Hands-on experiences using Collaborative Protégé (CP)

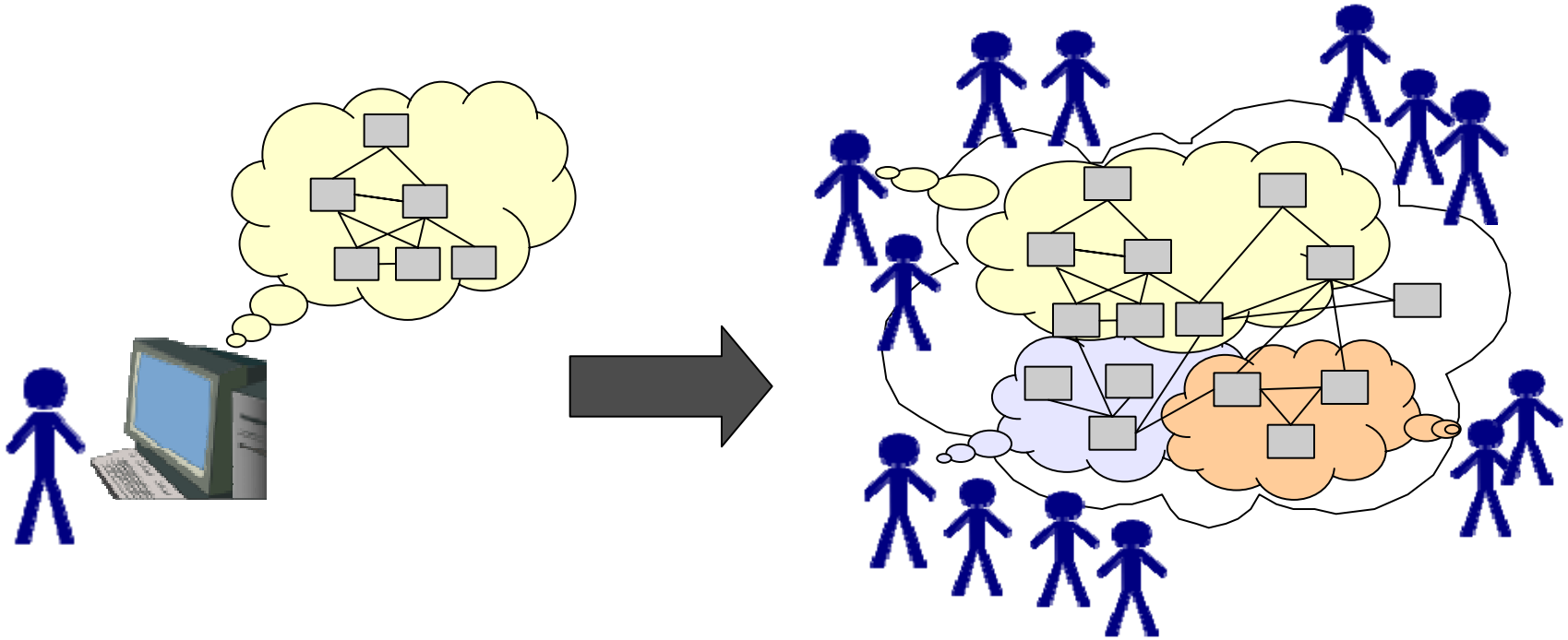
Daniel Schober, UKLFR

UNIVERSITÄTS KLINIKUM FREIBURG

Institut für Medizinische Biometrie und Medizinische Informatik

Arbeitsgruppe Medizinische Informatik

Paradigm shift Collaborative Ontology Editing



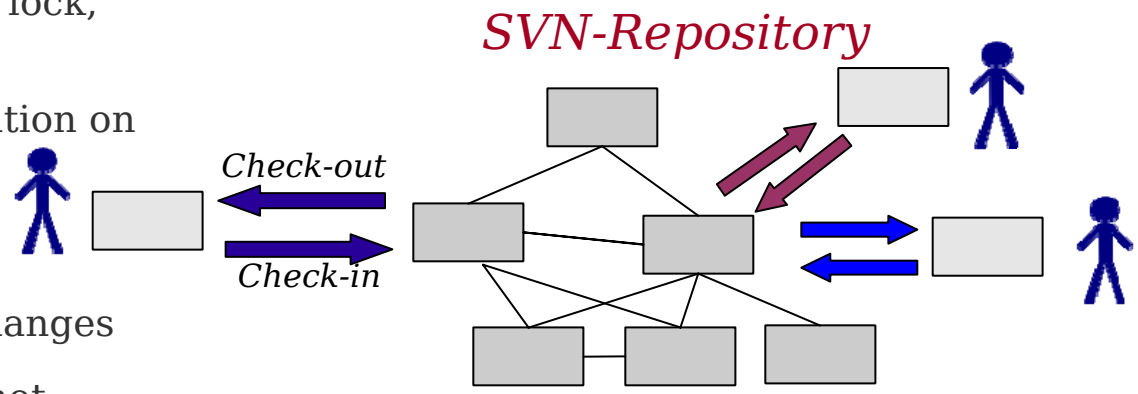
- Realize own idea
- Locally centralized
- Communication not an issue
- You know where to look and find

- Realize community consensus
- Locally distributed
- Collaboration & Communication editing, discussion & annotations
- ‘Issue archeology‘ becomes an issue

