Collaboration Tool Examples

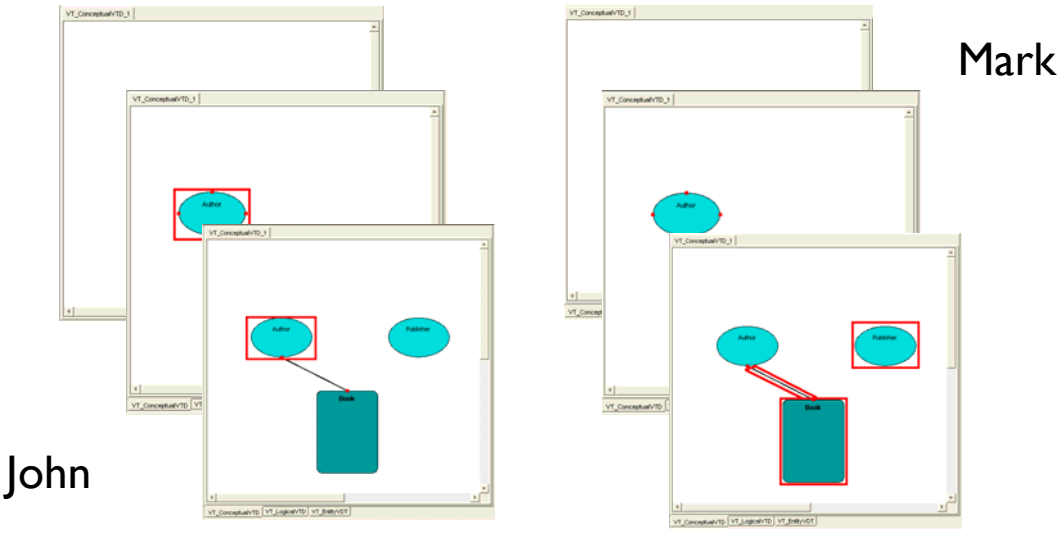
- Pounamu-based collaboration tools
  - Synchronous diagram editing
  - Group awareness
  - Architecture/implementation Issues
- Marama-based collaboration tools
  - Diagram differencing & merging
  - Sketch annotation (for collaboration)
  - Future plans…
- Evaluation of collaborative DSVL tools

See: Mehra, Grundy, Hosking - A Generic Approach to Supporting Diagram Differencing and Merging for Collaborative Design – Proc. IEEE/ACM Automated Software Engineering 2005

Collaborative Editing – Some Key Issues

- What artefacts to support for?
  - Code, diagrams, tests etc
- What kinds of collaborative editing should we support?
  - Synchronous – same time
  - Asynchronous – different time
- How do we add support to existing tools?
  - Build into tool from the start
  - Plug-and-play component support?

