Protégé Knowledgebase Coordinator

Noah Zimmerman
Herzenberg Laboratory
Department of Genetics
Stanford University

8th Intl. Protégé Conference
Madrid, Spain
July 20, 2005

Outline

1. Why build multi-ontology applications?
2. FacsXpert project background
3. Knowledgebase coordinator features
    1. Modeled knowledge structure
    2. Access controlled inclusion hierarchy
    3. Distributed resource coordination
4. Future work

Outline

1. Why build multi-ontology applications?
2. FacsXpert project background
3. Knowledgebase coordinator features
   1. Modeled knowledge structure
   2. Access controlled inclusion hierarchy
   3. Distributed resource coordination
4. Future work

Benefits of multi-ontology applications

- Utilize different ontologies for varying levels of granularity
  - upper level ontologies (SUMO, OpenCyc)
  - mid-level ontologies (MILO)
  - lower-level ontologies (MGED, FMA)
- Reuse existing ontologies
  - limit duplication of knowledge
  - save on development time
  - promote the development and reuse of validated ontologies

Benefits of multi-ontology applications (cont'd)

- Build systems that leverage knowledge from multiple domains
  - utilize the knowledge of domain experts (for free!)
  - create cross-disciplinary applications

Issues and difficulties

- Syntactic integration
  - How do I merge multiple ontologies?
  - How do I compare different versions of ontologies?
  - How can I promote instances/classes from included to including ontology/knowledge-base?
- Approach → Prompt

Issues and difficulties (cont'd)

- Technical integration
  - How do I get my ontology represented in format *X* to integrate with my other ontology represented in format *Y*
- Approach → standards (OWL) and translation services

Issues and difficulties (cont'd)

- Semantic integration
  - Is it sensible for one type of ontology to include another type?
- Approach → Knowledgebase coordinator

Issues and difficulties (cont'd)