SVN vs. Concurrent Editing in CP

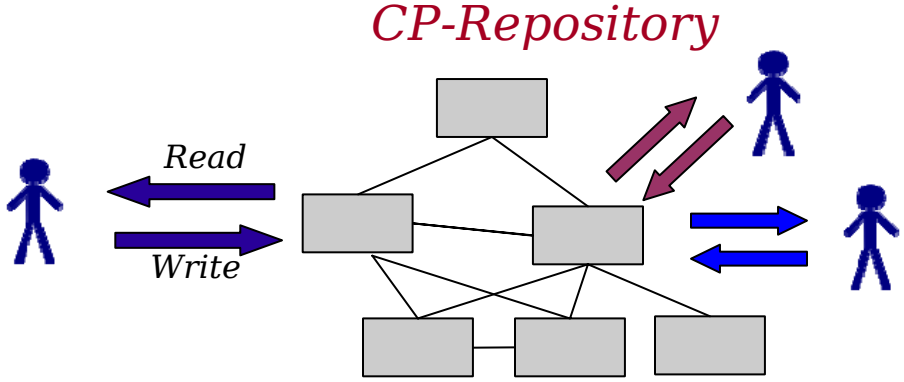
SVN

- Successive access (update, lock, modify, commit local copy)
- Complicated conflict resolution on whole RA, even with logically non-conflicting changes
- High threshold for small changes
- Change and diff functions not feasible for owl
- Annotations separate from actual RU



CP

- Simultaneous access
- Simple editing
- Annotations associated to RU



CP Features

Editing

- Concurrent distributed Ontology Editing

Metadata

- Annotations on RUs (editorial and administrative metadata)
- Annotations on Changes (annotations linked to delete actions and axiom edits)

