. Second graphical prototype using GEF
VI. Collaborative XML editing
VII. Concurrency control implementation
VIII. Final structure and Saros integration
IX. Miscellaneous
X. Validation and conclusion

Introduction

Freie Universität Berlin

Pair programming (PP)

- two programmers conduct their work together on one computer
- supposed to
  - improve programming discipline
  - produce better code
  - help to distribute knowledge between programmers

Introduction

Freie Universität Berlin

Distributed pair programming (DPP)

- the programmers are not collocated
  - they have to use tools that
    - replicate local edits on peers sides and
    - offer awareness for a group experience

Introduction

Freie Universität Berlin

Saros

- a plug-in for the IDE Eclipse
- originally a tool for DPP
  - evolved to a tool for collaborative programming by user feedback
- developed in 2006 by R. Djemili
  - continuously enhanced
  - actively supported by the AG Software Engineering

Motivation for a whiteboard feature

Freie Universität Berlin

The mentioned introductory work already:

- evaluated that a whiteboard would be a “nice-to-have” feature
- proposes a possible solution:
  - a *graphical editor* using
  - a *text based model* that
  - applies *replication* in the distributed context

Preliminary considerations

Freie Universität Berlin

I. Introduction and motivation
II. Preliminary considerations
III. Prototyping in the scope of this work
IV. First graphical prototype using batik
V. Second graphical prototype using GEF
VI. Collaborative XML editing
VII. Concurrency control data structure implementation
VIII. Final structure and Saros integration
IX. Miscellaneous
X. Validation and conclusion

Preliminary considerations

Freie Universität Berlin

Textual representation of graphical elements

- Scalable Vector Graphics, SVG standard
  - W3C standard, first published in 2001
  - structured text (XML) describes graphical data
  - basis for the whiteboard models of the instant messengers Coccinella, Gajim and Psi

Preliminary considerations

Freie Universität Berlin

Usage of an external library

- although given the graphical primitives from the system, rendering of graphical data remains complex
  - for comparison: SWT Paint Example
- out of scope of this work
  - to implement feedback, selection and manipulation as well as other editor features manually
  - even impossible to implement whole SVG standard

Prototyping

Freie Universität Berlin

I. Introduction and motivation
II. Preliminary considerations
III. Prototyping
IV. First graphical prototype using batik
V. Second graphical prototype using GEF
VI. Collaborative XML editing
VII. Concurrency control data structure implementation
VIII. Final structure and Saros integration
IX. Miscellaneous
X. Validation and conclusion

Prototyping

Freie Universität Berlin

Prototypes

- incomplete versions of an application
  - to gain knowledge
  - to trigger user feedback
  - to evaluate
    - external components
    - design decisions

Prototyping

Freie Universität Berlin

In practice, this work evaluated:

- graphical components in the Eclipse context
  - batik and GEF
- an approach for collaborative XML editing
  - SXE
- the usefulness
  - of graphical primitives like rectangles
  - of enabling the manipulation of existing elements

  by the feedback of team members

First prototype using batik

Freie Universität Berlin

I. Introduction and motivation
II. Preliminary considerations
III. Prototyping
IV. First graphical prototype using batik
V. Second graphical prototype using GEF
VI. Collaborative XML editing
VII. Concurrency control data structure implementation
VIII. Final structure and Saros integration
IX. Miscellaneous
X. Validation and conclusion

batik

Freie Universität Berlin

- Java framework for SVG
- 2nd most compliant SVG rendering library after Opera
- Swing based
  - graphical library and widget toolkit
  - part of the Java API
  - in contrast to the Standard Widget Toolkit (SWT) that is Eclipse based on
    - Batik UI components have to be embedded in SWT

Prototype results

Freie Universität Berlin

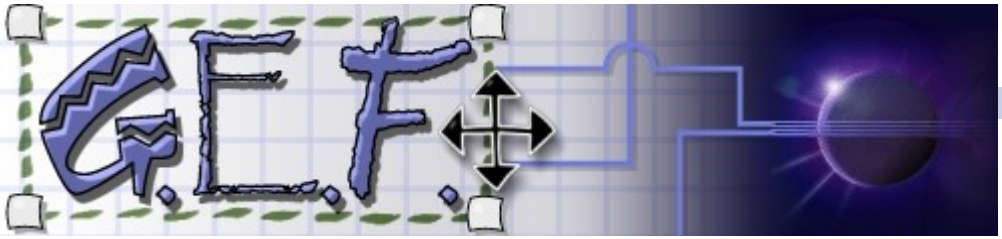
- Embedding batik in SWT caused unresolvable problems
  - performance issues and rendering got stuck
  - presumably because of the double threaded nature of embedding to different widget toolkits
- the prototype had to be discarded