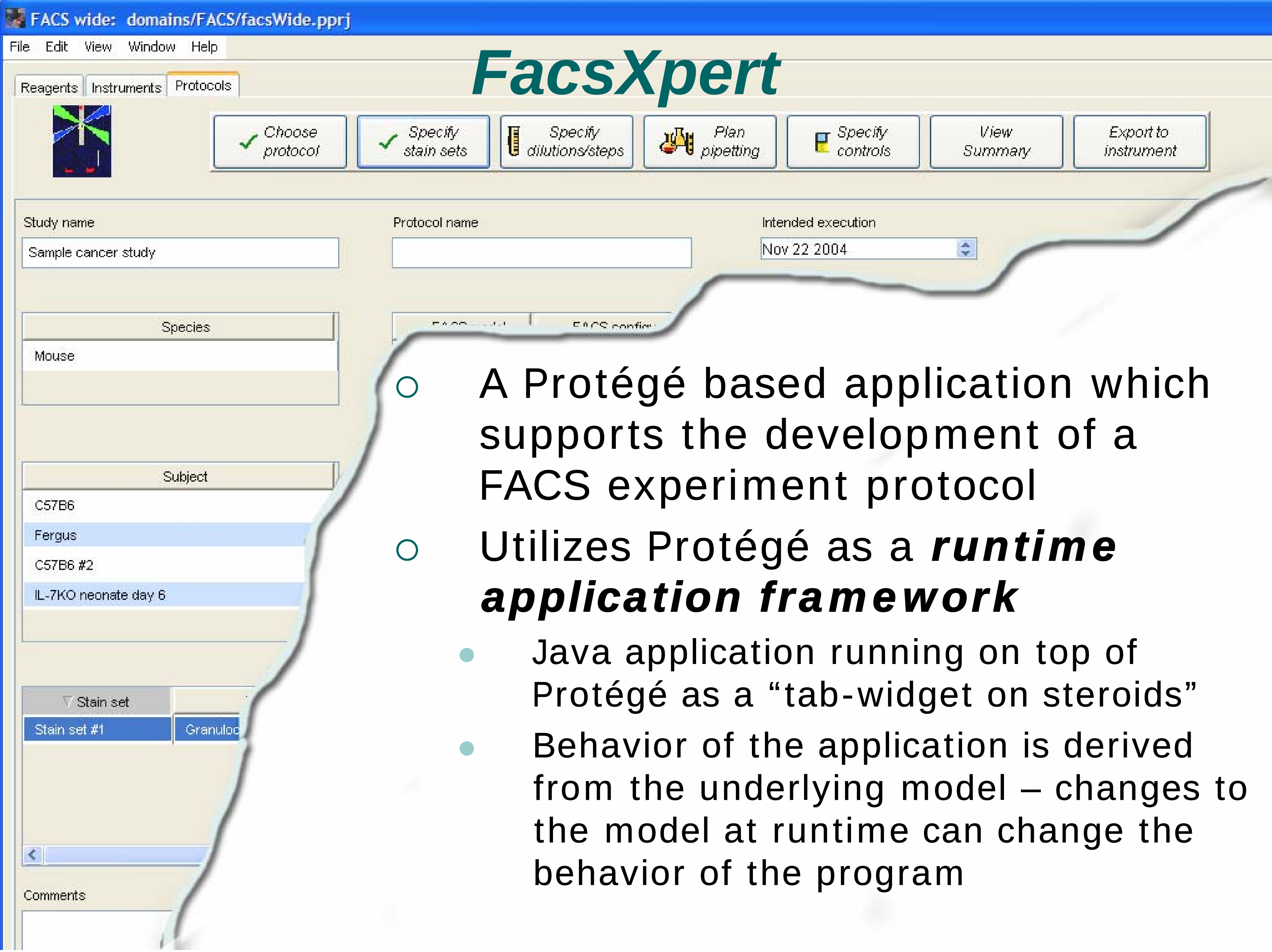
- Secure integration
  - Who can access which ontologies?
  - Who can modify which ontologies?
  - Which version of the ontology am I working with? Are other people trying to use/modifying this ontology?
- Approach → Knowledgebase coordinator

Issues and difficulties (cont'd)

- Distributed resources integration
  - How can we get multiple ontologies existing in different locations seamlessly into a single project?
  - How can we manage access to these ontologies online AND offline?
- Approach → Knowledgebase coordinator

Outline

1. Why build multi-ontology applications?
2. FacsXpert project background
3. Knowledgebase coordinator features
    1. Modeled knowledge structure
    2. Access controlled inclusion hierarchy
    3. Distributed resource coordination
4. Future work