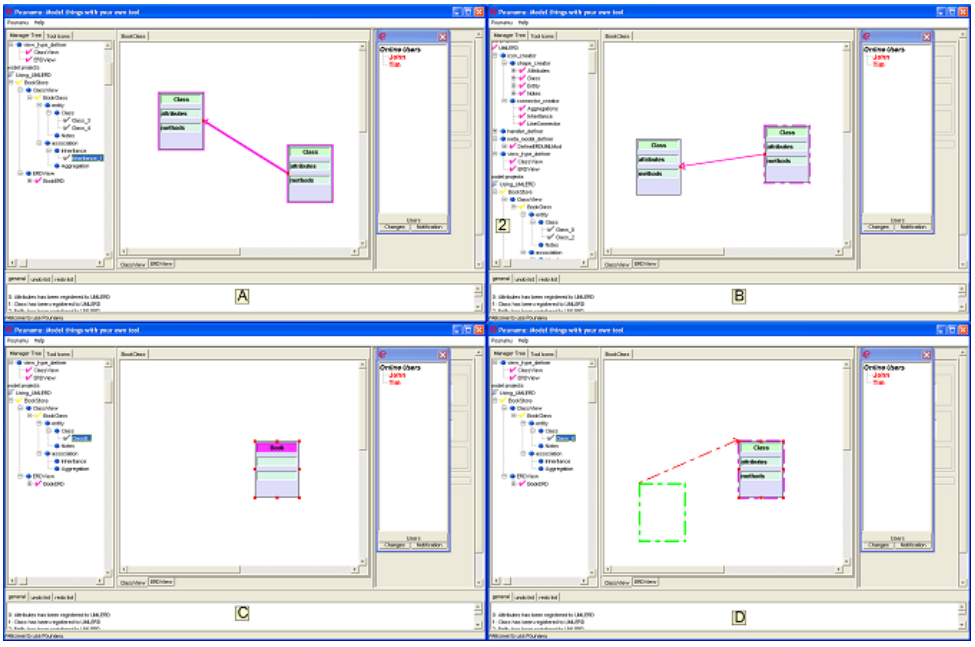
Pounamu Example



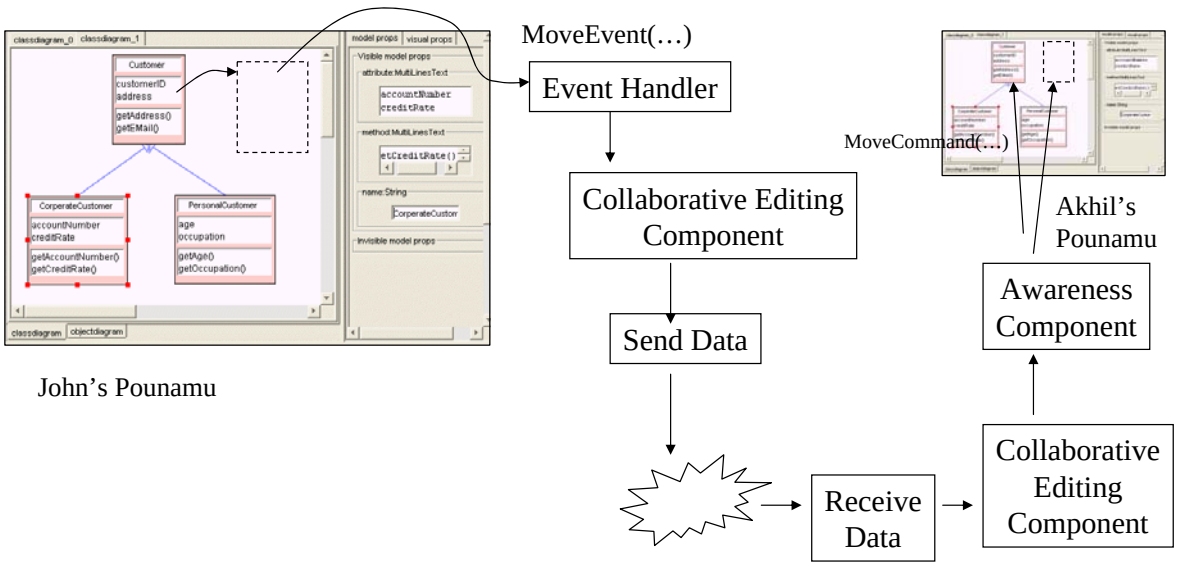
Awareness Support

- Want to keep others "aware" of what doing to your version of artefact (eg diagram, code, documentation, wiki, …)
- Various approaches:
  - Multiple cursors, scroll bars
  - "Area of interest"
  - Annotation/highlighting of changes, locked areas
  - "Intention" indicators e.g. menu just selected highlighted
  - "Progressive" highlights e.g. fade away

Pounamu Example



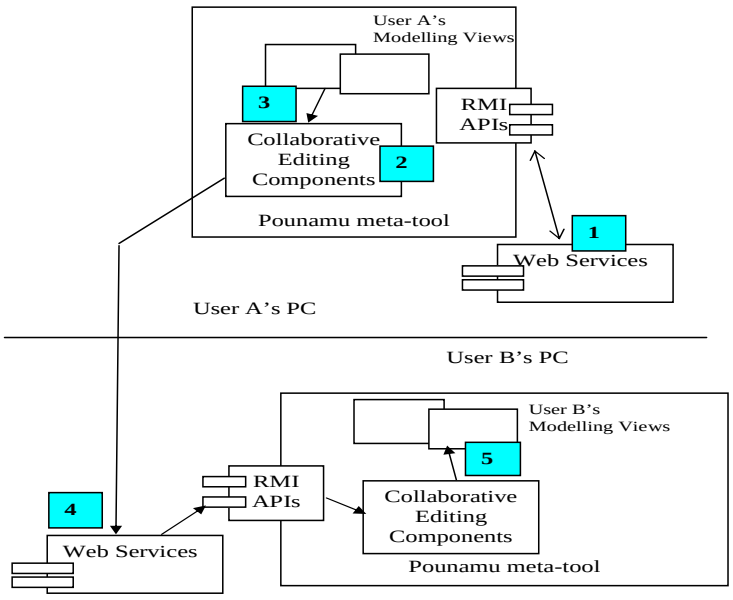
Pounamu Collaborative Editing & Group Awareness