Second prototype using the Graphical Editing Framework

Freie Universität Berlin

I. Introduction and motivation
II. Preliminary considerations
III. Prototyping in the scope of this work
IV. First graphical prototype using batik
V. Second graphical prototype using GEF
VI. Collaborative XML editing
VII. Concurrency control data structure implementation
VIII. Final structure and Saros integration
IX. Miscellaneous
X. Validation and conclusion

Graphical Editing Framework



- Eclipse project for graphical editing
- 10 years of active development
- strictly following the Model-View-Controller pattern

Graphical Editing Framework

Freie Universität Berlin

- offers graphical editor components:
  - editor features: drag and drop and a tool palette
  - direct editing by plug-able edit policies
  - request/command infrastructure for editing and enabling undo/redo
  - Draw2d subsystem for graphical primitives (figures)
- extenders only have to provide the application specific details (model, view, controller, edit policies...)

Graphical Editing Framework

Freie Universität Berlin

Main missing components:

- freehand drawing
- collaborative editing

GEF prototype

Freie Universität Berlin

- extends GEF by freehand drawing
  - specialized request, command and edit policy
- provides whiteboard specific models, controllers and commands
- embeds an editor in an Eclipse view part

- A Saros whiteboard view

GEF prototype

Freie Universität Berlin

Presentation I

GEF evaluation

Freie Universität Berlin

- technically:
  - GEF is perfectly feasible for the whiteboard purposes
- by team members:
  - all testers preferred to use a rectangular shape instead of drawing it freehand
  - they preferred to rearrange/delete existing elements instead of erasing them
  - undo/redo facilities are useful
  - most important feature missing: maintain a synchronized state

Collaborative XML editing

Freie Universität Berlin

I. Introduction and motivation
II. Preliminary considerations
III. Prototyping in the scope of this work
IV. First graphical prototype using batik
V. Second graphical prototype using GEF
VI. Collaborative XML editing
VII. Concurrency control data structure implementation
VIII. Saros integration
IX. Miscellaneous
X. Validation and conclusion

Collaborative XML editing

Freie Universität Berlin

How to synchronize hierarchical data?

- modifications have requirements, like
  - target or parent must exist
  - parent must exist
  - there must not be a cyclic relationship in the hierarchy
  - XML specific (must be “well-formed”)
    - attribute names are unique etc.

Research

Freie Universität Berlin

- Scientific research in the recent years, most to mention:
  - the treeOPT algorithm by C. Ignat

    Ignat, C.-L.; Norrie, M. C., Customizable Collaborative Editor Relying on treeOPT Algorithm, 2003
  - the Collaborative Editing Framework for XML (CEFX) by A.R.S. Gerlicher

    Gerlicher, A.R.S., Developing Collaborative XML Editing Systems, 2007
    - extended by M. Voigt and ported to Android by D. Hering
    - The work from D. Hering contains a comparison of the current state of the art

    Hering, D., Entwicklung eines Dienstes für Real-time Collaborative Editing für die Mobilis-Plattform, 2009

XMPP standardization efforts

Freie Universität Berlin

- There was standardization effort for whiteboarding over XMPP from the very beginning
  - a lot of proposals, only one became experimental in June 2010:
- Shared XML Editing (XEP-0284)
  - originated in the Coccinella whiteboard protocol
    - improved and sent to XMPP by J. Govenius
  - obsolete versions are used in Gajim and Psi

Collaborative XML editing

Freie Universität Berlin

Why do we need a specialized approach?

Or: Why can we not use the existing concurrency control algorithm?

- see requirements before
- An XML as part of the application will be accessed as parsed DOM tree
  - serializing to XML, applying the edit and parsing again
    - would lead to bad performance
    - could cause problems because of implementation specific parsing (attribute order is not specified in XML)

Freie Universität Berlin

Why do we need a specialized approach?

Or: Why can we not use the existing concurrency control algorithm?