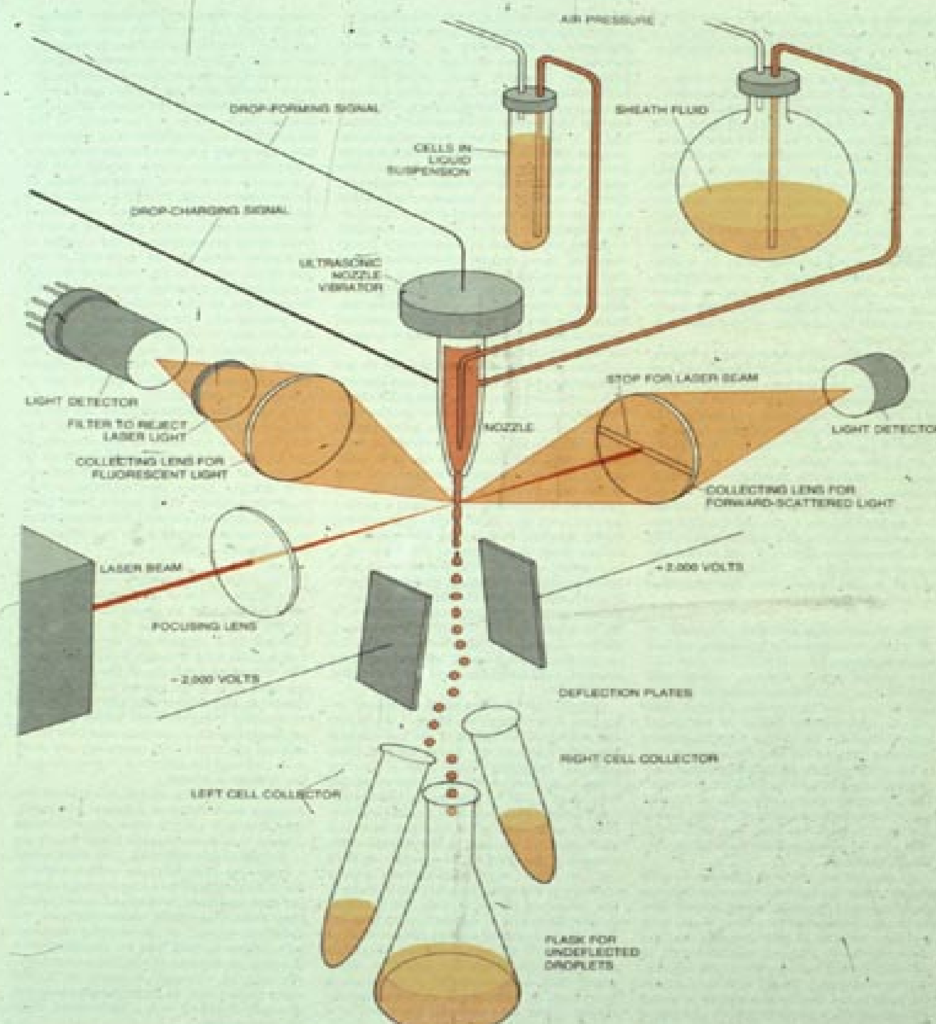


FacsXpert

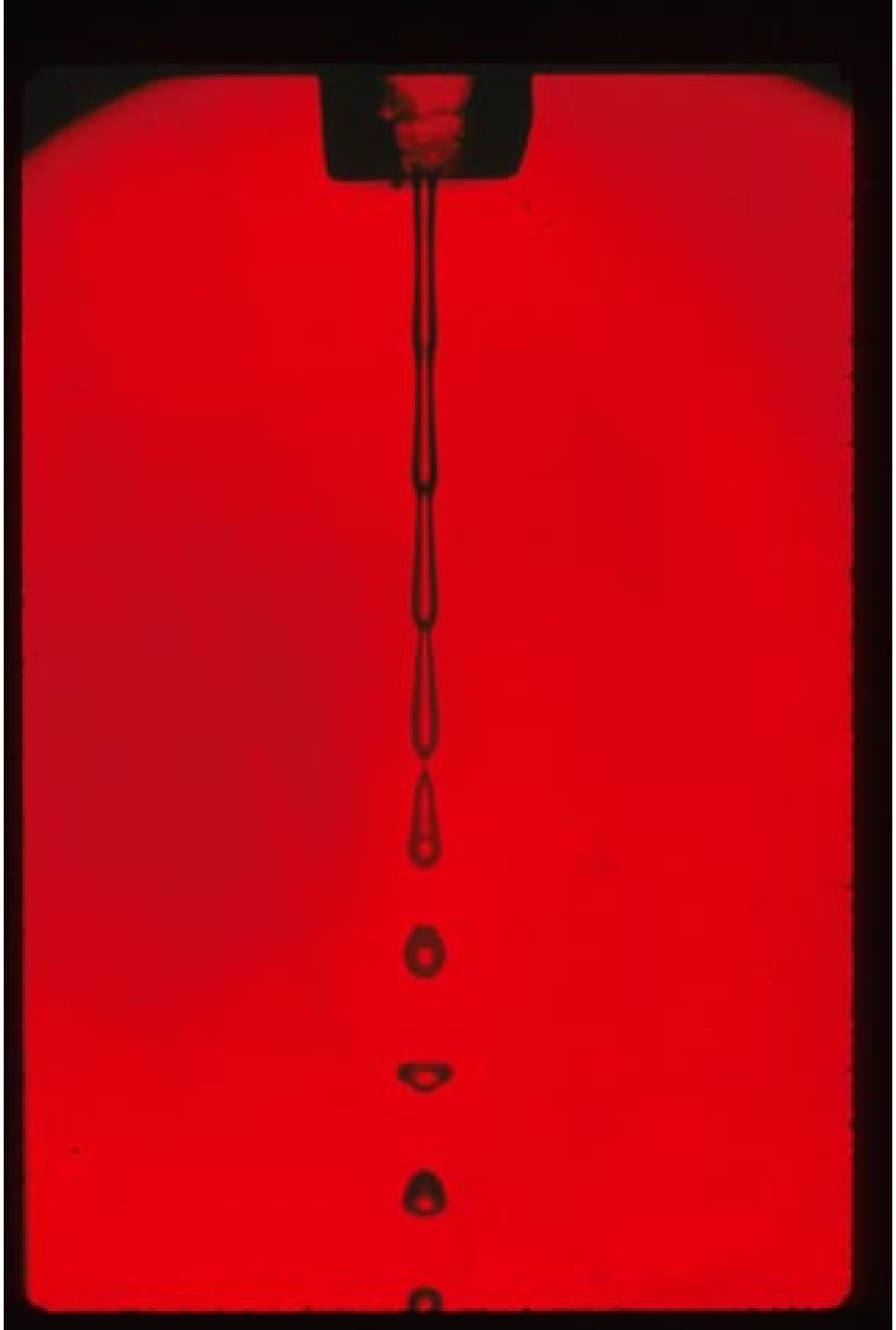
- A Protégé based application which supports the development of a FACS experiment protocol
- Utilizes Protégé as a ***runtime application framework***
  - Java application running on top of Protégé as a “tab-widget on steroids”
  - Behavior of the application is derived from the underlying model – changes to the model at runtime can change the behavior of the program