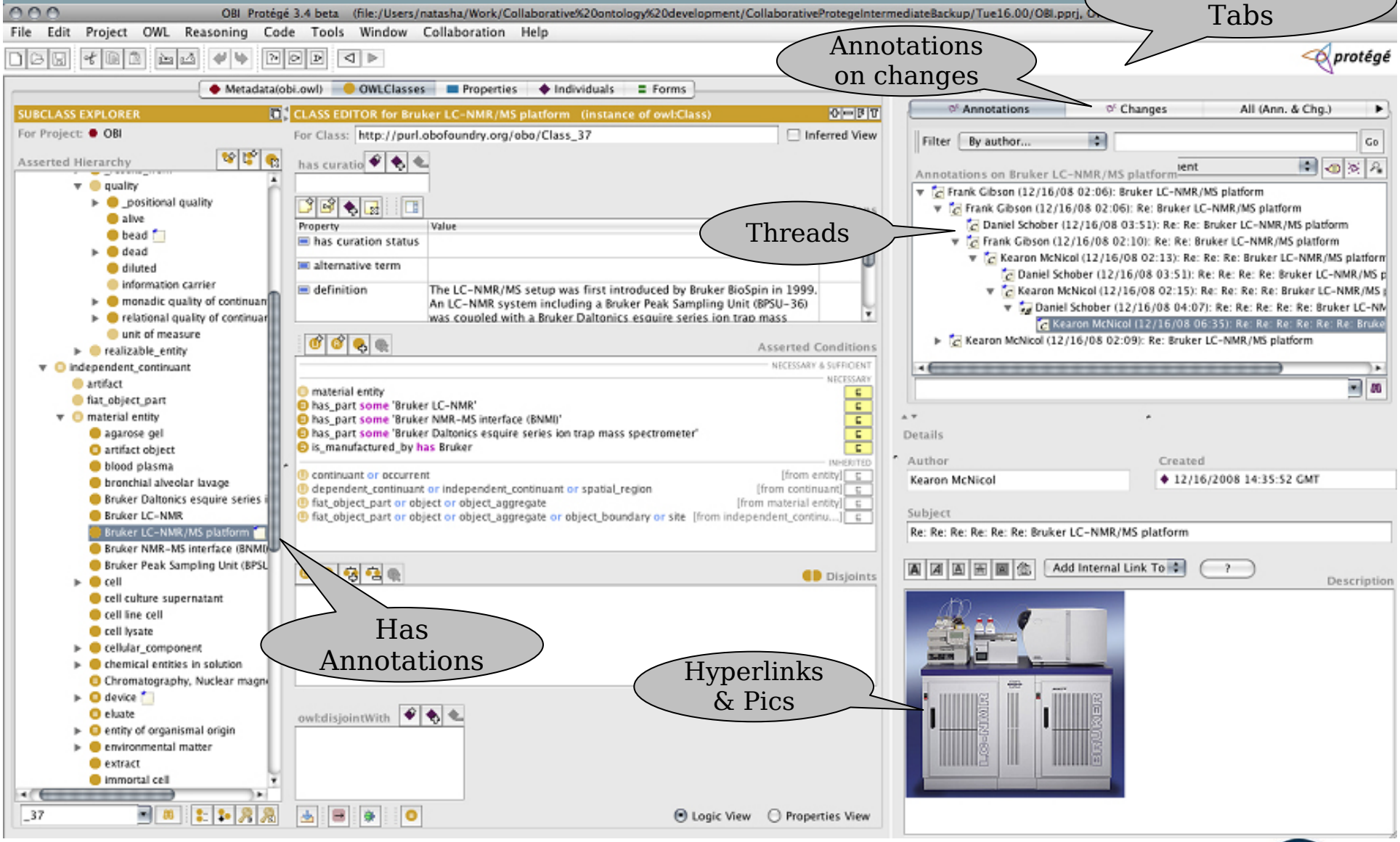
Searching

- Search via user, annotation type & datestamp

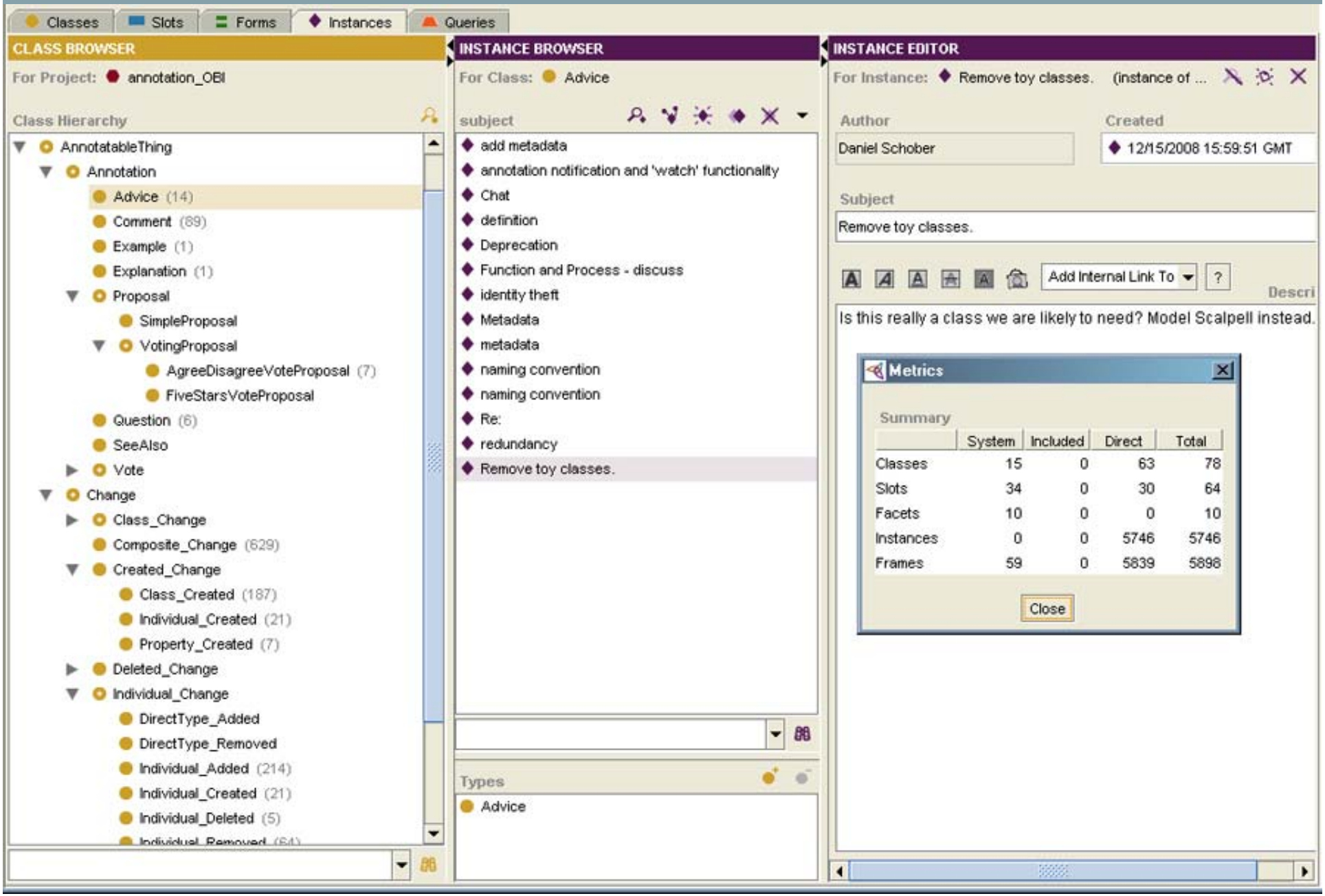
Communication

- Discussion threads
- Chat function (instant messaging)
- Voting for decision support

Changes Tab and Change Annotation



Changes & Annotation Ontology (ChAO)



CP Tool Evaluation Method

- OntoGenesis network meeting at EBI (n=13, 2 days)
- Enrich OBI (OWL-DL)
- 'Devices/Instruments' branch
  - All members could contribute
  - Devices from
    - User domains
    - List provided by the Metabolomics Standard Initiative
- Feedback to CP developers

CP Tool Evaluation Method

Ad hoc additions under OBI (device and functions)

Duplication possible

How are conflicts resoved ?

Controlled additions

Placement of devices from provided term list

How is agreement (on subsets) coordinated ?

'Agent Provocateur‘

Secretly adding conflicting and incorrect content

How transparent are faults and nonsense edits to others ?