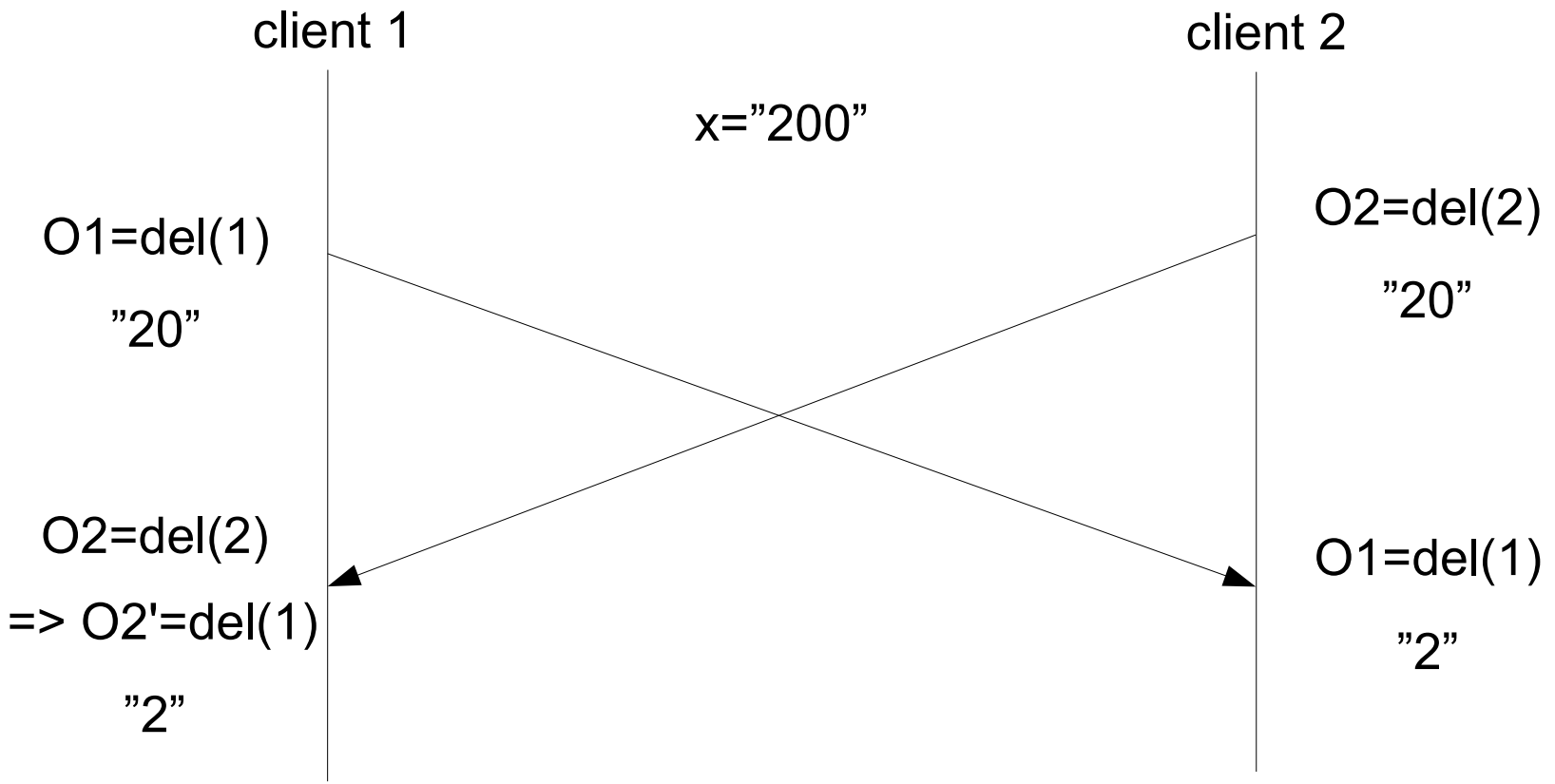
- OT in this context may not make sense
  - context dependent:
    - still applicable for structured text
    - but maybe not for application data:

      empty attributes or unintended values may be created

Collaborative XML editing

Freie Universität Berlin

- unintended values with OT



Collaborative XML editing

Freie Universität Berlin

treeOPT

- a recursive application of a OT algorithm for linear data
- very little information, not guaranteed if convergence is ensured in all cases

CEFX

- evaluated by M. Voigt and D. Hering
  - difficult to understand
  - very difficult to extend
  - problems with late join