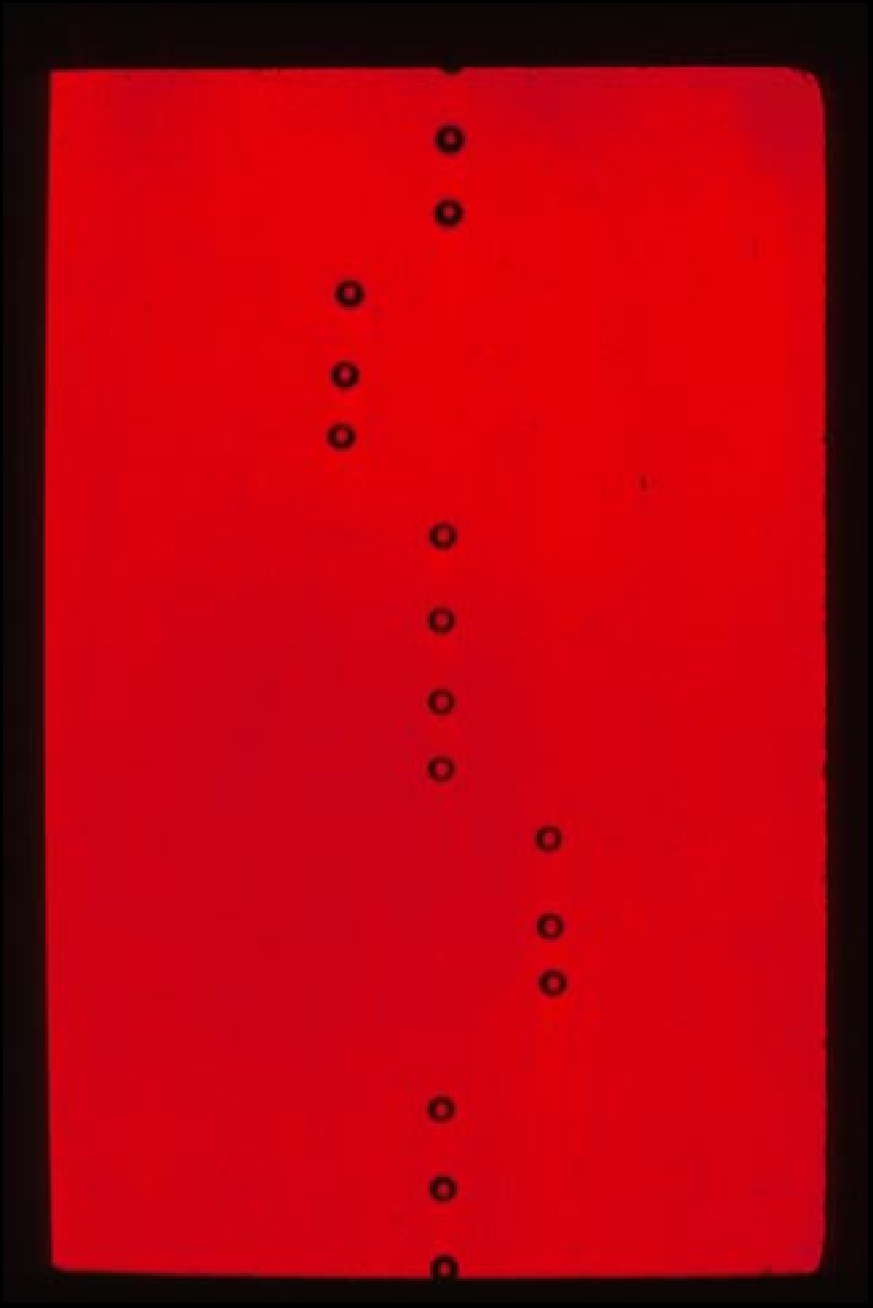
What is FACS?

- Fluorescent Activated Cell Sorter
- Purpose
  - Allows us to examine large numbers of cells one at a time so that we can analyze/sort specific subsets of cells (blood cells, e.g. T-cells, B-cells; tumor cells, stem cells, etc.)



MAIN COMPONENTS of the fluorescence-activated cell-sorting machine are depicted in this schematic diagram. The cells are kept near the axis of the effluent jet by forcing them under pressure through a nozzle into the center of a stream of cell-free sheath fluid. The nozzle assembly is vibrated axially at 40,000 cycles per second, breaking the jet into 40,000 droplets per second. The detected fluorescent light generates an electrical signal that is proportional to the number of fluorescent molecules in each cell. The signal that is generated by detecting the forward-scattered light is related to the volume of the passing cell. The drop-charging signal is triggered by means of an electrical system that processes the signals that are received from two light detectors (*see illustration on opposite page*).





FACS experiment design issues

- The fluorescent tags used in an assay depend on the specific FACS instrument and configuration being used as well as the available inventory
- When different fluorescent tags are used, signal spillover can occur between fluorescence channels.
- And many, many more . . . . . .

FacsXpert approach

- Leverage an expert knowledge-base to assist in design time decision support
  - What machines do I have available?
  - How are the machines configured?
  - What color fluorescent tags does this machine recognize?
  - Which fluorescent tags can I use together?
  - What controls do I need to do in order to properly calculate for compensation?
  - What reagents do I have in my inventory?

What is the knowledgebase coordinator?