Controlled Communication

Restricted to specified channels during each editing session

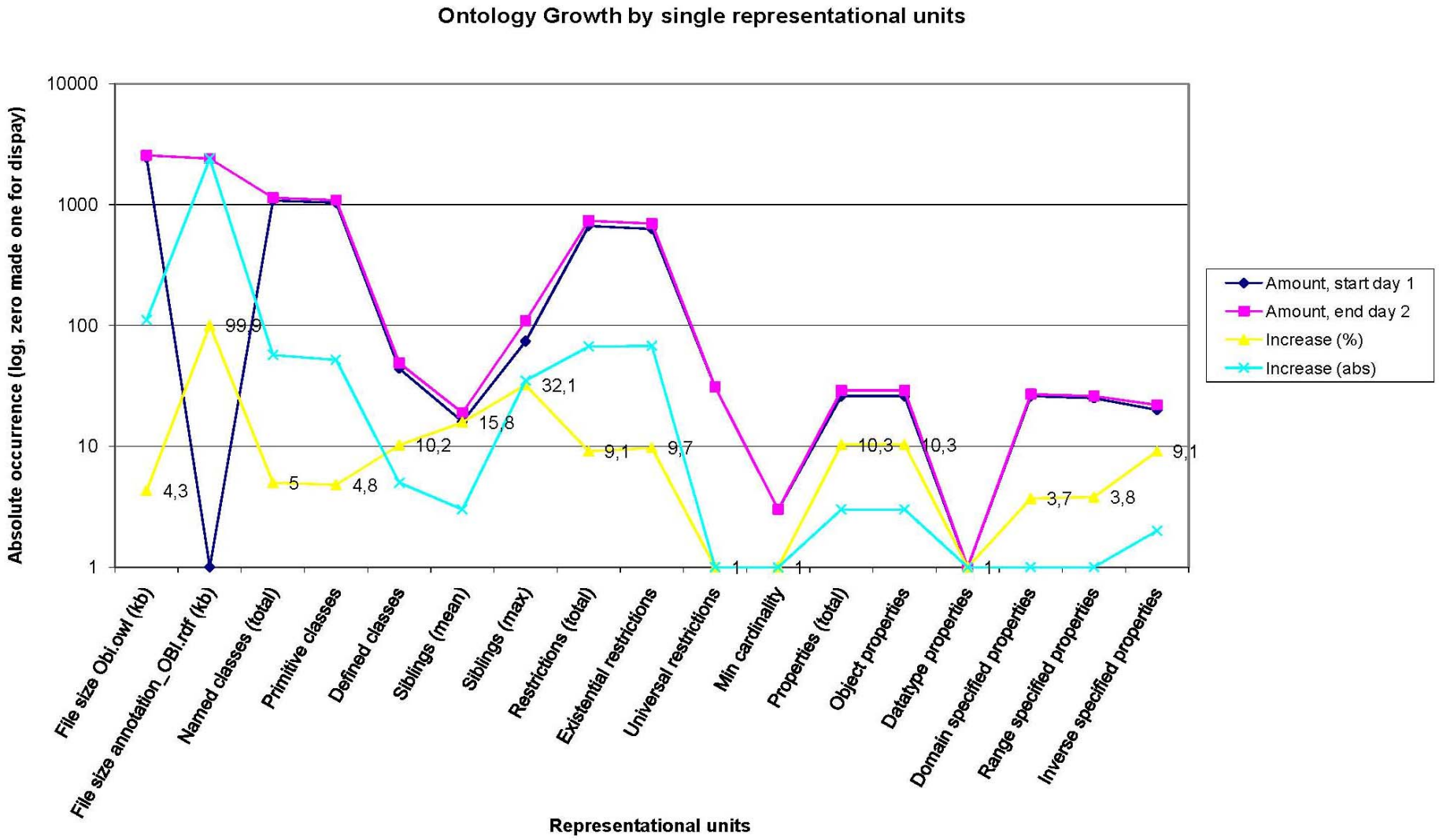
Verbal shout-out, notes, discussion threads and chat

How does CPs foster problem solving in communication ?

CP Tool Evaluation Method

- Single group
  - Familiarization with CP & GUI
- Two groups
  - *Ad hoc* additions of own instruments
- Four groups
  - Add subsets of provided term list
  - Discuss, comments by other groups adding annotations
- Single group
  - Add more terms from list
  - Test communication channels
    - chat only (for comments, annotations and discussions of additions)
    - voice only
    - chat and voice together
      - Deploy *Agent Provocateur*
- *Reasoning done every half hour or so*

