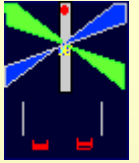


FacsXpert

July 8, 2004

7th International Protégé Conference

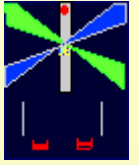


- *Builds protocols for studies with FACS instruments*
- *Uses a modified Protégé-based architecture that promotes runtime extensibility for the end-user*



FacsXpert

- *Built*
 - At Stanford University's Herzenberg Lab
 - In close collaboration with Lee Herzenberg, James Tung, David Parks, Wayne Moore and other researchers in the Herzenberg Laboratory
- *With help from*
 - Mark Musen, Ray Fergerson, Monica Crubezy, Natasha Noy and the Protégé community



FacsXpert

- *Presented by*
 - Stephen Meehan, software engineer
Herzenberg Lab, Stanford University
meehan@darwin.stanford.edu
 - Since 1992, Stephen has pursued a “dream” of a “software democracy” architecture where:
 - An application derives the majority of its behavior from runtime interpretation of its UML design model
 - A user can extend this model while the application runs

Thus the model is more than a design artifact, it is runtime government by the people ... for the people



Scope of Presentation

- 1. The FACS domain and need for runtime extensibility*
- 2. FacsXpert's approach to data structure extensibility*
- 3. FacsXpert's approach to data integrity extensibility*
- 4. FacsXpert's approach to GUI extensibility*

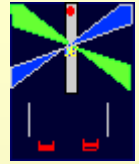


FACS instruments

Count and characterize fluorescent-labeled cells in suspension to

- Monitor the progress of HIV infections by counting the number of CD4 T lymphocytes in blood from HIV-infected people**
- Classify and stage tumors and to monitor bone marrow transplant survival**
- Do the basic science and studies that underlie all these clinical advances**

FACS protocols specify what will be done in a FACS assay

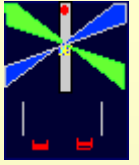


For each tube in the assay

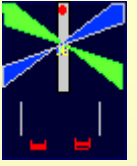
- Add up to 20 reagents, each linked to one of 12 fluorescent dyes
- Add one cell sample of up to 5 million cells
- Incubate and analyze with a multi-laser (Hi-D) FACS instrument
- Store the resulting data file (often about 5 megabytes)

One assay often has 60 or more tubes

Sophisticated software is needed



- Must compute a feasible combination of staining reagents (which “label” cells) by taking into consideration:
 - *Targeted species*
 - *Targeted cellular markers*
 - *Highly variable optical characteristics of fluorescent elements*
 - *Optical detection capabilities of a FACS instrument’s configuration*
 - *Fluorescence interference between fluorescent elements*
 - *Purpose of cell labeling: to gate, dump or discover cell populations*
 - *Availability of inventory*
 - *Bio-chemical affinities between staining steps*



Sophisticated software is needed

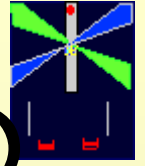
- Pipetting guide's
 - *Reagent and specimen amounts must follow highly variable dilution recommendations*
 - *Organization must cater to pipetting technician work flow which differs from the planning work flow of the scientist*
- Final protocol knowledge must interoperate with
 - *Instrument software (offline as well as real time)*
 - *Analysis software*
- GUI for decision making must be highly intuitive

Senior FACS scientists are often somewhere between computer naïve and totally computer-phobic

Highly extensible software is needed



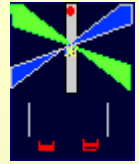
- **FACS vocabulary and “best practices” are**
 - Minimally standardized
 - Rapidly changing
- **From day to day, FACS scientists cannot predict**
 - What questions they will be asking
 - What materials they will be working with
- **Hence, their “research planning” software applications must be highly flexible and customizable**



Highly extensible software (cont.)