Pounamu Collaborative Editing Components

- RMI API, Web Services API (via SOAP) to this
- Notification, event sender/receiver components
  - Provides messaging infrastructure between tools
- Collaboration Component
  - Co-ordinates sending/receiving for collaborative work
- Group awareness components
  - Provide additional support e.g. locking, highlighting
- Peer-to-peer or client-server supported

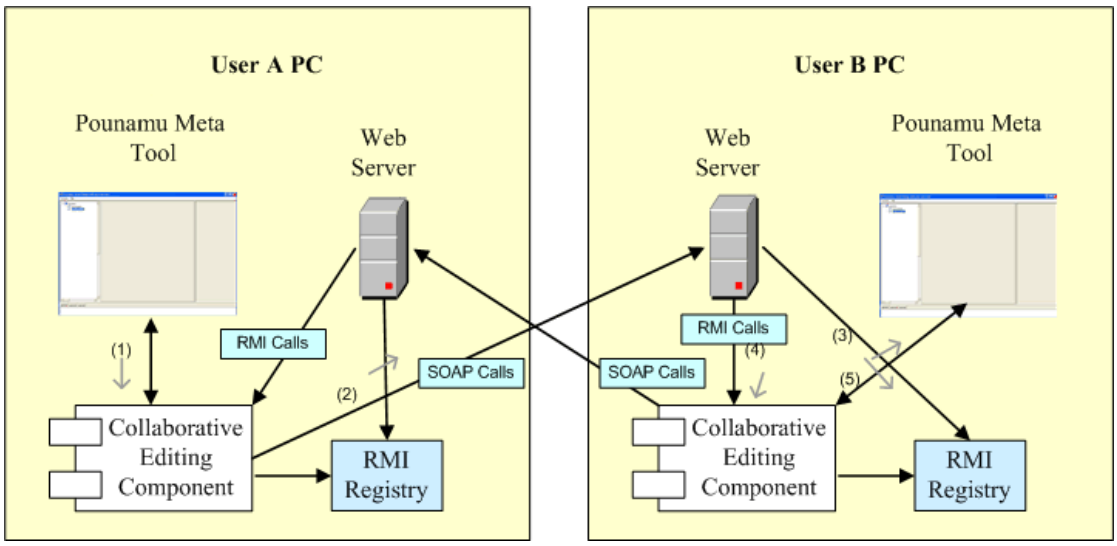
Architecture



Basic Architecture

- User requests collaboration facilities be added
- Dynamically plug-in collaborative work and sender/receiver components
- Discover other available users
- Communicate via events, XML encoding of diagram contents
- UDDI registry to discover/WSDL component descriptions
- SOAP-based protocols