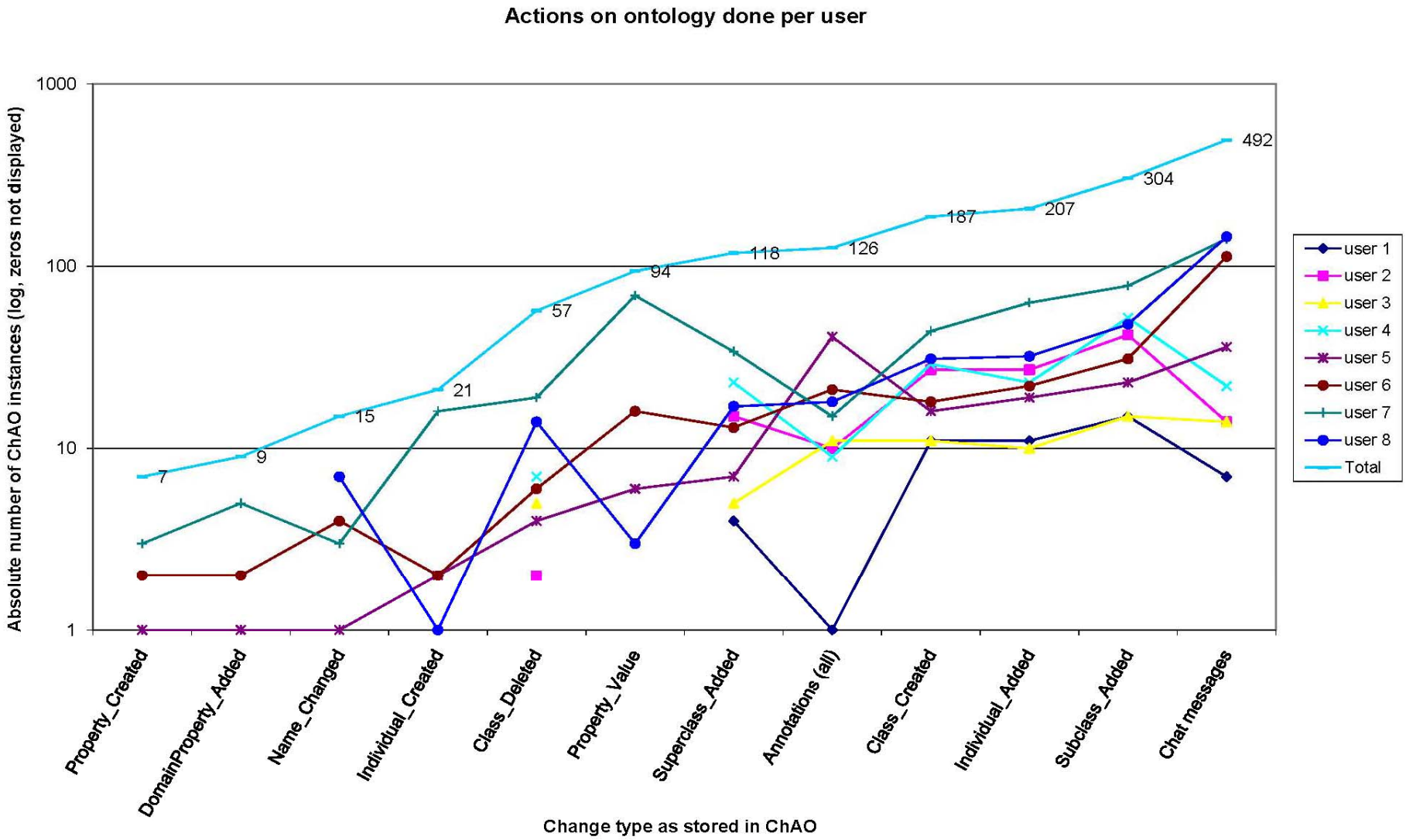
Results: Increase of ontology size



Results: Increase of ontology size

- Quick setup, installation guide was clear
- Metrix
  - 4.3% increase in OBI file size
    - 40 classes added, 13 refined/defined
  - 10.2% increase in defined classes, 4.8% in primitive classes
    - In OBI dev group primitive classes increase faster than defined classes
    - DL experienced Ontogenesis members
  - Only 3 object properties were created
    - 10.3% increase
    - Mainly re-use from OBI and RO
    - Relations used in 68 new existential restrictions (9.7% increase)
  - 46,1 % increase in annotation_OBI.rdf (per day)
    - 77 annotations (20 class annotations)
    - linear growth, no performance problems here

