



## Protégé Plug-in Library: A Task-Oriented Tour

Tutorial at  
Seventh International Protégé Conference  
Bethesda MD, July 6 2004

Samson Tu and Jennifer Vendetti  
Stanford Medical Informatics  
Stanford University

1

## Goals

- Give a basic introduction to the Protégé plug-in architecture
- Describe a range of tasks in the life cycle of frame-based knowledge base development using Protégé
- Discuss Protégé plug-ins available for these tasks
- Answer questions
- Not to
  - Cover every single plug-in
  - Give a detailed tutorial on any one plug-in
  - Explain OWL plug-ins

2

## Outline

- Introduction
  - Review of plug-in types (Jennifer)
  - Review of plug-in architecture (Jennifer)
  - Describe example application and ontology (Samson)
- Tasks
  - Conceptualization (Samson)
  - Reusing/importing existing resources (Jennifer)
  - Visualizing ontologies (Jennifer)
  - Customizing displays (Jennifer)
  - Dealing with non-standard data types (Jennifer)
  - Navigating and browsing (Jennifer)
  - Validating integrity constraints (Samson)
  - Querying knowledge bases (Samson)
  - Publishing and exporting knowledge bases (Samson)
  - Developing applications (Samson)
  - Managing projects (Jennifer)
- Conclusion (Samson and Jennifer)

3

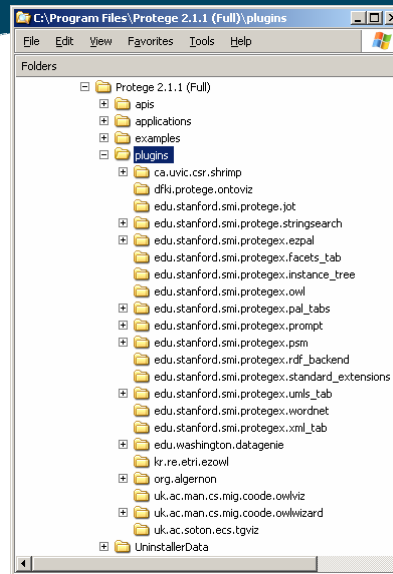
## Plug-in Types

- Tab widget
- Slot widget
- Back-end
- Import/Export
- Project

4

# Plug-in Architecture

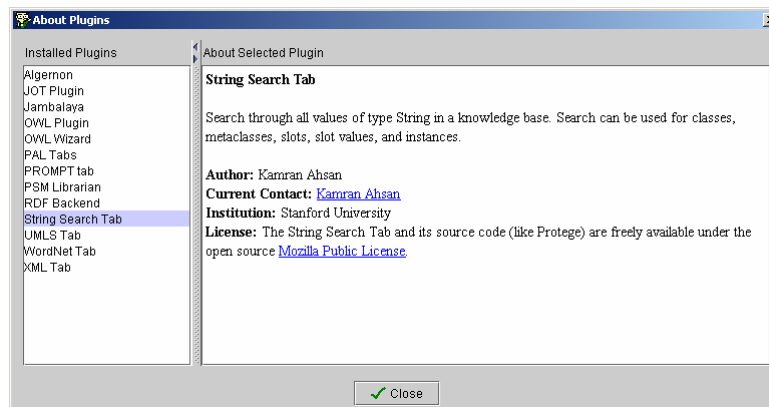
Where's my plug-in?



5

# Plug-in Architecture

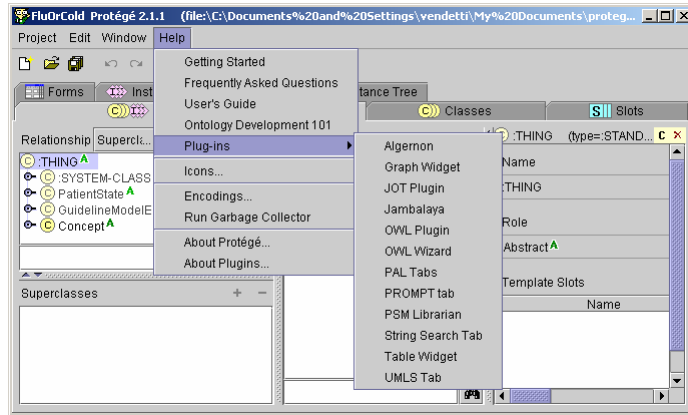
Built-in support for an "About Box"



6

# Plug-in Architecture

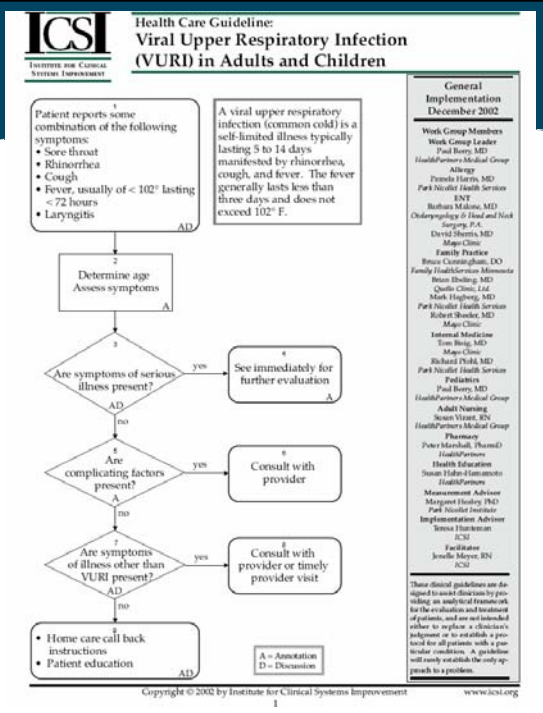
## Built-in support for documentation



7

## Example scenario

- Develop a medical decision-support application that generates recommendations based on clinical practice guideline
- Guideline example: management of common cold



## Conceptualization

- Protégé gives little support
- FCATab: Tab to support "Formal Concept Analysis" (FCA)
  - Really an export plug-in to support use of other tool
  - create 'context' table from Protégé classes and slots
  - generate input file for other FCA tool (ConExp)

Classes/Slots	ad_name	ad_salesperson	purchaser
.THING			
Content			
Advertisement	X	X	X
Article			
Author			
Layout_info			
Billing_Chart			
Employee			
Columnist			
Content_Layout			
Editor			
Person			
Library			
News_Service			
Newspaper			
Organization			
Personals_Ad	X	X	X
Prototype_New...			
Rectangle			