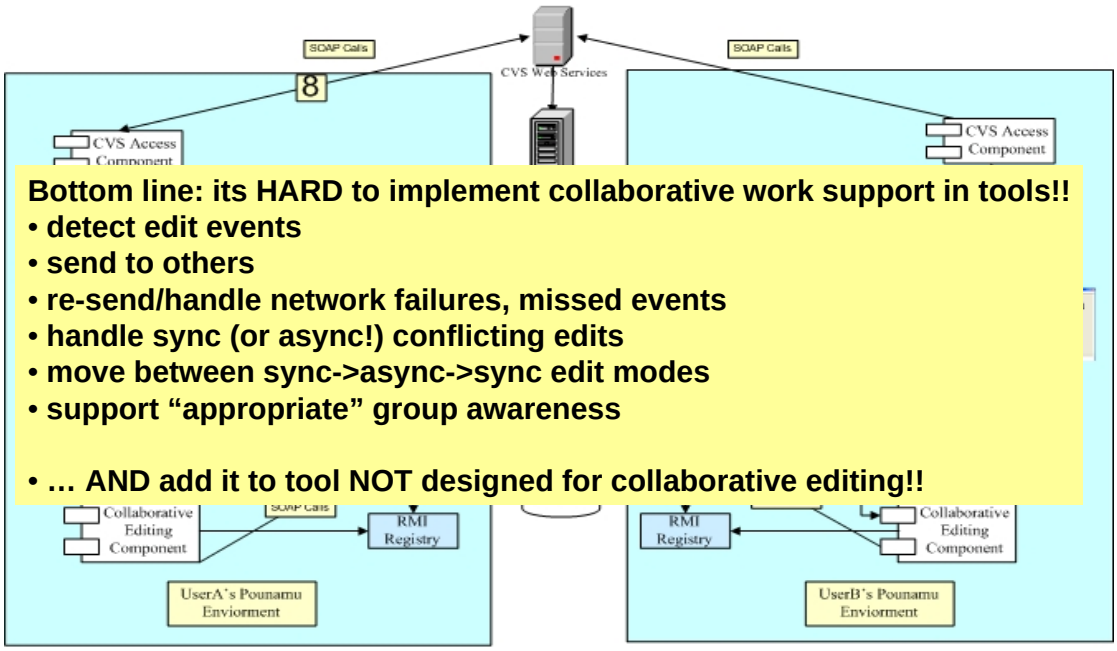
How it works (P2P)



Implementation

- Uses Pounamu's RMI API to get/set XML, detect events, run editing commands
- SOAP web services API as transport mechanism between tools
- Plug-in components to provide send/receive, collaborative work co-ordination, additional group awareness facilities
- Can dynamically deploy collaboration, switch different group awareness facilities on/off
- Can dynamically discover other collaborators