Results: Changes done per user



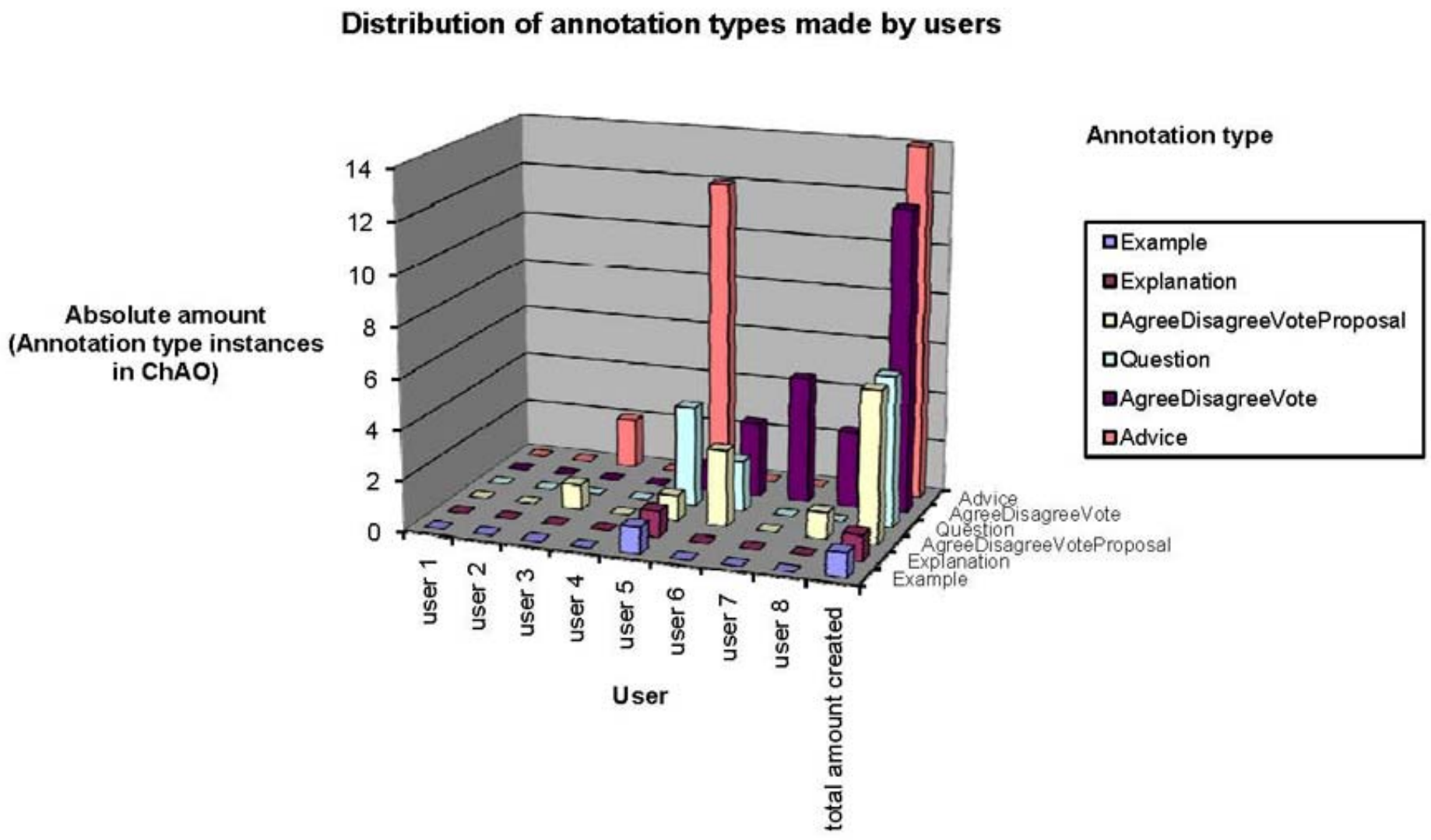
Results

- Large differences in overall activity
  - result of personality-structure, experience and confidence level
  - Quality of changes not yet evaluated
- Chat activity ~ overall editing activity
- Development of interest domains
  - E.g. user 7 worked on relations, user 5 on annotations
- Development of ‘user roles’
  - Users making comments don‘t nesessarily implement them
  - Some users created tasks for others
    - e.g. 'add metadata', 'remove redundancy'
  - ChAO Patterns can be used to infer user roles
    - e.g. 'moderator, 'commenter', 'chatter', 'changer'
- Most classes edited by several editors (avrg. 2 per cls)
  - Changed classes: 13, (removed and added restrictions, changed superclasses, changed from primitive to defined, added annotations)

Results

- No power law distribution for comments per person
  - Most made ca 10 comments, only ‘moderator‘ made 20
  - Role motivations could be Competition, Altruism, Narcissism, …
- Discussion thread mean depth was 2,5, max depth was 5 responses
- Chat Issues
  - What to work on next, modeling issues, new features & implementation
- Only 12 chat-lines used internal hyperlinks (increasing over time & CP familiarity)
- Experimental helperclasses
  - '_Kearon's collect devices by function classes', 'Frank's new meaning of function', 'asserted_gibbon_disco,
  - Only one user adhered to the OBI policy to indicate such play-classes with the underscore prefix (see first expl.)

Usage of ChAO Annotation Types



Usage of ChAO Annotation Types

- Comment used due to 'default' setting
  - For 2 users comment was the only annotation
  - Comment per class distribution followed power law
    - Few classes had 10-17 comments
    - Most classes had only 1-4 comments
- Advice and AgreeDisagreeVotes were used second abundandly
- There were a few AgreeDisagreeVoteProposals and Questions
- Example and Explanation were used most seldomly
  - Distribution of annotations over the annotation types was highest among experienced users
- No annotations on changes
- No SimpleProposal, FiveStarProposal, FiveStarVote and seeAlso used

Overall Performance

- GUI updating
  - Expanding full class hierarchy in larger artefacts (took ca. 20 sec first time)
  - Opening a class with many direct subclasses will slow down clients and impair performance when done the first time
- Performance increased by larger Heap Size & removing concurrent projects from metaproject KB
- Protégé project loaded in 3 Min (on a 512MB P4 PC)
  - 2 Min for project, 1 for GUI
- Using DTB backend would increase performance (dynamic loading) & risk of data loss minimized (rollback)

Desired Features

- RU and module locking mechanism
  - Can't prevent others from editing classes currently worked on
  - Parent class edits by unaware users can contradict definitions under construction
    - Highlight edited areas e.g. by user colour scheme
- Roll back function
  - Aid in conflict resolution
  - Undoing of deleted classes
  - Properties were found to be sub-properties of deprecated properties
    - Global change list to allow to see changes and annotations on deleted entities
- Subscription and Notification
  - Notification of changes would help to stay up to date and proceed faster in conflict resolution
  - E.g. a 'change view' on selected watch list items (see ICBO paper on how to implement)
  - Notification on duplicate RU labels