- **THUS ... scientists (in addition to knowledge/software engineers) must be able to extend**
 - Data structures (classes, attributes, etc)
 - Data integrity (rules that check knowledge inputs)
 - The GUI

... without crashing the current application or future upgrades!!!
- *Protégé provides the foundation for such a runtime evolvable system*
- *Commercial alternatives (e.g. Oracle, Rational Rose) suffer from higher financial cost, higher sys admin cost, closed source and design-time centricity*

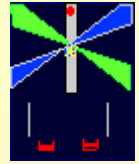


Data structure extensibility

- Protégé supports data structure extensibility through object-oriented subclassing
- However, the its model editor GUI overwhelms the scientist by exposing the whole model

FacsXpert required a model editor that only focuses on a specific part of the model *at the specific time* that it is relevant to the application

FacsXpert solution for data structure extensibility



Study name: B cells Protocol name: Avidin sample

Species
Mouse

Subject: C57B6 #2

Context menu for Mouse:

- Edit (Ctrl+E)
- New target species (Ctrl+N)
- Choose target species from list (Ctrl+A)
- Explore (Ctrl+X)
- Manage columns
- Manage rows
- Group
- Manage view
- Examine tool tip (Ctrl+F1)

A class-type slot widget that supports a create action which limits the model editor's focus to the slot's allowed parents

Pop up focuses on species class hierarchy starting at Mouse

Modify Mouse types

Relationship: Mouse

- Cartoon
- Disney
- Looney tune

Name: Looney tune

Concrete

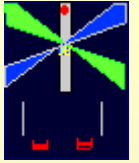
Template Slots

Inherited from	Name	Type	Cardinality	Other facet
Cartoon	Artist	String	single	
Cartoon	has own movie	Boolean	single	
Mouse	cage id			
Mouse	strain			
Species	key			
Species	subject group			subject group}
Species	subject TO facts			facts protocol proposal}
Trackable entity	:CREATION-TIME			
Trackable entity	:CREATOR			
Trackable entity	:MODIFICATION-			
Trackable entity	:MODIFIER			
Trackable entity	is active	Boolean	single	default={true}
Trackable entity	version	Integer	single	

Context menu for 'has own movie':

- Change entire column
- View selected attribute/slot
- View selected attributes/slots at class
- Create attribute/slot
- Add attribute/slot
- Remove attribute/slot overrides
- Remove attribute/slot
- Explore (Ctrl+X)
- Manage columns
- Manage rows
- Manage view

FacsXpert solution for data structure extensibility



- A small # of programming idioms ready *FacsXpert* for such extensions; for example:

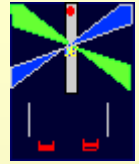
- *Creating instances*

```
myProject.createInstance( null,  
    DisplayUtilities.pickCls( null,  
        Collections.singletonList( designTimeLeafClass  
    )))
```

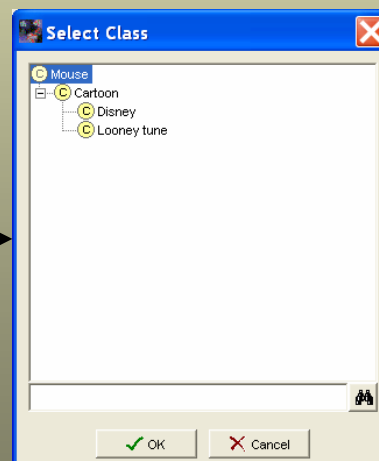
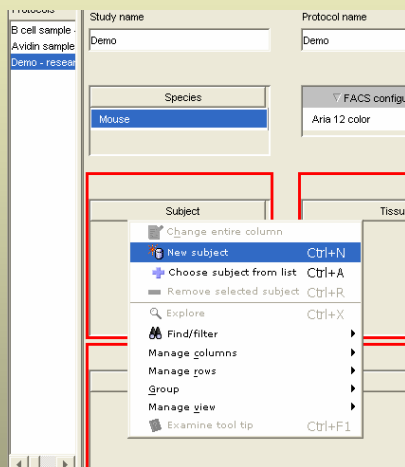
- *Querying class type*

```
designTimeLeafClass .equals( unknownClass ) ||  
    unknownClass.hasSuperclass( designTimeLeafClass )
```

Instantiating a new subject



The programming idiom detects the need to query for sub classes to the mouse class