Implementation Overview

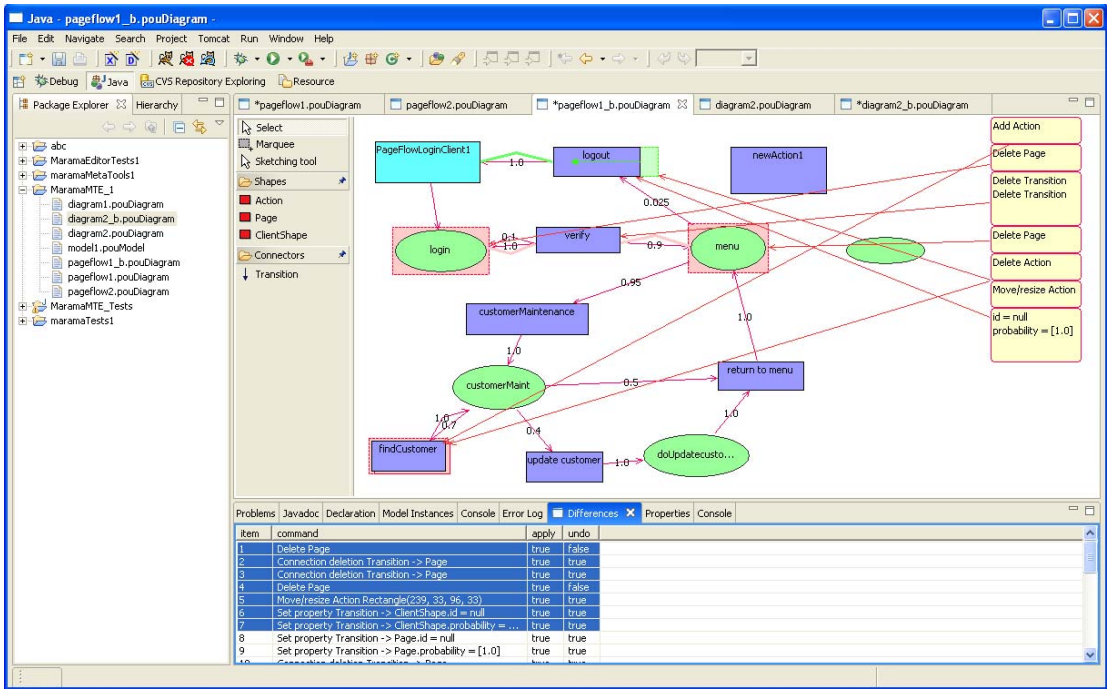


Bottom line: its HARD to implement collaborative work support in tools!!
- **detect edit events**
- **send to others**
- **re-send/handle network failures, missed events**
- **handle sync (or async!) conflicting edits**
- **move between sync->async->sync edit modes**
- **support "appropriate" group awareness**
- **… AND add it to tool NOT designed for collaborative editing!!**

Marama Diagram Differencing

- To support collaborative, asynchronous work we need:
  - Support for multiple versions of software artifacts & configuration management
  - Ability to compare versions ("differentiation")
  - Ability to combine versions ("merging")
- Good support exists for textual & XML-based versioning differentiation and merging
- Limited or no support for visual design artifact versioning, differentiation and merging
- Wanted to add such support to a design environment meta-tool for use across wide variety of design tools