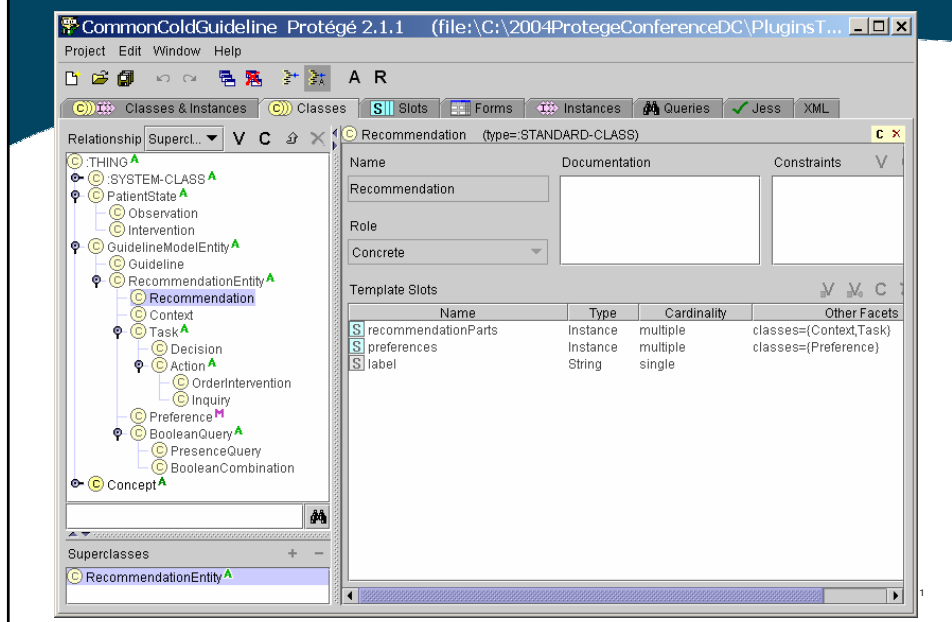
9

## Conceptualization of example domain

- A *guideline* is a set of *recommendations* consisting of
  - *Contexts* (e.g. presentation of symptoms)
  - *Tasks*
    - *Actions* (e.g. home care or referral)
    - *Decisions*: choice of action based on preference criteria (e.g. symptoms of serious problem)
- *Patient state* encodes information about a particular patient
  - e.g. Observations, prescribed medications
- Medical *concepts* represent abstractions of medical terminology
  - e.g. cough, fever, laryngitis

10

## Example ontology in Protégé



## Outline

- Introduction
  - Review of plug-in types (Jennifer)
  - Review of plug-in architecture (Jennifer)
  - Describe example application and ontology (Samson)
- Tasks
  - Conceptualization (Samson)
    - Reusing/importing existing resources (Jennifer)
  - Visualizing ontologies (Jennifer)
  - Customizing displays (Jennifer)
  - Dealing with non-standard data types (Jennifer)
  - Navigating and browsing (Jennifer)
  - Validating integrity constraints (Samson)
  - Querying knowledge bases (Samson)
  - Publishing and exporting knowledge bases (Samson)
  - Developing applications (Samson)
    - Managing projects (Jennifer)
- Conclusion (Samson and/or Jennifer)

## Reuse/import of existing resources

### Protégé Ontologies Library

- Gene Ontology
- HL7-RIM
- Guideline Interchange Format
- OWL Ontology Library

...

*Submissions welcomed and encouraged !!*

<http://protege.stanford.edu/ontologies/ontologies.html>

13

## Reuse/import of existing resources

*UMLS Tab Demo*

14

CommonColdGuideline Protégé 2.1.1 (file:\D:\MyDocuments\protegeworkshop\2004ProtegeConferenceDC\PluginsTutorial...

Project Edit Window Help

Classes Slots Forms Instances Queries

Relationship V C X

Concept (type=CodedConcept)

Name: [ ]

Concept: [ ]

Role: [ ]

Abstract: [ ]

Template Slots: [ ]

Superclasses: [ ]

CodeSystem: [ ]

Diagram:

```

    graph TD
      1[Patient reports some combination of the following symptoms:  
• Sore throat  
• Rhinorrhea  
• Cough  
• Fever, usually of < 102° lasting < 72 hours  
• Laryngitis] --> 2[Determine age  
Assess symptoms]
  
```

AD

A

15

CommonColdGuideline Protégé 2.1.1 (file:C:\Program%20Files\Protege%202.1.1%20(Full)\examples\protege%20conference%202004\Commo...

Project Edit Window Help

Classes Slots Forms Instances Queries UMLS

Search Word: Laryngitis

Exact Search All Sources

Approximate search 100 results

Narrow Tree Size 50 nodes

Search

Search Result

Concept Name	Semantic Type
laryngitis	Finding, Disease or Syndrome

laryngitis

Broader Narrow Related UMLS Narrow Tree

Definition

Inflammation of the larynx. This condition presents itself with dryness and soreness of the throat, difficulty in swallowing, cough, and hoarseness.