Instance: researcherSpecificKB_00038

Key Tracking

Strain

Cage Id

Groups

name

Protocols

proto

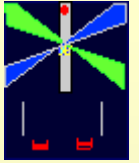
Artist

☐ Has Own Movie

Done

End user's extended attributes automatically appear

Darwin “crash-proofs” data structure extensibility



We built a tool named Darwin that guards model evolution by handling model editions which break hard-coded expectations

- For any given class, Darwin prevents the “**extinction**” of one or more slot associations and one or more slot facets
- One can set 1 of 2 “watch dogs” to monitor changes to “**endangered**” model elements:
 - *Golden retriever* barks when user changes such an element and allows rollback (for engineers)
 - *Doberman Pincher* prevents user from making the change (for end-users)



Darwin's JAVA generator

- Darwin generates JAVA modules that benefit client code by
 - *Establishing a simple and consistent idiom for type-safe access to the **endangered** portions of the Protégé model*
 - *Adding JAVA compiler checking as a means of tracking model dependencies*
- Darwin translates
 - All Protégé classes into JAVA interfaces hence supporting multiple inheritance
 - Concrete Protégé classes into JAVA classes that implement all associated JAVA interfaces

From Protégé class...



Classes Slots Forms FacsXpert 1.0 beta 2.0

Relationship Superclass

Facs single-reagent proposal (type=:STANDARD-CLASS)

Name: Facs single-reagent proposal

Documentation: This is a proposal for FACS channel that detects a single determinant.

Constraints: [V] [C] [+] [-]

Role: Concrete

Template Slots

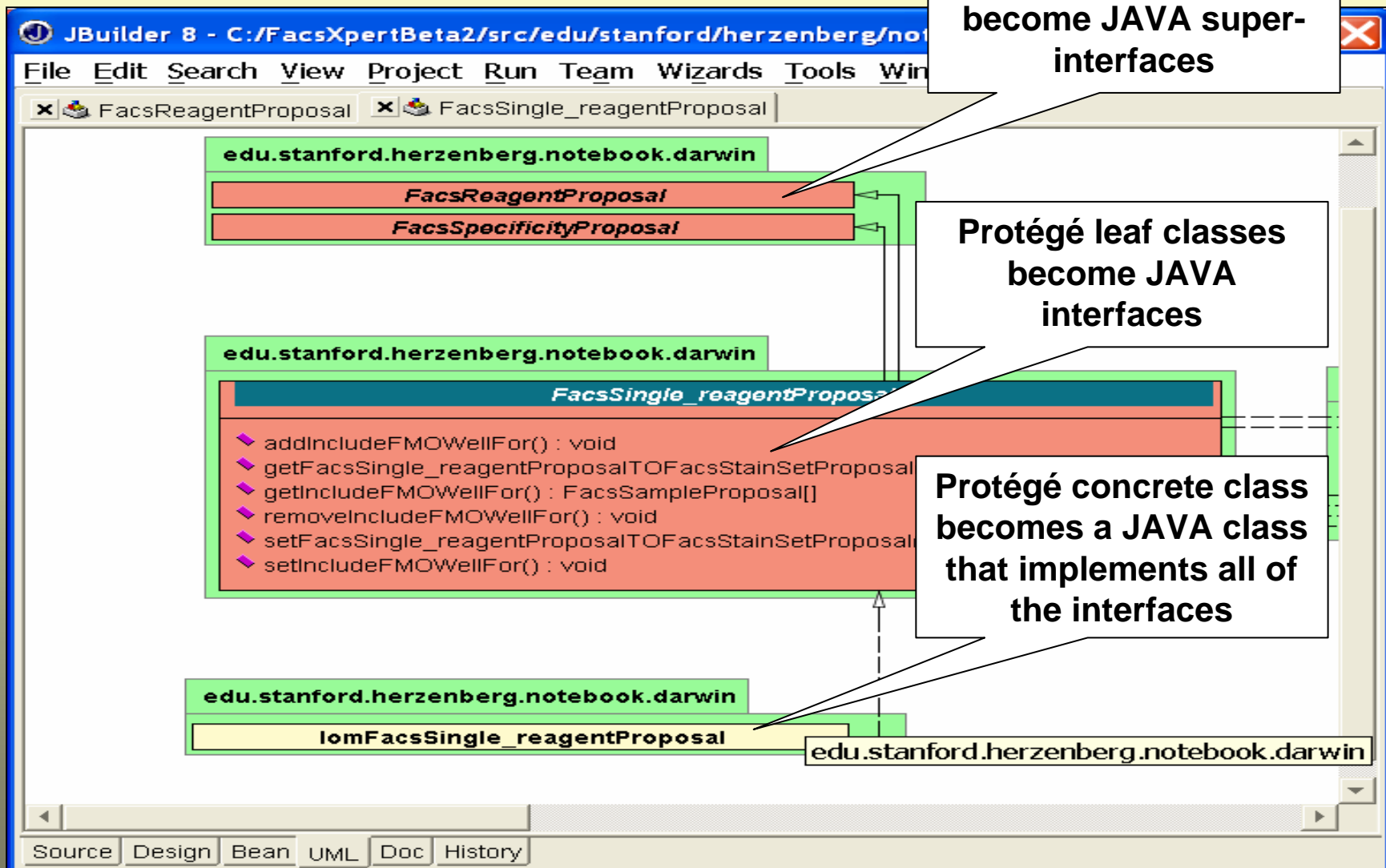
Inherited from	Name	Type	Cardinality	Other fac
Facs specificity ...	assigned handle	Instance	single	classes={Molecule}
Facs specificity ...	assigned optical detector	String	single	
Facs specificity ...	chosen fluorescent element	Instance	single	classes={Fluorescent element}
Facs specificity ...	chosen handle	Instance	single	classes={Molecule}
Facs specificity ...	determinant	Instance	required si...	classes={Determinant}
Facs specificity ...	facs lot and dilution proposal	Instance	multiple	classes={Facs lot and dilution prop...
Facs specificity ...	one step recommended titration	Symbol	single	allowed-values={high,low,medium,
Facs specificity ...	picked chemical reagent	Boolean	single	default={false}
Facs specificity ...			single	allowed-values={high,low,medium,
Facs specificity ...			single	allowed-values={high,low,medium,
Facs specificity ...			multiple	
Facs specificity ...			single	
Trackable entity	CREATOR	String	single	
Trackable entity	:MODIFICATION-TIMESTAMP	String	single	
Trackable entity	:MODIFIER	String	single	
Trackable entity	is active	Boolean	single	default={true}
Trackable entity	version	Integer	single	
	facs single-reagent proposal TO...	Instance	required si...	classes={Facs stain set proposal}
	include FMO well for	Instance	multiple	classes={Facs sample proposal}