Desired Features

- Planning
  - A mechanism that changes the ontology based on vote outcomes would increase development time and could be implemented using ChAO information and formalized voting outcomes.
  - Issue tracker
    - A scratch pad or todo list that can be worked through and 'checked', e.g. indicating a proposed plan & what has been already realized at a certain time point
  - Connection with e.g. SF term trackers ?
- Chats
  - ‘Retreat room' was desired
  - Filter function on user names or particular ontology fragments
  - Emoticons could increase transmittance of pragmatic communication aspects

Further observations

- Annotation on RUs
  - For minor annotations providing annotation type, subject heading and value is overkill
  - Change track in ChAO KB is sometimes overly granular (overkill)
    - Users like high level abstractions, e.g Class X moved under Class C
- Communication
  - Threads and notes were misused for chats and *vice versa*
    - The latter due to the chats' instant visibility
  - Difficult to find cut off, when to move from chat to RU note or thread
  - Consequences of using wrong annotation channel
    - A user adviced the group not to use an obsolete object property in a tread rather than in a note on that object property itself
    - As a consequence people used the obsolete property

Overall CP benefits

- Changes immediately visible to all clients
  - Use during telecons directly rather than redundantly keeping notes and later inplement them
- Rich set of annotation properties
  - Advice, comment, explanation, question, example, ...
  - Change-annotations ease deprecation and versioning
- Dentralized access to otherwise distributed contextual metadata
  - *Issue-archaeology* much easier
- Flexibility of ChAO metadata scheme
  - Annotation types can be expanded, searched and filtered
  - Granular annotation types to suit own needs and evaluation approaches
  - Exploit for statistics
  - Use for proof and trust
  - Use for all non-DL add-ons, e.g. epistemiology
  - Use for mapping and alignment implementations
- Personalized views based on
  - User roles and tasks
  - User level of expertise
  - User trust network

Conclusions

- Rich CHAO metadata set provides audit trail of edits and decision making
- Tool in advanced stage with good performance
- Can be used in practice with sufficient stability
- Copes with complicated setups
  - Flexible enough to allow for corresponding adjustments
- Desired features
  - More sophisticated communication mechanisms are desired
  - Conflict resolution, e.g. 'undo/redo' is needed, as well as transaction management
  - Notifications on changes to notes and threads
  - Chats to specific RUs and for specific groups would enhance annotation traceability
- Feedback valuable for CP version of P4

