Collaboration via the model repository

David Nickerson
PMR2 – Andre's view

- Website
  - exposure pages
  - predefined model listings (keywords)
- Database
  - some metadata pulled from models
- Collection of independent mercurial repositories
  - workspaces
  - can store arbitrary data, not just CellML XML documents!
Mercurial

- Each workspace is a standalone mercurial repository
  - distributed version control system (DVCS)
  - no central repository
- Mercurial repositories can be used independently from PMR2
  - best to create them using PMR2
Typical workflow

- **create workspace**
- **clone**
- **make changes & commit**
- **push**
- **expose**
Embedded workspaces

● Mercurial subrepos

● Assemble workspaces by pulling together workspaces
  ● at specific points in the workspace's evolution
  ● from other authors/owners
  ● plug-and-play
“Advanced” workflow

PMR2

- create workspace
- clone
- embed

Andre

- make changes & commit
- push
- expose

CellML Workshop 2010, 24-26 February 2010, Auckland
“Advanced” workflow

PMR2
- embed
- clone
- create workspace
- expose
- push
- pull
- push
- merge

Andre
- make changes & commit

David
- make changes & commit
Collaborating via the model repository

- Background technology all in place.
- Works well now for people familiar with and happy to use mercurial directly.
- What about everyone else?
  - extend PMR2 to provide a range of web services to allow better integration with tools?
  - turn PMR2 into a heavyweight feature rich web application?
Conclusion

- Collaborating via the model repository pretty much possible – at the mercurial level.
- Collaborating via the model repository still needs work to hide the details of mercurial
  - managing embedded workspaces.