The class "FACS single-reagent proposal" has two direct super classes

Superclasses:

- Facs reagent proposal
- Facs specificity proposal

...To JAVA





Data integrity extensibility

Primary goal

- Runtime extensibility of constraints that check knowledge inputs (hereafter checks). *Known to the Protégé community as “knowledge acquisition”*

Out of scope

- Runtime extensibility of constraints that
 - infer new knowledge
 - validate ontologies (consistency checking, disjointedness, semantic imbalance etc.)

In practical terms...

- Allow scientists to author and authorize checks while FacsXpert runs



Requirements

- A. Decouple check inquirers from check providers
- B. Support reuse for both of the above actors:
 - A. Inquirer: "Is anything wrong?"
 - B. Provider:
 - A. *Domain independent*
 - B. *Parameter-izable check or check template; user completes parameters when "binding"*
- C. Make checks model associated *and* oriented
- D. Integrate checks with Protégé forms & pick lists



Requirements

E. Make checks scalable

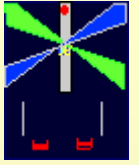
- A. Focus on limited instance context**
- B. Incremental execution**

F. Stay in sync with model re-factoring

G. Support stepwise debugging

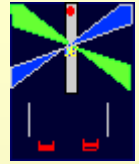
H. Represent constraints in model

- A. Start at :CONSTRAINT super class**
- B. Support inquirer querying of characterizations such as: advisory, warning, error and fatal**



Requirements

- I. Allow user to add checks while FacsXpert runs
- J. Support end user, ad hoc check authoring
- K. Allow computability of dependencies, conflicts & bottlenecks
 - A. Prevent conflicts between checks
 - B. Prevent conflicts between checks and model facets and onto-clean meta classes/slots
- L. Have reasoning agents that use computability
(otherwise prior requirement is "pie in the sky")