Collaborative XML editing

Freie Universität Berlin

SXE

- used in instant messengers
- experimental standard for an XMPP Extension
- introduces mechanics to reduce complexity

- Preferable for Saros

SXE

Freie Universität Berlin

I. Introduction and motivation
II. Preliminary considerations
III. Prototyping in the scope of this work
IV. First graphical prototype using batik
V. Second graphical prototype using GEF
VI. Collaborative XML editing
VII. Concurrency control data structure implementation
VIII. Saros integration
IX. Miscellaneous
X. Validation and conclusion

SXE

Freie Universität Berlin

Basic idea: records, where

- new-records correspond to DOM nodes
- set-records correspond to the local history
  - the history of every single node
- remove-records remove an existing record from the tree

SXE

Freie Universität Berlin

A sample edit to add a polyline and a points attribute:

```
<sxe id="2" session="4840" xmlns="urn:xmpp:sxe:0">
    <new primary-weight="0" parent="root"
        visible="true" rid="1" name="polyline"
        type="element" version="0">
    </new>
    <new primary-weight="1" parent="1"
        visible="true" chdata="390,1110 400,1120
        390,1110 420,1150" rid="2" name="points"
        type="attr" version="0">
    </new>
</sxe>
```

SXE

Freie Universität Berlin

A sample edit to add a polyline and a points attribute:

```
<sxe id="2" session="4840" xmlns="urn:xmpp:sxe:0">
    <new primary-weight="0" parent="root"
        visible="true" rid="1" name="polyline"
        type="element" version="0">
    </new>
    <new primary-weight="1" parent="1"
        visible="true" chdata="390,1110 400,1120
        390,1110 420,1150" rid="2" name="points"
        type="attr" version="0">
    </new>
</sxe>
```

SXE

Freie Universität Berlin

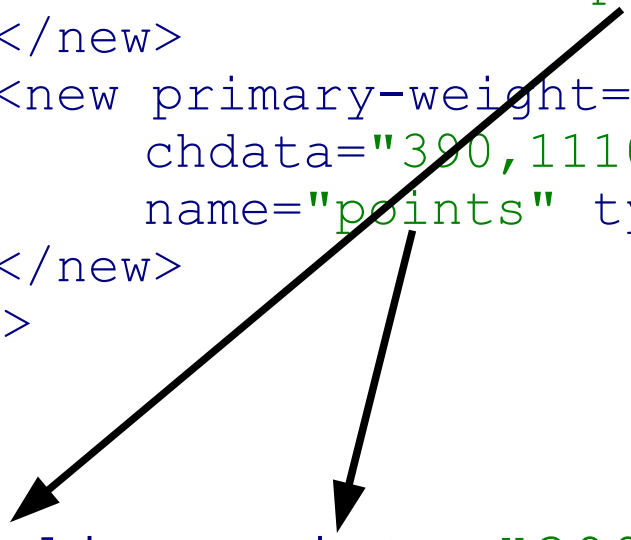
A sample edit to add a polyline and a points attribute:

```
<sxe id="2" session="4840" xmlns="urn:xmpp:sxe:0">
    <new primary-weight="0" parent="root"
        visible="true" rid="1" name="polyline"
        type="element" version="0">
    </new>
    <new primary-weight="1" parent="1"
        visible="true" chdata="390,1110 400,1120
        390,1110 420,1150" rid="2" name="points"
        type="attr" version="0">
    </new>
</sxe>
```

SXE

Freie Universität Berlin

```
<sxe id="2" session="4840" xmlns="urn:xmpp:sxe:0">
     <new primary-weight="0" parent="root" visible="true"
          rid="1" name="polyline" type="element" version="0">
     </new>
     <new primary-weight="1" parent="1" visible="true"
          chdata="390,1110 400,1120 390,1110 420,1150" rid="2"
          name="points" type="attr" version="0">
     </new>
</sxe>
```



```
<polyline points="390,1110 400,1120 390,1110
420,1150"/>
```

SXE - conflicts

Freie Universität Berlin

Assumption:

- Conflicts can only occur due to concurrent set-records

Conflict detection

- by version field
  - the received set-record must have its target version+1

Conflict resolution:

- conflicting edits are undone

SXE - enhancements

Freie Universität Berlin

- made P2P ready
  - idea can be described by “causal readiness” (from treeOPT algorithm)
  - non causal ready records are queued and applied as soon as applicable
  - simpler solution than using a Multi-User Chat (XMPP chat room)
  - performs better than having the host relaying the messages