Resources and Acknowledgements

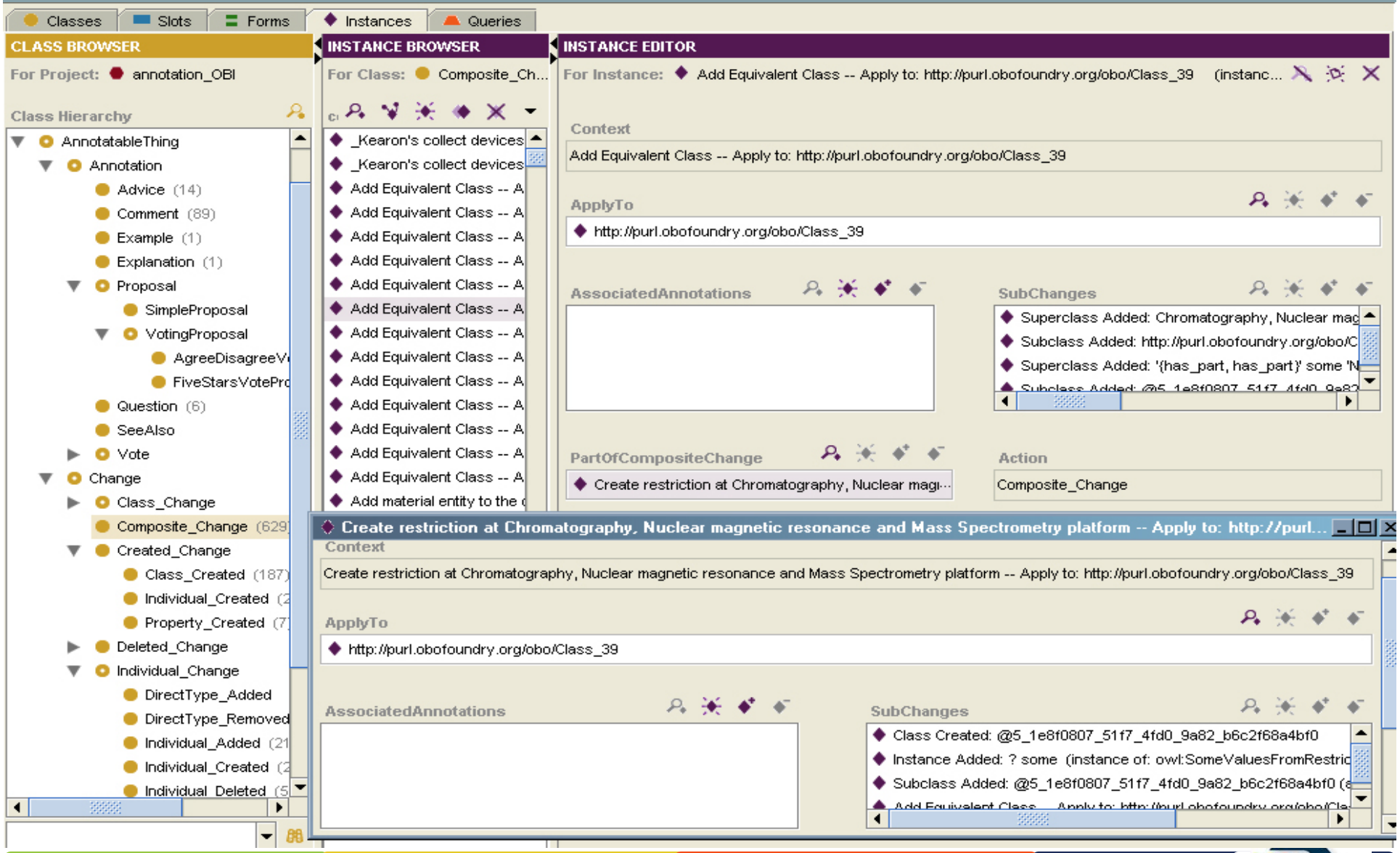
Resources

- Ontogenesis Website
  - http://ontogenesis.ontonet.org/moin/NetworkMeeting7
- CP Demo
  - http://protege.stanford.edu/doc/collab-protege
- Documentation
  - http://protege.stanford.edu/doc/collab-protege/doc/collabProtege_demo.pdf

Acknowledgements

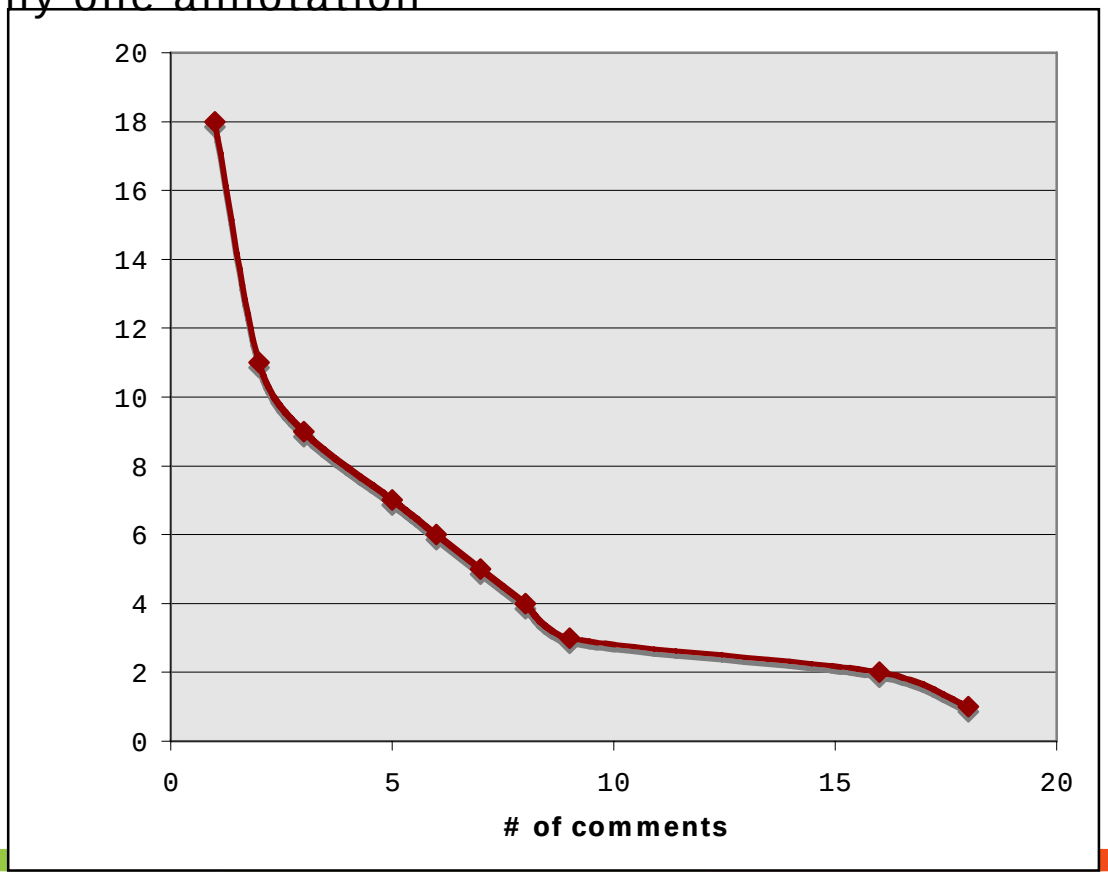
- Robert Stevens, James Malone, Susanna Sansone, Stefan Schulz
- Tania Tudorache, Timothy Redmond, Natasha Noy
- OBI Consortium
- DebugIT EU 7th FP ICT-2007.5.2-217139
- EBI NET-project, www.ebi.ac.uk/net-projects

Changes & Annotation Ontology (ChAO)



debugIT

- Power law distribution
  - a few classes with large number of annotations (> 15 each)
  - a large number of classes with only one annotation



- The ratio of created to deleted classes was 2,1 for user7, 2,2 for user 8, 2,3 for user 3, 3 for user 6, 4 for user 5, 4,1 for user 4 and 13,5 for user 2
  - Ratio smaller in users that generally made more changes (outlier user 4), than in more 'careful' users