Instances

Classes

THING A

SYSTEM-C

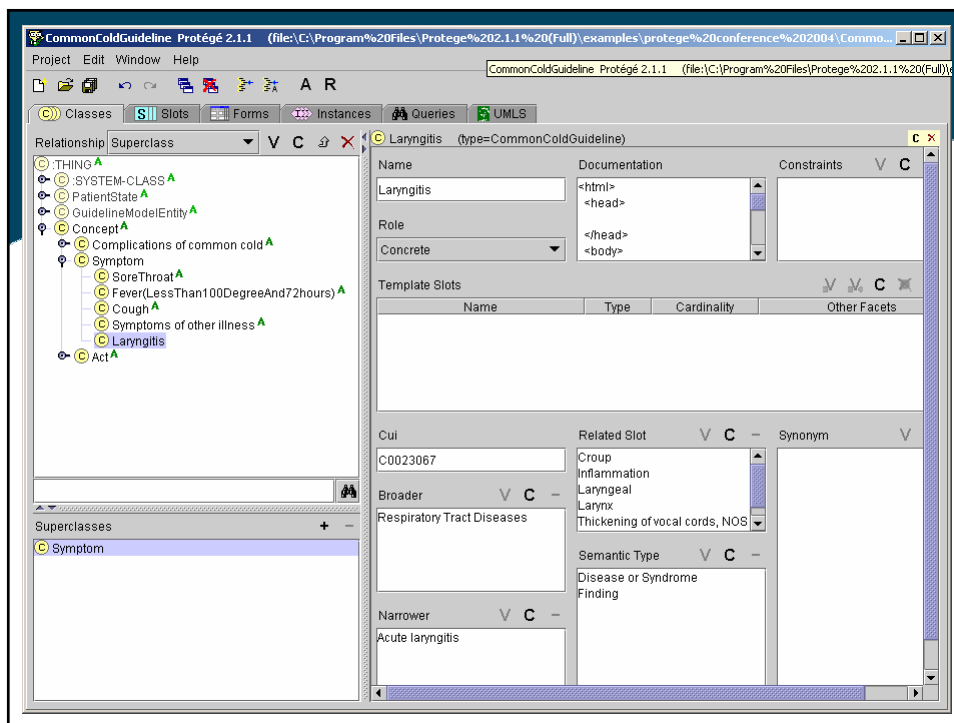
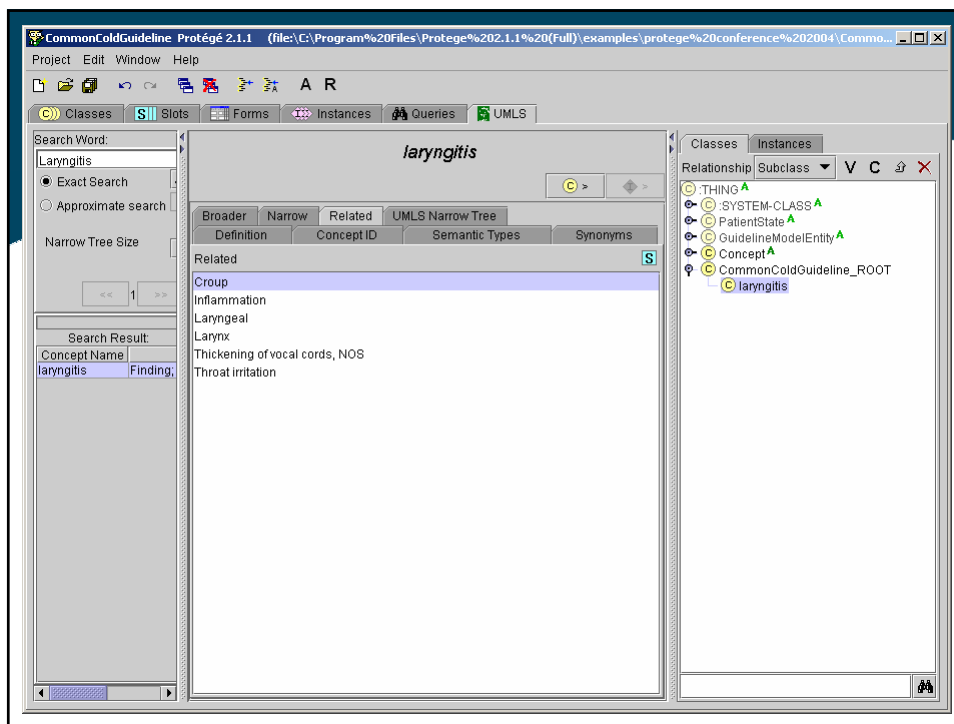
PatientStat

GuidelineM

Concept A

CommonC





## Reuse/import of existing resources

- DataGenie – import data from arbitrary databases
- OKBC Tab – import/export OKBC compliant ontologies
- OntoBase – read, navigate, update arbitrary databases
- XML Tab – import arbitrary XML documents
- WordNet Tab – import lexical content from WordNet
- TXRuleML Tab – RuleML to taxonomic class hierarchies
- Jess Tab – Jess scripting for... well... anything really

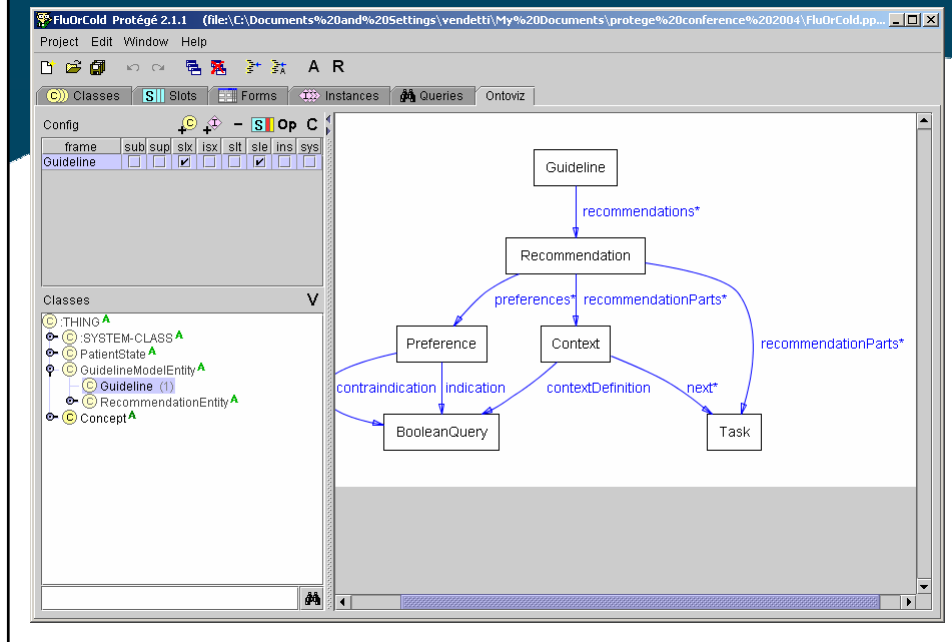
19

## Visualizing ontologies

- OntoViz Tab – visualize ontologies with GraphViz
- Jambalaya – visualize ontologies with SHriMP (Simple Hierarchical Multi-Perspective)
- TGViz – visualize ontologies with TouchGraph

20

## OntoViz tab demo



## Customizing displays

### Graph Widget

- Alternative to Forms for creating and populating instances of classes
- Visualize networks of instances and relationships between instances