SXE - enhancements

Freie Universität Berlin

- visible mechanic
  - idea from WOOTO algorithm
    Molli,P.; Oster, G.; Urso,P.; Imine, A.
    Data Consistency for P2P Collaborative Editing, 2006
  - instead of removing records, they are marked “invisible”
  - original cause: missing undo support of SXE and problems with the command stack

SXE - enhancements

Freie Universität Berlin

- visible mechanic
  - facilitates undo/redo a lot, references can be maintained
  - solves cases where SXE did not ensure convergence
  - reduces complexity of applying a remove-record
    - removes some possible causes for errors
  - similar approach used by T. Pusateri, another implementer of SXE, to enable a playback function
    - *http://jabberpad.net/*
    - uses the SVG “hidden” attribute
    - confirmed in January 2011 only

Presentation II

SXE – problems

Freie Universität Berlin

- well-formedness of the XML
  - cyclic relationships are intercepted but cause a divergence at current state
- start-synchronization may result in a divergent state if a last record of a local history had conflicted

SXE – missing features

Freie Universität Berlin

- Blocking of elements and subtrees
- re-synchronize after a disconnect
- validate an XML
- free memory
  - clear history after some time

Saros integration

Freie Universität Berlin

I. Introduction and motivation
II. Preliminary considerations
III. Prototyping in the scope of this work
IV. First graphical prototype using batik
V. Second graphical prototype using GEF
VI. Collaborative XML editing
VII. Concurrency control data structure implementation
VIII. Saros integration
IX. Miscellaneous
X. Validation and conclusion