- A FacsXpert plug-in to Protégé's inclusion mechanism, extending it so that it
  - is based on modeled knowledge
  - is access controlled
  - can support distributed resources

Outline

1. Why build multi-ontology applications?
2. FacsXpert project background
3. Knowledgebase coordinator features
    1. Modeled knowledge structure
    2. Access controlled inclusion hierarchy
    3. Distributed resource coordination
4. Future work

Modeled knowledge structure

- Project inclusion is constrained by a formal model to
  - Specify classes of ontologies which are “legal includees”
  - Specify classes of ontologies which are “legal includers”
  - Specify cardinality of includers and includees

Jambalaya demo

Four-tiered hierarchical knowledge structure

- Domain wide
  - Concrete knowledge shared by all FACS researchers
    - fluorescent emission spectra
    - lasers
    - instruments
- Organization wide
  - Knowledge shared by a facility of FACS researchers
    - Instrument configurations
    - Instrument-specific experiment templates
- Group wide
  - Knowledge shared by a lab of FACS researchers
    - Shared reagents
    - Shared protocols
- Individual
  - Knowledge specific to an individual researcher
    - Unshared reagents
    - Unshared protocols
    - Personalize instrument templates

Example

- A FACS machine, the Aria, is defined at the domain-wide level
  - 3 lasers
    - Argon
    - UV
    - Solid State
  - Recognizes 12 fluorescent colors with *N* fluorescent detectors

Example

- A single Aria can have multiple configurations
- The specific configuration is defined at the organization-wide level
  - The order of the lasers
    - UV
    - Solid State
    - Argon
  - The specific fluorescent detectors that the facility's Aria is using

Benefits

- Allows the system to reuse common knowledge at varying levels of granularity
- Ontologies are decoupled along the hierarchy and can be restructured according to different needs
- The *constrained* Protégé inclusion allows us to present a complex knowledge structure transparently to the user through a single Protégé-based interface

Outline

1. Why build multi-ontology applications?
2. FacsXpert project background
3. Knowledgebase coordinator features
    1. Modeled knowledge structure
    2. Access controlled inclusion hierarchy
    3. Distributed resource coordination
4. Future work

Controlled inclusion hierarchy

- Control access to each individual ontology within the inclusion hierarchy
  - Allows a user to simultaneously view four different ontologies, while only allowing them to edit two of them
  - Allows users to edit instances of both included and including projects

Controlled inclusion hierarchy (cont'd)

- The inclusion ontology provides "traceability" through the inclusion hierarchy
  - Track which ontology this particular instance was included from
  - Track which version of this ontology this instance came from

Outline

1. Why build multi-ontology applications?
2. FacsXpert project background
3. Knowledgebase coordinator features
   1. Modeled knowledge structure
   2. Access controlled inclusion hierarchy
   3. Distributed resource coordination
4. Future work

Distributed Resource Coordination

- Allow the user to operate both online AND offline
- Hide the details of:
  - Where the data sources for the ontologies are located
  - How to connect to them
  - How to make them seamlessly available to the user online and offline

Distributed Resource Integration - details

- Application is distributed in application and resource *jars* using Java Web Start
- Knowledgebase coordinator caches local copies of the knowledgebases
- At startup
  - If connected to internet
    - Compare local versions against server versions
    - Include the more recent of the two
    - Put local version to server if necessary
  - Else
    - Refer to copies cached locally
- User has the most recent copy of the ontology
- Only permits a single user to modify a single ontology at the same time

Summary

- The knowledgebase coordinator extends the standard Protégé inclusion mechanism to provide
  - An ontology for modeling the semantics of a project inclusion hierarchy
  - Ontology specific access control
  - Transparent online/offline ontology access

Outline

1. Why build multi-ontology applications?
2. FacsXpert project background
3. Knowledgebase coordinator features
    1. Modeled knowledge structure
    2. Access controlled inclusion hierarchy
    3. Distributed resource coordination
4. Future work

Future Work

- Incorporate Prompt for the promotion of classes/instances in the hierarchy
- Increase support for simultaneous offline modifications
- Package and document for distribution to Protégé community

Acknowledgements

- Stephen Meehan
- Herzenberg Laboratory
  - Len and Lee Herzenberg
  - James Tung
- Ethan Stone
- Protégé team