## Graph Widget Demo

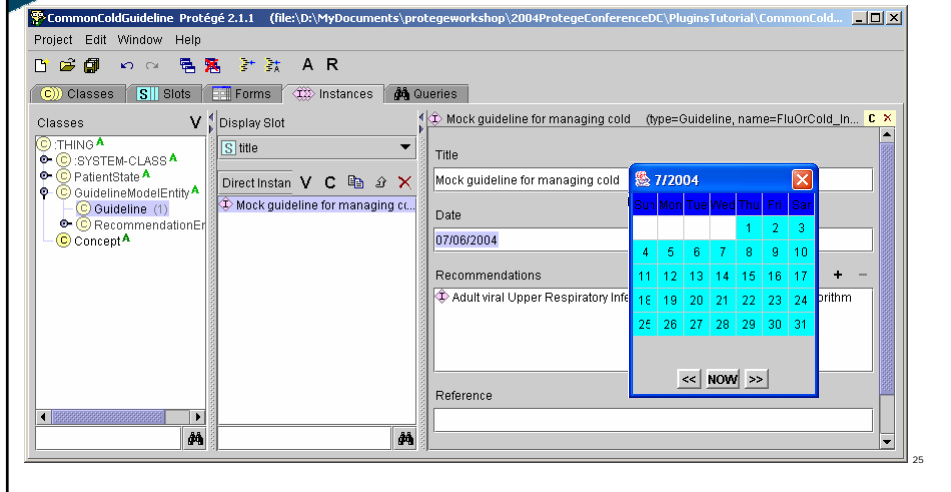
The screenshot displays the Protegé 2.1.1 software interface. The main window is titled "Adult viral Upper Respiratory Infection (common cold) guideline algorithm". The interface is divided into several panes:

- Classes:** A tree view on the left showing a hierarchy of classes including `THING`, `SYSTEM-CLASS`, `PatientState`, `GuidelineModelEntity`, `Guideline (1)`, `RecommendationEntity`, `Recommendation (1)`, `Context (1)`, `Task`, `Preference (2)`, `BooleanQuery`, and `Concept`.
- Display Slot:** A central pane showing a "Label" slot with the text "Infection (common cold) guideline algorithm".
- Recommendation Nodes:** A large grid area on the right containing a flowchart. The flowchart starts with a red circle labeled "Context". An arrow points to a yellow hexagon labeled "Decision". From the "Decision" node, an arrow labeled "Inquiry" points to a green diamond labeled "Assess symptoms". From "Assess symptoms", an arrow points to a yellow hexagon labeled "Home care?". From "Home care?", an arrow labeled "Home care OK" points to a blue rectangle labeled "Referral". Another arrow labeled "Referral necessary" points from "Home care?" to the "Referral" node. On the left side of the grid, there are several other nodes: a red circle "Context", a yellow hexagon "Decision", a blue rectangle "Order/Intervention", and a green diamond "Inquiry".