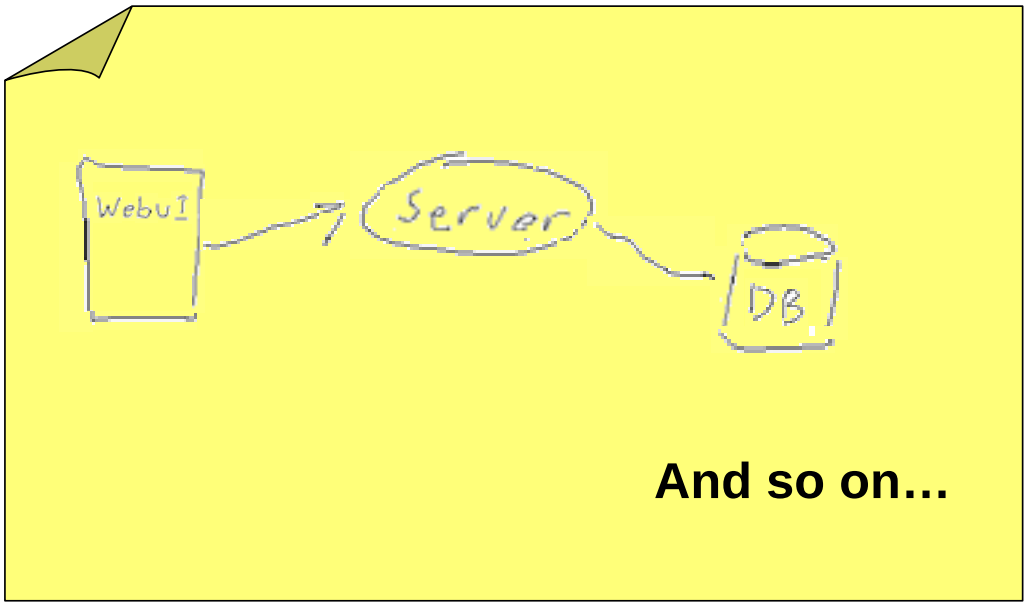
Example



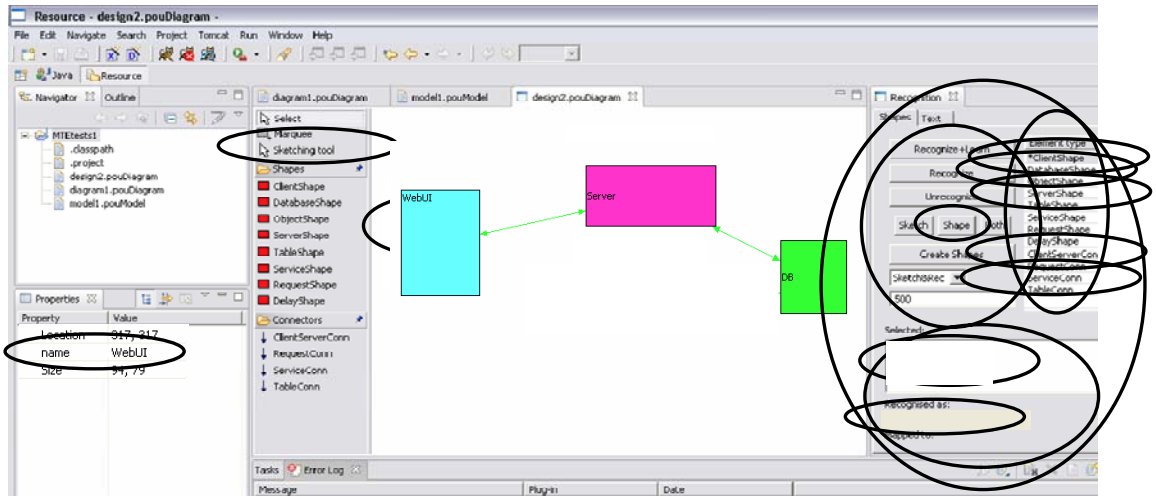
Differencing Algorithm

- Differencing does 2 passes over diagram datastructure – shape then connector comparison
- Uses a "root ID" for each shape to determine which items in two versions share same root version
- Compares attribute values
- Compares position, size
- Distinguishes "contains" and "related to" connectors
- Builds Create/DeleteShape, SetProperty, MoveShape, ResizeShape, Create/DeleteConnector Commands
- Highlighting temporarily annotates diagram with Command list info