Data integrity extensibility

Engineering options (in 2002)

1. Reuse an existing runtime computable grammar
 - ✓ Pal with EzPAL
 - ✓ JESS
 - ✓ OCL
2. Invent a new “non programmatic” grammar; implementation options included:
 1. Translate invented grammar to accepted grammar to reuse interpreter
 2. Interpret the invented grammar directly
3. New JAVA validity checking framework that promotes “parameterized check templates”



We took choice 3

... and met 7 of 12 requirements

- ✓ **A. Decouple check inquirers from check providers**
- ✓ **B. Support reuse in both of the above actors**
- ✓ **C. Make checks model associated *and* oriented**
 - By sub classing Darwin
- ✓ **D. Integrate checks with Protégé forms & pick lists**
- ✓ **E. Make checks scaleable**
- ✓ **F. Stay in sync with model re-factoring:**
 - by sub classing Darwin class
- ✓ **G. Support step-wise debugging**

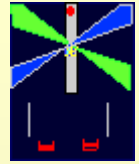


An example of validity checking

Optical detector class contains validity checks such as

- Number assigned for a FACS parameter must be
 - Between 1 and the number of colors for the parent FACS configuration
 - Unique amongst sibling optical detectors
- The laser configuration associated with a detector must also be associated with the FACS configuration parent

Checking integrity for parameter number value



Aria 12 color (type=Facs, name=reagents_16789)

Configuration name: Aria 12 color Max colors: 12 Tracking

Laser configurations

Wavelength	Laser
407	Krypton
488	Diode 488
633	HeNe

Aria/DiVa configuration template: Herzenberg settings.conf

Optical detectors

Laser configuration	Band pass filter	Parameter name	Parameter order
Diode 488	780/60	Cy7PE	12
Diode 488	710/40	Cy55PE	22
Diode 488	665/30	Cy5PE	2
Diode 488	610/20	TRPE	3
Diode 488	575/25	PE	4
Diode 488	525/50	FL	5
HeNe	780/60	Cy7APC	6
HeNe	710/40	Cy55APC	7
HeNe	660/40	APC	8
Krypton	540/80	Alexa430	9
Krypton	465/30	CasYel	10
Krypton	440/40	CasBlu	11

Protégé form for optical detectors signifies that the user has entered an incorrect parameter number

Cy55PE (type=Cy55PE, name=reagents_16789)