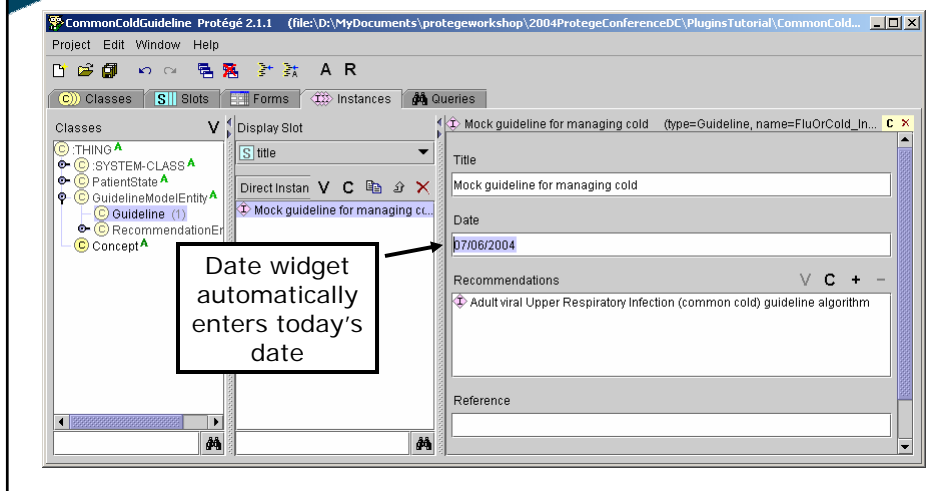
## Dealing with non-standard data types

- Calendar Widget
- Date Widget
- URL Widget
- Image Widget
- Indirect Instances

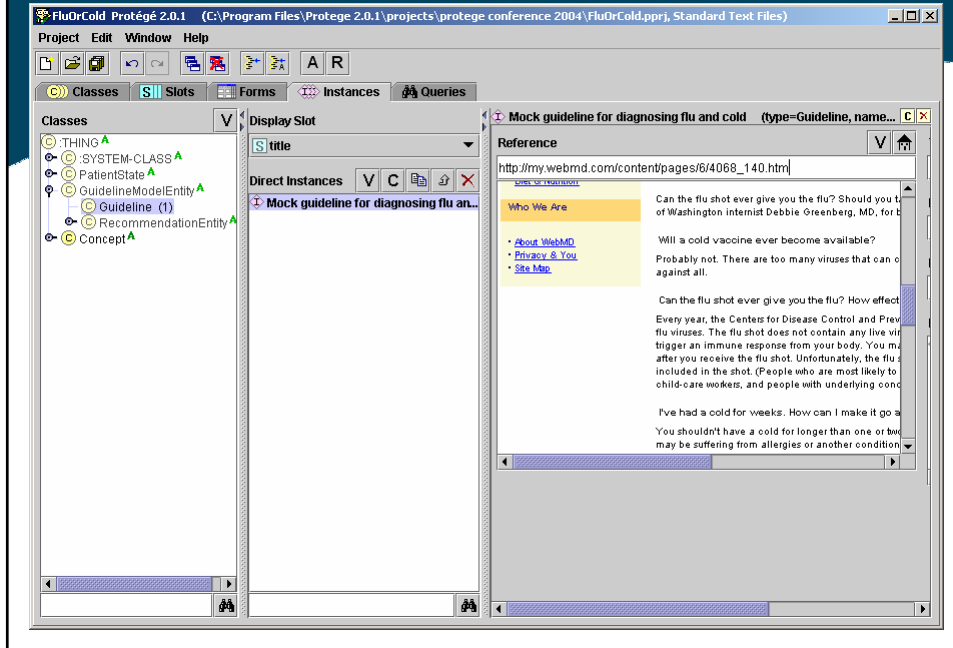
# Calendar Widget



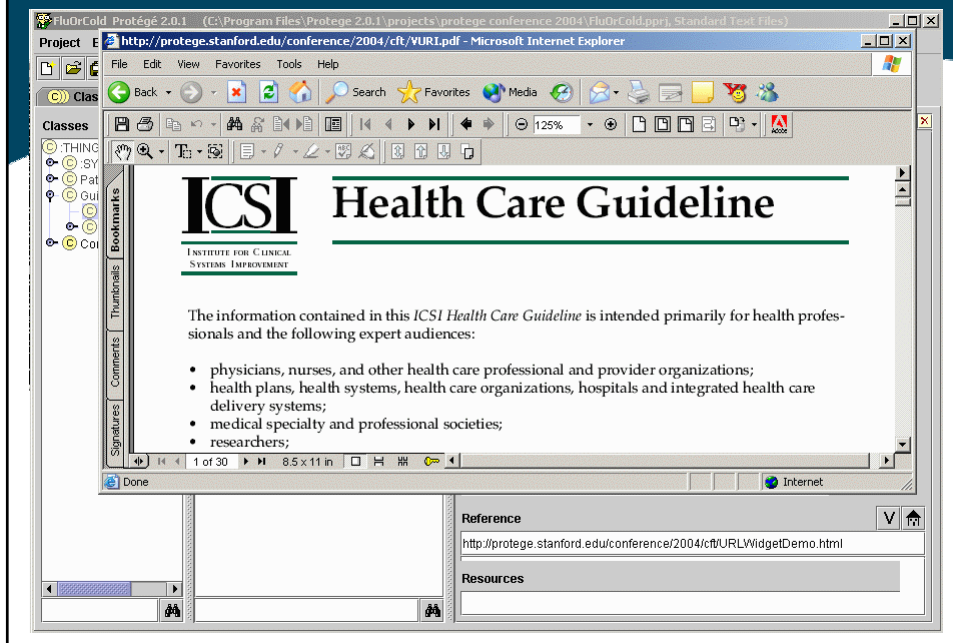
# Date Widget



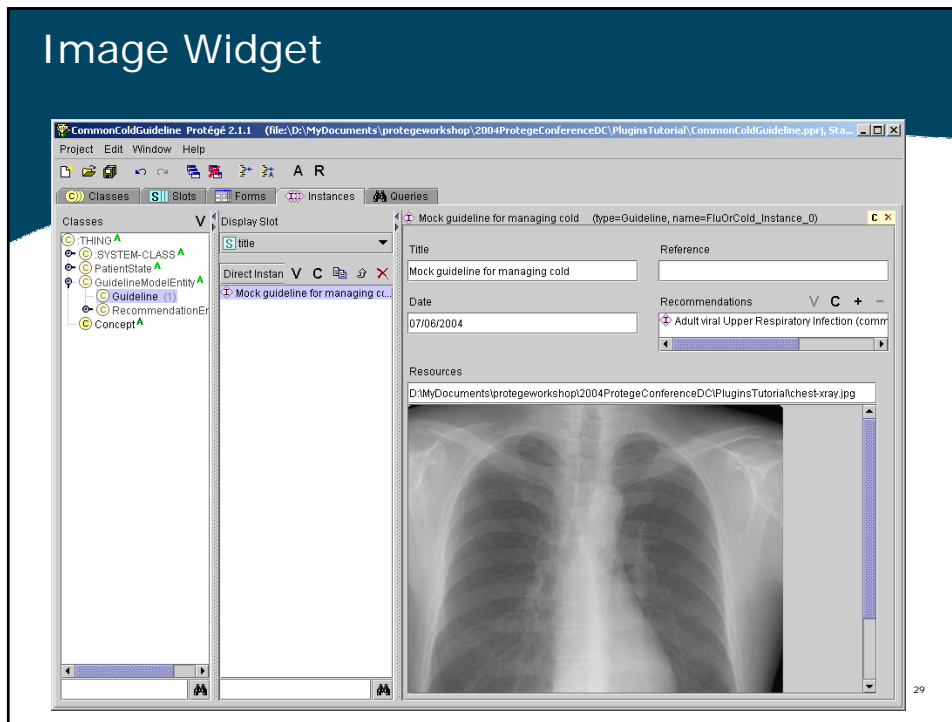
# URL Widget



# URL Widget 2



## Image Widget



## Navigating & Browsing

- Instance Tree
  - view instances of classes as root nodes of trees
  - trees contain directly and indirectly referenced frames
- Knowledge Tree
  - designate a top-level instance and navigate a tree of "contained" instances

## Instance Tree & Knowledge Tree Tabs Demo

The screenshot shows the Protege software interface with the following components:

- Project:** FluOrCold Protégé 2.1.1 (file: C:\Documents%20and%20Settings\wendetti\My%20Documents\protege%20conference%202004\FluOrCold.pprj...)
- Menu:** Project Edit Window Help
- Toolbar:** Includes icons for file operations and a toolbar with 'A' and 'R' buttons.
- Class Browser (Left):** Shows a hierarchy of classes including .THING, :SYSTEM-CLASS, PatientState, GuidelineModelEntity, Guideline (1), RecommendationEnt, and Concept.
- Instance Tree (Center):** Displays a tree structure for the 'Guideline' class. The root is 'Mock guideline for diagnosing flu and cold', which has a 'recommendations' slot. This slot contains an instance 'Adult viral Upper Respiratory Infection (common cold) guideline algorithm', which in turn has a 'recommendationParts' slot. This slot contains several instances: 'presentation of symptoms', 'Home care?', 'Assess symptoms', 'Home care', 'Referral', 'preferences', 'Home care OK', and 'Referral necessary'.
- Knowledge Tree (Right):** Shows the 'Guideline' class with its 'recommendations' slot. The slot contains three instances: 'title', 'date', and 'recommendations'.