Marama – sketch annotation

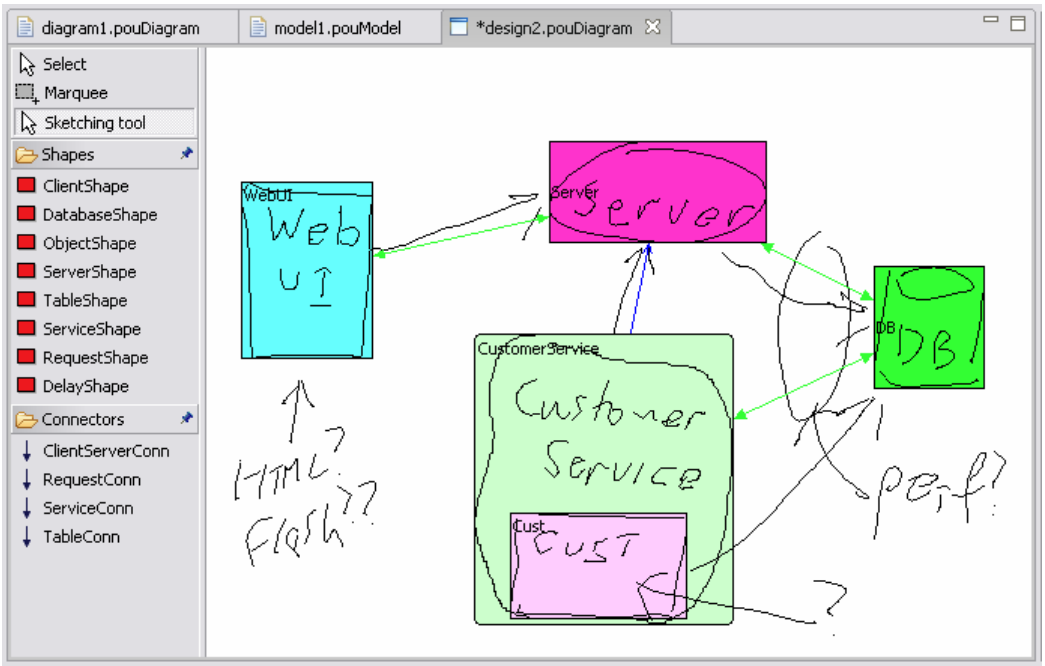


Example – Sketch & formalise

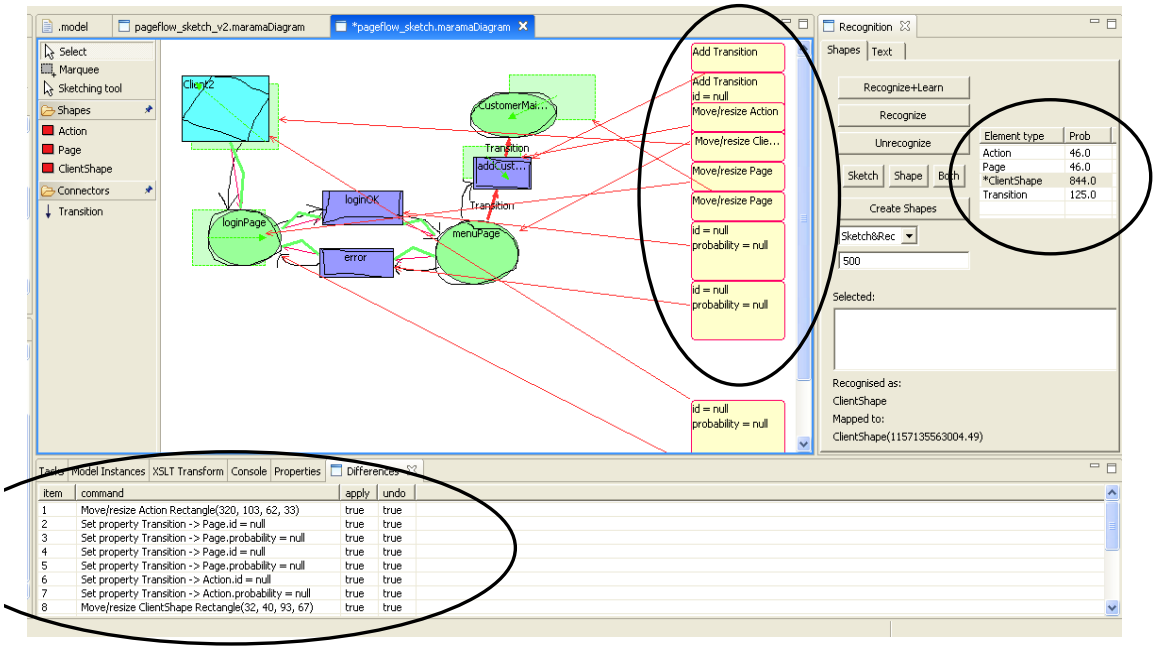


How cool is that…?! ☺

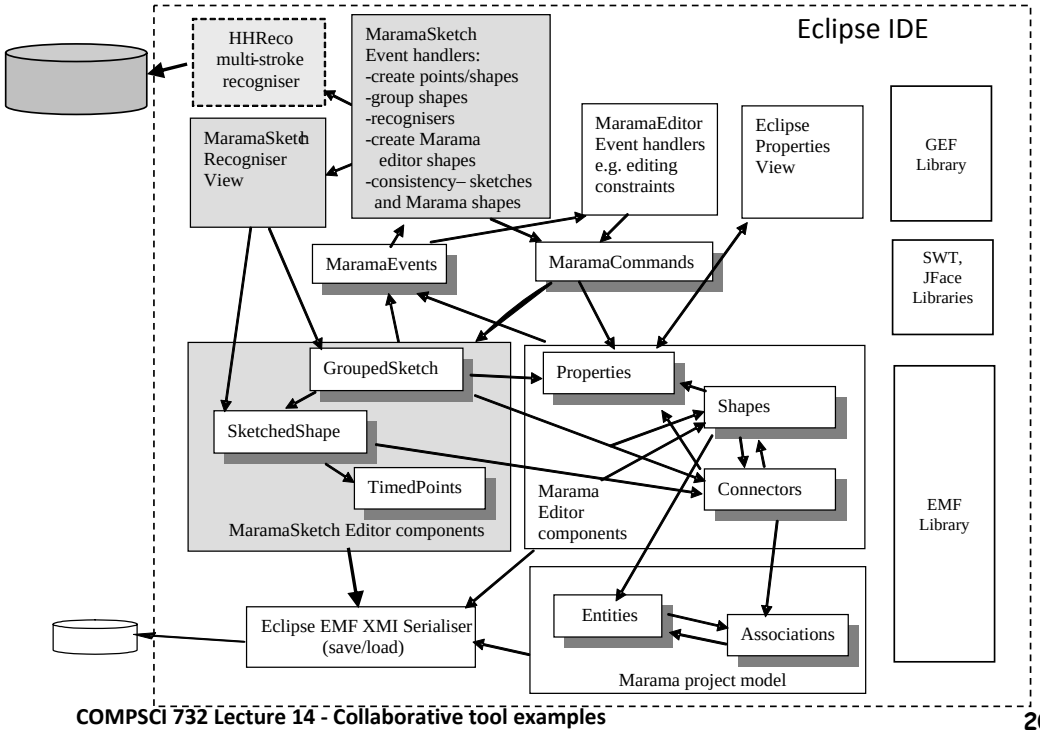
Review/Collaboration



Diffing/Other view types



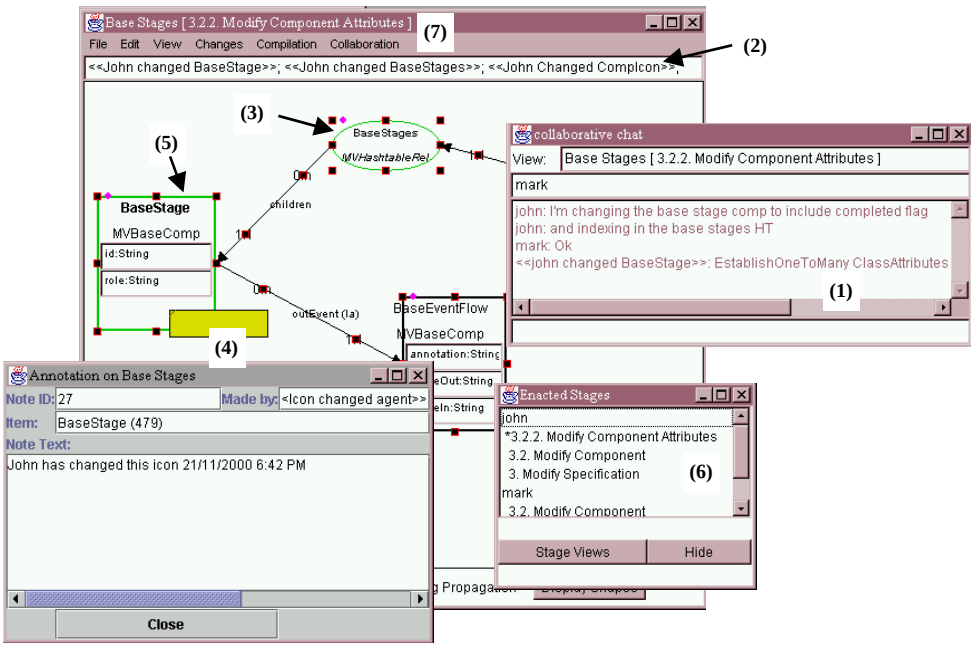
How its done…



Marama – Collaboration Future Work

- More group awareness/collaboration facilities:
  - Indicate to other users each collaborator's identification, role, task
  - Notes, annotations, mark-up e.g. sketching over diagrams
  - Locking during editing of items
  - Editing histories captured and viewable
  - Version merging (of XML-based view structures)
  - Remote CVS repository for view versions/histories
  - Voice-driven edits; replay; highlight etc ☺
  - Gesture-based e.g. touch screen, Eye toy… ☺ ☺
- Infrastructure:
  - 3rd party tool collaboration support
  - Reusable notification services, end user configurable event handling for collaborative work support

Earlier Examples (JViews)…



Evaluations

- Evaluating software tools is hard
- Evaluating DSVL software tools is very hard
- Evaluating collaboration support in DSVL-based software tools is VERY, very hard!
- Issues:
  - Set up experiment, choose users/tasks
  - Run experiment
  - Collect and analyse results
- JViews – plug-in collaborative editing components
- Pounamu – group awareness
- Marama – differencing

MaramaDiffer - Evaluation

- Usability analysis via survey and Cognitive Dimensions
  - Visibility – displays differences in-situ
  - Viscosity – user can accept/reject chanages directly
  - Hidden dependencies – reduces
  - Consistency – of change representation/acceptance
  - Error-proneness and hard mental operations – reduced
- Gutwin's groupware assessment framework
  - Presence & authorship – clear
  - Awareness of change – explicit
  - Intention awareness – some support