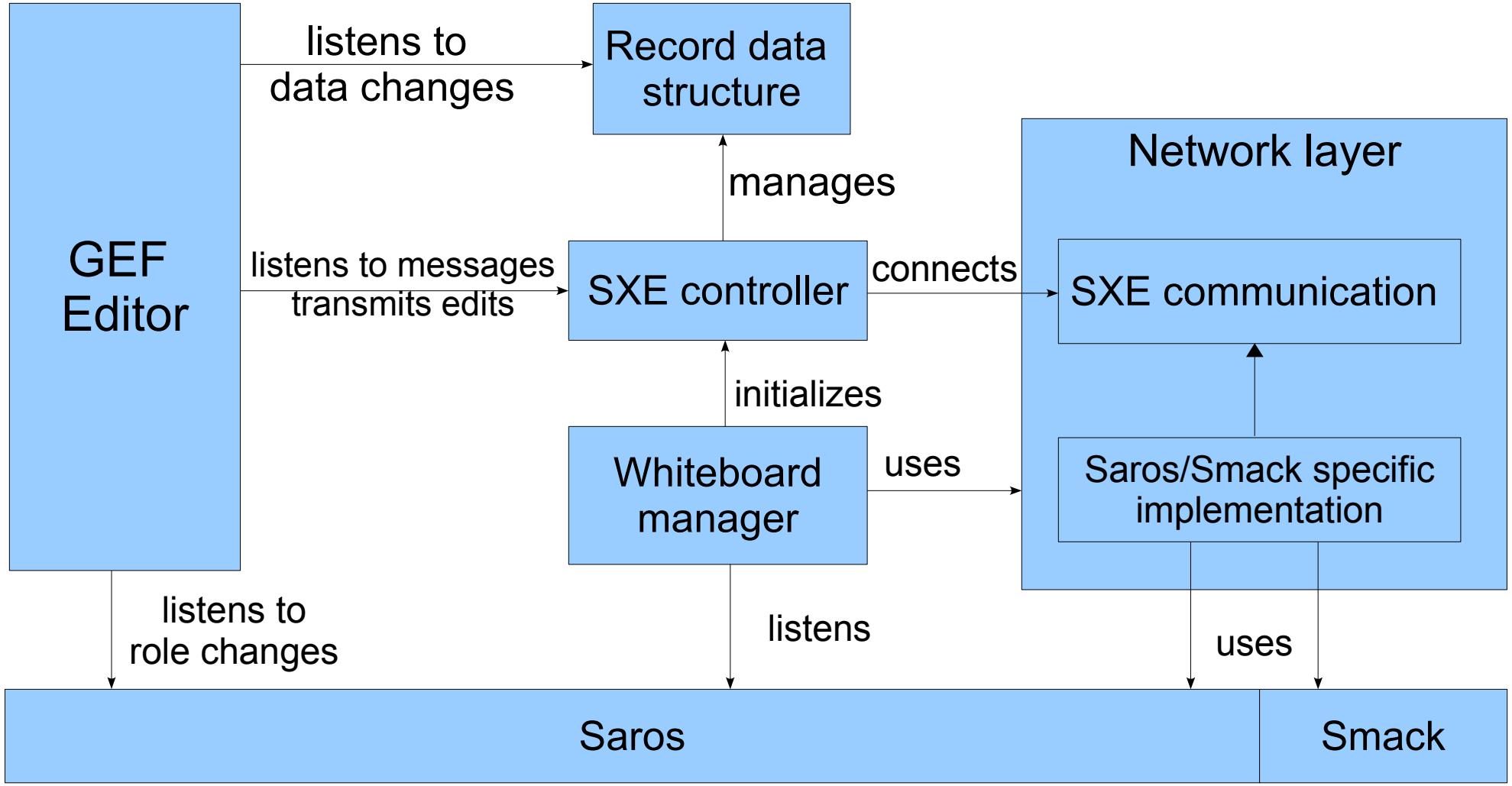
Saros integration

Freie Universität Berlin

- Realized as separate project on top of the Saros session
  - does not interfere with other parts of Saros
  - made the work easier
    - could use Saros head revision all the time
  - no extra time needed to merge a separate branch
  - made it less likely to introduce unnecessary interconnectivity
  - a prototype for the coming refactoring to multiple Saros projects (Januar 2011 only)

Final architecture

Freie Universität Berlin



Miscellaneous

Freie Universität Berlin

I. Introduction and motivation
II. Preliminary considerations
III. Prototyping in the scope of this work
IV. First graphical prototype using batik
V. Second graphical prototype using GEF
VI. Collaborative XML editing
VII. Concurrency control data structure implementation
VIII. Saros integration
IX. Miscellaneous
X. Validation and conclusion

Network refactoring

Freie Universität Berlin

- Refactor the Saros bytestream infrastructure to use the improvements of the Smack API implemented by H. Staib
  - use SOCKS5 bytestreams instead of Jingle File Transfer
  - implement the IBB fallback in Saros if no SOCKS5 bytestream could be established
- Refactor the network interface to offer a method with abstraction to the sending mechanic (bytestream, chat)
  - used by the whiteboard
    - required to send messages bigger than the maximum XMPP packet size

Bug fixing (selection)

Freie Universität Berlin

- help to fix a bug in the Smack API improvements
- leaving a session didn't cancel running invitations
- exaggerated invitation delay on non-windows platforms
- fix the concurrency control implementation
- fix IBB or mediated SOCKS5 connection after reconnect

Validation and conclusion

Freie Universität Berlin

I. Introduction and motivation
II. Preliminary considerations
III. Prototyping in the scope of this work
IV. First graphical prototype using batik
V. Second graphical prototype using GEF
VI. Collaborative XML editing
VII. Concurrency control data structure implementation
VIII. Saros integration
IX. Miscellaneous
X. Validation and conclusion

Validation - just to mention:

Freie Universität Berlin

- CEFX was a doctor thesis of 3 years
  - it was not error-free
    - some errors fixed by M. Voigt
    - not well-formed XML documents still throw uncaught exceptions
- the work on treeOPT took several years
    - the publicly available documents leave it unclear how to handle non well-formed documents
- The whiteboards of Coccinella, Psi and Gajim
  - do not allow nesting of elements
  - do not implement undo/redo
  - Psi, Gajim: are not very stable

Validation

Freie Universität Berlin

- The work of D. Hering:
  - could not implement late join (start synchronization) because of problems with CEFX
  - does not support modification or removal of elements

- The provided implementation is an adequate solution, maybe even more than one could have expected

Conclusion

Freie Universität Berlin

This work

- evaluated different graphical libraries in the context of Eclipse
- provided a prototype and an improved version of the experimental XMPP Extension SXE
- provided a prototype that tries to fulfil l the idea of separation of concerns by embedding a new feature in a new project

Conclusion

Freie Universität Berlin

This work

- results in a Saros whiteboard view, complying the initial task
- helped to improve Saros by bug fixing and refactoring

Thanks!