## Outline

- Introduction
  - Review of plug-in types (Jennifer)
  - Review of plug-in architecture (Jennifer)
  - Describe example application and ontology (Samson)
- Tasks
  - Conceptualization (Samson)
  - Reusing/importing existing resources (Jennifer)
  - Visualizing ontologies (Jennifer)
  - Customizing displays (Jennifer)
  - Dealing with non-standard data types (Jennifer)
  - Navigating and browsing (Jennifer)
  - Validating integrity constraints (Samson)
  - Querying knowledge bases (Samson)
  - Publishing and exporting knowledge bases (Samson)
  - Developing applications (Samson)
  - [Managing projects](#) (Jennifer)
- Conclusion (Samson and/or Jennifer)



## Validating integrity constraints

- Facet Constraint Tab
  - Protégé facets are constraints on values of slots (e.g. minimum cardinality)
  - FacetConstraint Tab brings all instances with facet constraint violations together in one place
- PAL Constraint Tab
  - Protégé Axiom Language (PAL) lets you write integrity constraints across multiple slots and multiple instances
  - PAL constraint tab allows checking of PAL constraints
- EZPAL Tab
  - Provides templates for easier authoring of PAL constraints

33

## Facet-constraint tab

The screenshot displays the Protégé interface with the Facet Constraints tab active. The main window shows a project titled 'IntegrityConstraintsExample Protégé 2.1.1'. The interface includes a menu bar (Project, Edit, Window, Help, PAL Constraints) and a toolbar with icons for various actions. The main workspace is divided into several panes:

- Classes:** A tree view showing a hierarchy of classes including `THING`, `SYSTEM-CLASS`, `PatientState`, `GuidelineModelEntity`, `Property`, `Template`, and `Concept`.
- Classes with Instance Violations:** A list of classes that have instances with facet-constraint violations, including `BooleanCombination` (1), `CodedConcept` (1), `Context` (1), `Decision` (1), and `Inquiry` (1).
- Instances with:** A list of instances that have facet-constraint violations, including `common cold symptoms`.
- Facet Constraints:** A pane showing the details of a facet constraint, including a label, a value (e.g., `BooleanOperator`), and a description (e.g., `booleanOperator: A value is required`).

Annotations in the image highlight key features:

- A box labeled "Classes with instances that have facet-constraint violations" points to the "Classes with Instance Violations" pane.
- A box labeled "Instances that have facet-constraint violations" points to the "Instances with" pane.
- A box labeled "Button to start facet-constraint checking" points to the "Check Facet Constraints of Selected Classes" button at the bottom left.

## PAL-constraint tab

- Allows specification and checking of complex integrity constraints

The screenshot shows the 'PAL Constraints' window in Protégé. A constraint is selected, and its details are shown in a separate window titled 'All decisions are'. The constraint name is 'All decisions are part of recommendations'. The range is defined as `(defrange ?decision :FRAME Decision) (defrange ?recomm :FRAME Recommendation)`. The statement is `(forall ?decision (exists ?recomm (recommendationParts ?recomm ?decision)))`. Annotations include:
 

- 'Instances with PAL constraint violations' pointing to the constraint list.
- 'PAL constraint being checked' pointing to the 'All decisions are' window title.
- 'Example of PAL constraint' pointing to the constraint details window.

## EZPAL tab

- Templates for fill-in-the-blanks method of defining PAL constraints

The screenshot shows the 'EZPAL' window in Protégé. A template is selected, and its details are shown in a separate window titled 'Generated PAL const...'. The template name is blank. The range is defined as `(defrange ?Context :FRAME Context) (defrange ?Recommendation :FRAME Recommendation)`. The statement is `(forall ?Context (exists ?Recommendation (recommendationParts ?Recommendation ?Context)))`. Annotations include:
 

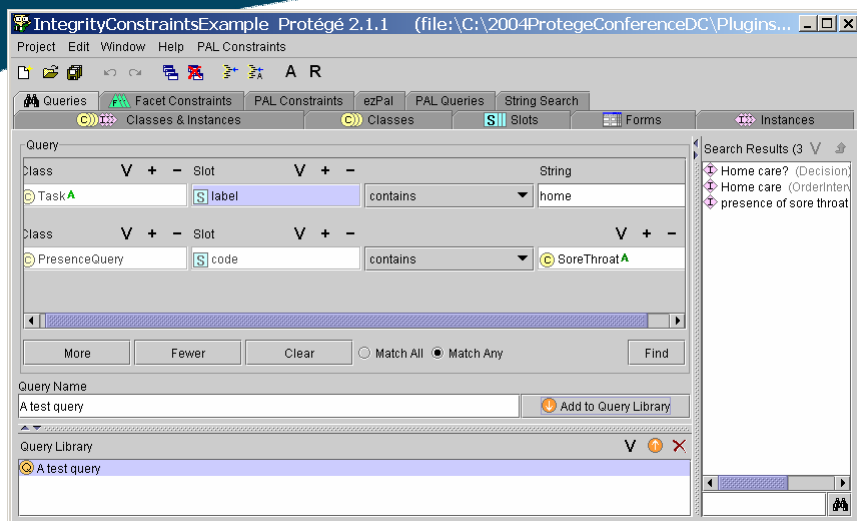
- 'Templates for classes of PAL constraints' pointing to the template list.

## Querying knowledge bases

- **QueryTab**: search instances of a class
- **StringSearch**: search for a string in entire project
- **PALQueryTab**: complex search condition
- **JessTab/JOT/Algernon**: programming tools with tab interface
- **SearchAPI**: Java API for constructing complex criteria for searching instances in entire project

37

## Query tab: search for instances of a class whose slots satisfy some criteria



38

## String search: search for a string anywhere in the project

The screenshot shows the Protege 2.1.1 interface with the 'String Search' tab active. The search term 'home' is entered in the search box, and the results table below shows 8 matches. The table columns are Frame, Direct Type, Matched Slot, and Matched Value. The results include frames like HomeCare, Home care?, Assess symptoms, Referral necessary, Home care OK, and Adult viral Upper R... with various direct types such as CodedConcept, Decision, Inquiry, Preference, and RecommendationEntity.

Frame	Direct Type	Matched Slot	Matched Value
HomeCare	CodedConcept	NAME	HomeCare
Home care?	Decision	alternatives	Home care?
Home care?	Decision	label	Home care?
Assess symptoms	Inquiry	next	Home care?
Referral necessary	Preference	FROM	Home care?
Home care OK	Preference	label	Home care OK
Adult viral Upper R...	RecommendationEntity	preferences	Home care OK
Home care	Orderintervention	label	Home care

To the right, a decision graph is visible, showing a flow from 'Assess symptoms' to 'Home care?' and then to 'Home care OK' and 'Home care'. A 'Referral' node is also shown, connected to 'Home care?' via a 'Referral necessary' edge.