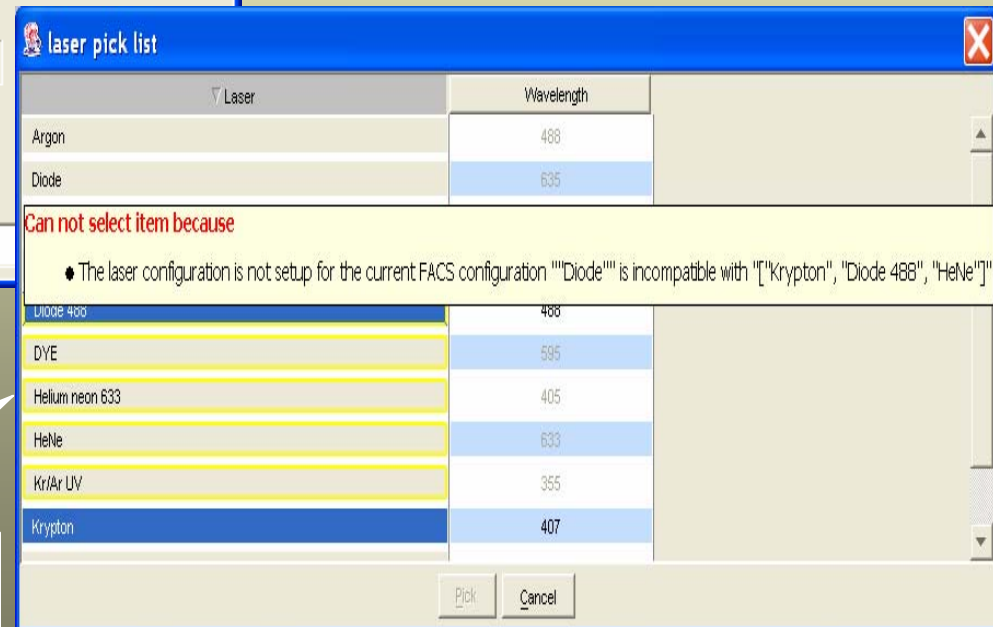
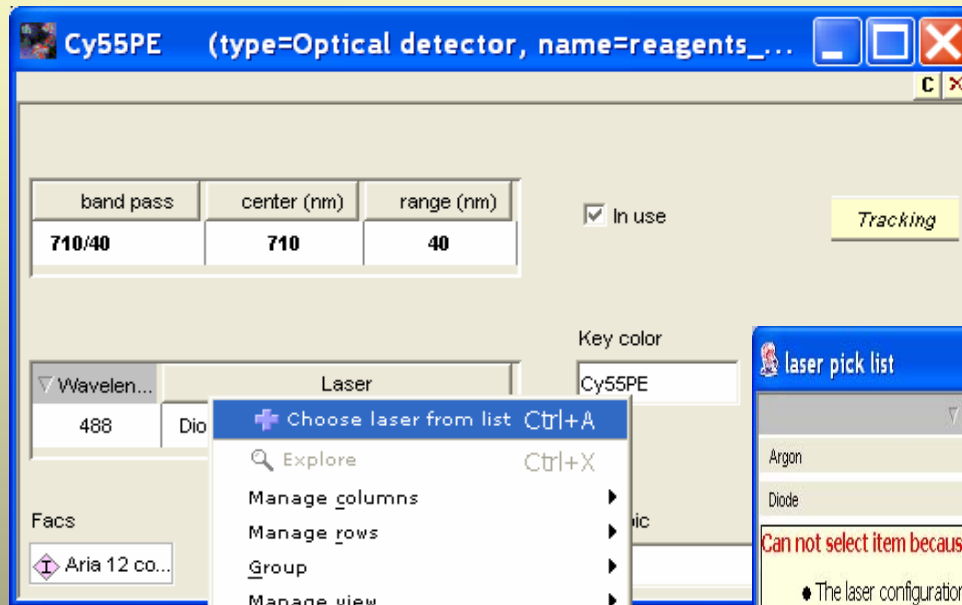
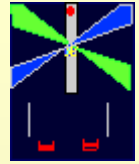
band pass: 710/40 center (nm): 710 In use Tracking

Key color: Cy55PE PMT Label:

Wavelength: 488 Diode 488

Parameter order must not exceed the max colors (12) for the FACS configuration

Checking integrity of Laser configuration relationship



Pick list displays invalid choices as disabled. An explanatory tool tip appears when the mouse is hovered over the disabled item.



What are our next steps?

Meet remaining requirements

- H. Represent constraints in model
- I. Allow user to add checks while FacsXpert runs
- J. Support end user, ad hoc check authoring
- K. Allow computability of dependencies, conflicts & bottlenecks
- L. Have reasoning agents that use computability, otherwise prior requirement is pie in the sky

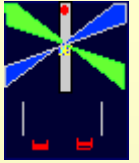
Current approach is limited:

- H and I are easily achievable, but not J, K and L
- At best it will become hidden plumbing for approach that hits all 12 requirements



And then there's the GUI

Protégé has GUI extensibility... but frankly

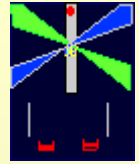


**FACS scientists and the Protégé GUI
were like oil and water**

- They were confused by the slot buttons with +, -, C, V, X ... and they balked at the dialogs
- They begged for a highly customizable Xcel-like widget (grid) for manipulating tables of data
- Then, after many iterations of developing this table widget, they ultimately decided that navigation and group-based modifications of items in the table had to be further simplified.

They wanted to access the table via a tree!

From a table (linear) view to...