## PAL Query

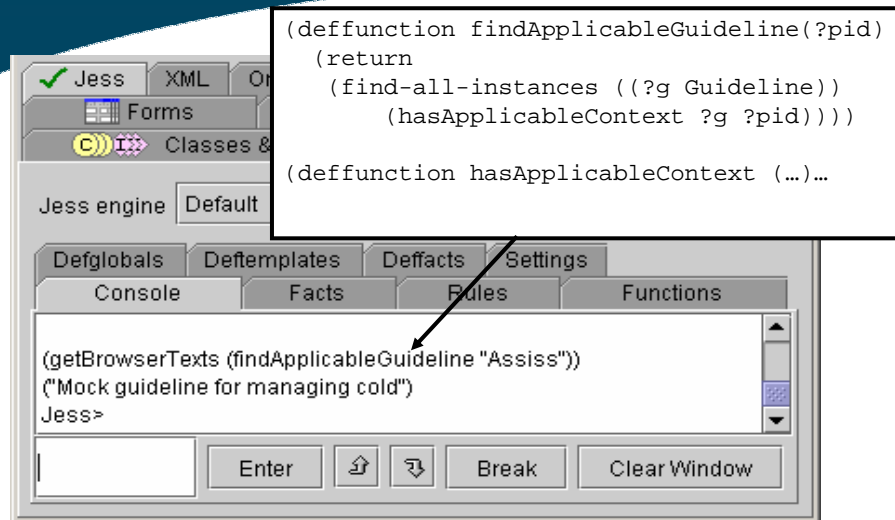
- Allows search with complex search criteria

The screenshot shows the Protege 2.1.1 interface with a PAL Query window open. The query is titled 'Nodes that should be decision alternatives'. The query text is as follows:

```
(findall ?node
  (exists ?decision (and (exists ?preference (and (FROM ?preference ?decision)
    (TO ?preference ?node)))
    (not (alternatives ?decision ?node))))))
```

The query description states: 'Find all all nodes following a Decision that are not a value of the "alternatives" slot of the decision'. The range is defined as: (defrange ?node :FRAME RecommendationEntity) (defrange ?decision :FRAME Decision) (defrange ?preference :FRAME Preference).

## JesTab as a query interface



41

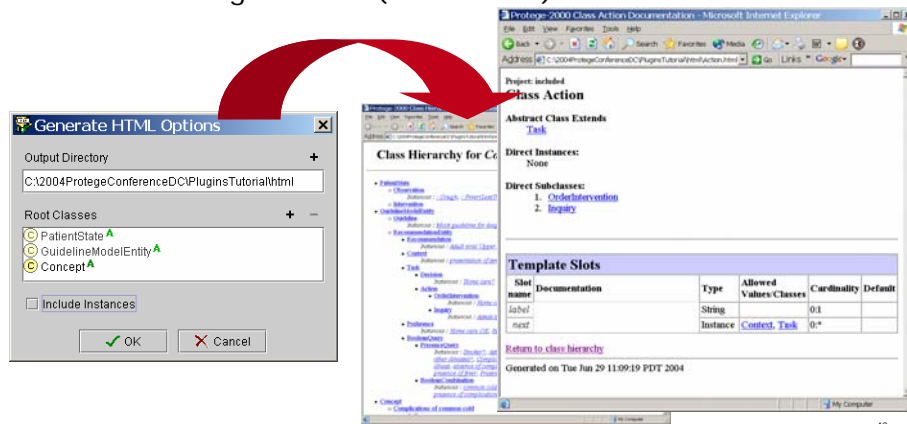
## Exporting and publishing and knowledge bases

- Alternative Protégé backends
  - **XML Schema** – saves a Protégé project using a fixed Protégé XML schema
  - **XML ontology** – saves a Protégé project by creating an XML schema based on the ontology in the project
  - **RDF, OWL** – save in formats used in semantic web formats
- Export functionalities
  - **HTML export** – Java-doc style html pages for classes and instances
  - **XMLtab** – XML documents for classes and/or instances
  - **JesTab/JOT/Algernon** – general purpose programming
  - TMRuleML, -export to specialized formats
- “Publishing”
  - **ProtegeWebBrowser** – browse Protégé projects on the web
  - **Protégé CORBA server** – access Protégé projects through CORBA interfaces

42

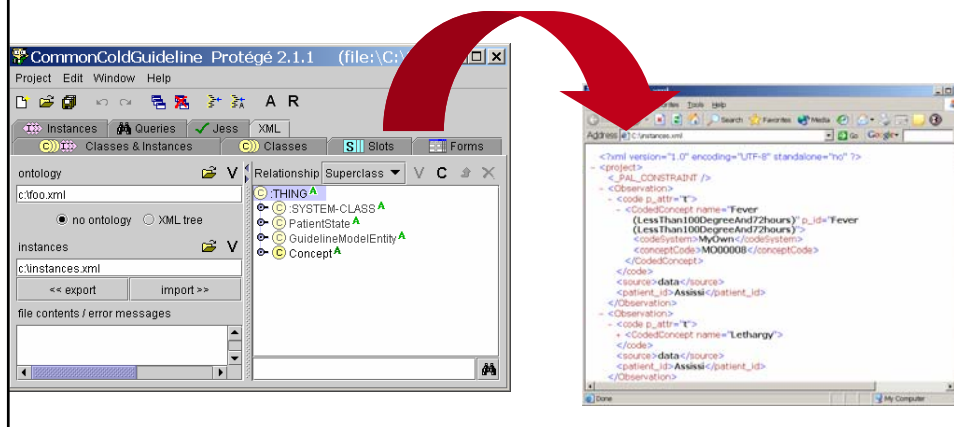
## HTML export

- Creates "Javadoc-style" HTML pages for Protégé classes (& instances)



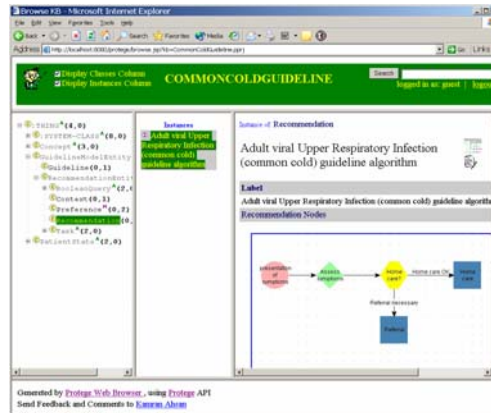
## XML tab as exporter

- Export classes and instances in separate XML files



## Publishing: Protégé web browser

- Requires installation servlet-capable web server (e.g. Apache Tomcat)
- Present read-only Protégé KB on the web
- Possible to add annotations
- Possible to get screen shots of Protégé GUI associated with each instance
- Possible to download projects



45

## Application development: different approaches (1)

- Use Protégé's Java application programming interface (API) to access and manipulate knowledge base
  - **JSave** creates Java class stubs based on Protégé classes
- Export Protégé knowledge to other environments
  - RDF XML files read as input to applications that use XML as inputs
  - **BeanGenerator**, **CLIPSTab**: export to agent/rule-programming environments

46

## Application development: different approaches (2)

- Plug-ins: Use programming paradigms that have been made interoperable with Protégé
  - **JessTab, Algernon**: Rule-based programming
  - **Prolog** tab: Logic-based programming
  - **JOT** (Jython), **JessTab, Algernon**: Scripting environment
- In the future: Protégé come with pre-made problem-solving methods that can be configured for different applications
  - **PSMTab**: allows mapping of domain knowledge bases to “method ontology” of generic problem-solving methods

47

## Application development: JessTab example

- Jess: Java Expert System Shell developed at Sandia National Laboratory, USA
  - Forward-chaining rule engine that match
  - Powerful scripting language
- JessTab: integrate Jess with Protégé
  - Mapping between Jess and Protégé
    - Protégé classes mapped to a Jess fact template
    - Protégé instances mapped to Jess facts and Jess facts mapped to instances
    - Changes to mapped facts in Jess reflected in Protégé; changes in Protégé reflected in Jess
  - Allows Jess to be run within Protégé GUI
  - Possible to embed both Jess and Protégé in Java program

48



## JesTab: Rule-based programming in Protégé

- If a patient has problem, then conclude that he/she has more generalize problem

```
(defrule ObservationDeductions
  (object (is-a Observation)      protégé class
   (code ?code)                  protégé slots
   (patient_id ?pid))
  ?super<-(object(is-a CodedConcept))
  (test (superclassp ?super ?code))
  =>
  (make-instance of Observation
   (code ?super)
   (patient_id ?pid)
   (source guideline)
  ))
```

creating protégé instance

49

## Outline

- Introduction
  - Review of plug-in types (Jennifer)
  - Review of plug-in architecture (Jennifer)
  - Describe example application and ontology (Samson)
- Tasks
  - Conceptualization (Samson)
  - Reusing/importing existing resources (Jennifer)
  - Visualizing ontologies (Jennifer)
  - Customizing displays (Jennifer)
  - Dealing with non-standard data types (Jennifer)
  - Navigating and browsing (Jennifer)
  - Validating integrity constraints (Samson)
  - Querying knowledge bases (Samson)
  - Publishing and exporting knowledge bases (Samson)
  - Developing applications (Samson)
  - Managing projects (Jennifer)
- Conclusion (Samson and/or Jennifer)

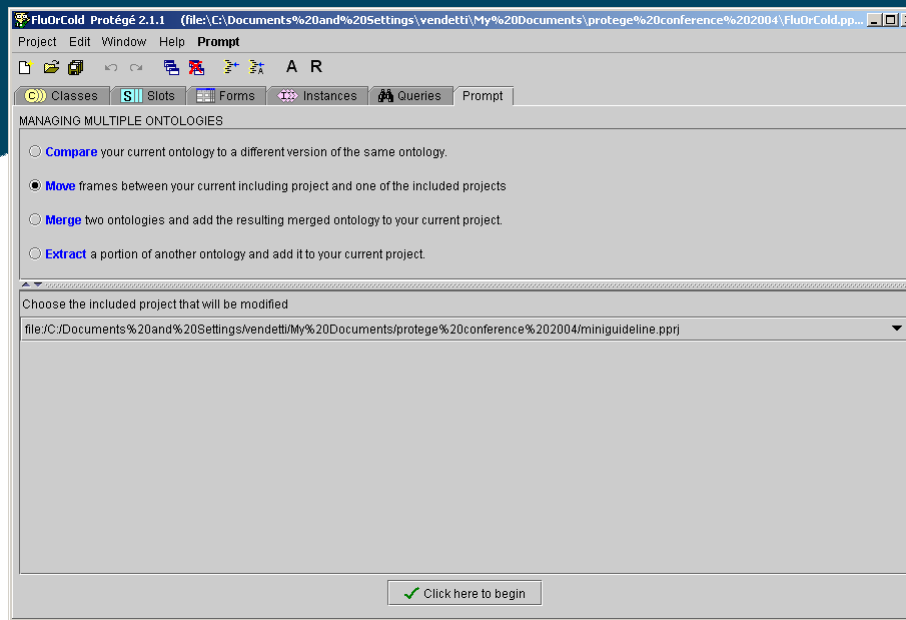
50

## Managing Projects

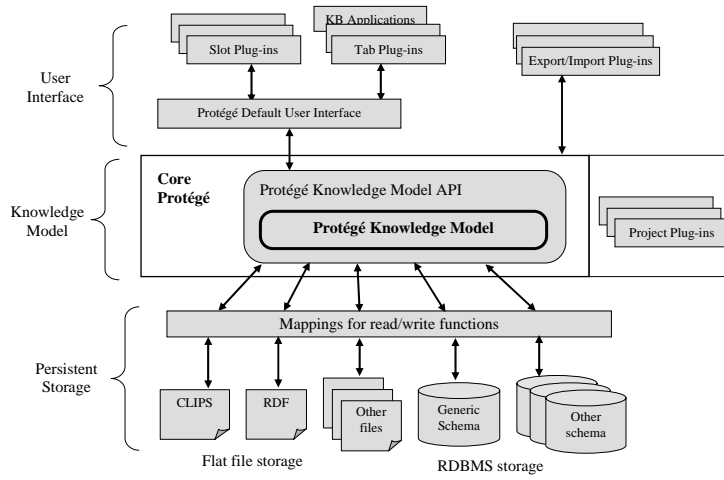
- Project Browser – manage libraries of Protégé projects
- Prompt
  - Move classes/instances up and down inclusion lattices
  - Extract portions of ontologies
  - compare versions of the same ontology

51

## Prompt Tab Demo



# Summary



53