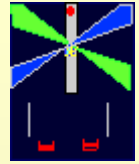
Pipetting plan

Done
Sample properties
Sample pipetting
Stain set pipetting
Comfort %
Plate setup
Full table
Help

Tissue	Stain set	Subject	Cell sample total ml	Cells/ml X 10 ⁶	Repeats	Cell sample ul to	Include	Fluorescence minus one (FMO)	Stain
									step volume (ul)
is not									
Lymph node									
Lung	B stain	C57B6 #2	2.5	25	1	25	<input checked="" type="checkbox"/>		
Lung	B stain	C57B6 #3	2.5	25	1	25	<input checked="" type="checkbox"/>		
Lung	B stain	CD22KO #4	2.5	25	1	25	<input checked="" type="checkbox"/>		1 25.0
Lung	B stain	CD22KO #5	2.5	25	1	25	<input checked="" type="checkbox"/>		1 25.0
Lung	B stain	CFSE C57B6	10	25	1	50	<input checked="" type="checkbox"/>		1 50.0
Lymph node leg	B stain	C57B6 #2	2.5	25	1	25	<input checked="" type="checkbox"/>		1 25.0
Lymph node leg	B stain	C57B6 #3	2.5	25	1	25	<input checked="" type="checkbox"/>		1 25.0
Lymph node leg	B stain	CD22KO #4	2.5	25	1	25	<input checked="" type="checkbox"/>		1 25.0
Lymph node leg	B stain	CD22KO #5	2.5	25	1	25	<input checked="" type="checkbox"/>	CD5/Cychrome , IgD/Cy7-PE	1 25.0
									1 50.0

Viewing 45 of 60 tube plan rows!

...To a tree (hierarchical) view



Pipetting plan

Done Sample properties Sample pipetting Stain set pipetting Comfort % Plate setup Full table Help

Tree: Stain set, Tissue, Subject 9 tree selections filter 8 of 60 tube plan rows

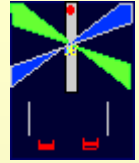
root

- B stain
 - Lung
 - C57B6 #2
 - C57B6 #3
 - CD22KO #4
 - CD22KO #5
 - CFSE C57B6
 - Lymph no...
 - C57B6 #2
 - C57B6 #3
 - CD22KO #4
 - CD22KO #5
 - CFSE C57B6
 - Lymph node ...
 - C57B6 #2
 - C57B6 #3
 - CD22KO #4
 - CD22KO #5
 - CFSE C57B6
 - Spleen
 - C57B6 #2
 - C57B6 #3
 - CD22KO #4
 - CD22KO #5
 - CFSE C57B6
- Mature B
 - Lung
 - C57B6 #2
 - C57B6 #3
 - CD22KO #4
 - CD22KO #5
 - CFSE C57B6
 - Lymph node
 - C57B6 #2

Stain set	Subject	Tissue	Include in assay	Fluorescence minus one	Stain volume (ul)
B stain	C57B6 #2	Lung	<input checked="" type="checkbox"/>		25.0
B stain	CD22KO #4	Lymph node	<input checked="" type="checkbox"/>		25.0
B stain	CD22KO #4	Lymph nod...	<input checked="" type="checkbox"/>		25.0
B stain	C57B6 #2	Spleen	<input checked="" type="checkbox"/>	1	25.0
B stain	C57B6 #3	Spleen	<input checked="" type="checkbox"/>	1	25.0
B stain	CD22KO #4	Spleen	<input checked="" type="checkbox"/>	1	25.0
B stain	CD22KO #5	Spleen	<input checked="" type="checkbox"/>	1	25.0
B stain	CFSE C57B6	Spleen	<input checked="" type="checkbox"/>	1	50.0

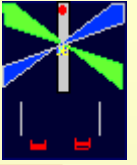
Viewing 60 of 60 tube plan rows!

Xpert grid/table supports



- **User customizability**
 - Customize column: names, sizes, display order and sort order
 - Filter rows
- **Persistence of user customizations**
 - Automatically remember and use the most recent customizations
 - Allow the user to save customizations
 - Allow the user to save and retrieve sets of customizations into property files
- **Tree-based read/write access**
 - Define a tree structure to summarize elements of the underlying table
 - Select any combination of tree nodes
 - See the column entries sieved by the node selections
 - Modify all cells in a column for the sieved rows

GUI extensibility



- Added pick list plug-in framework
- Added system menu plug-in and overrides
- Added numerous other slot widgets

