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

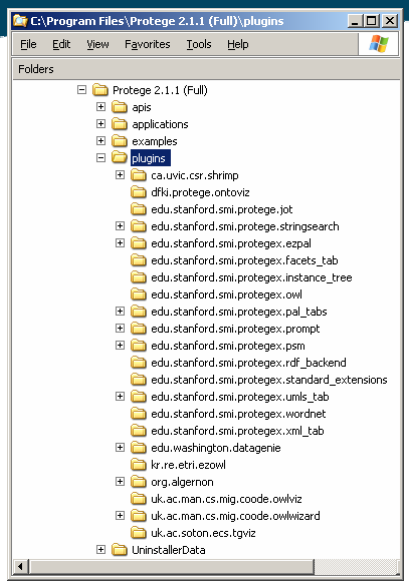
Plug-in Types

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

4

Plug-in Architecture

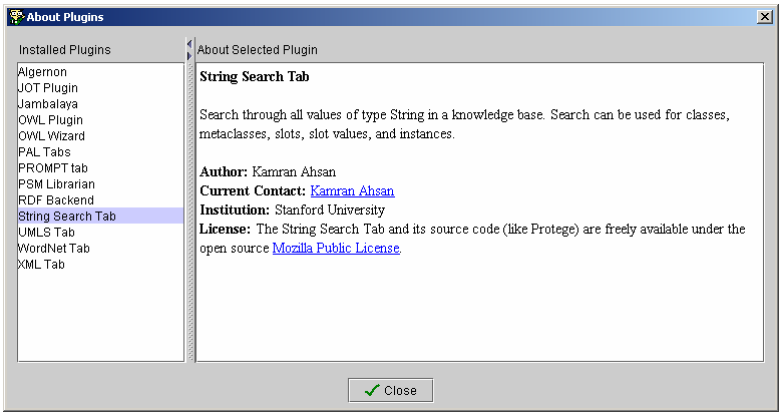
Where's my plug-in?



5

Plug-in Architecture

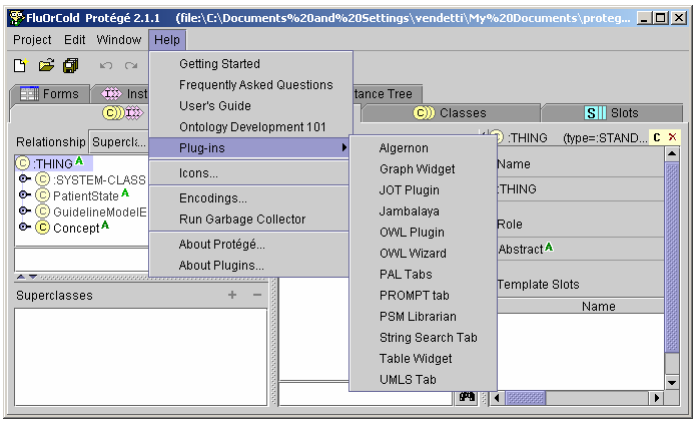
Built-in support for an "About Box"



6

Plug-in Architecture

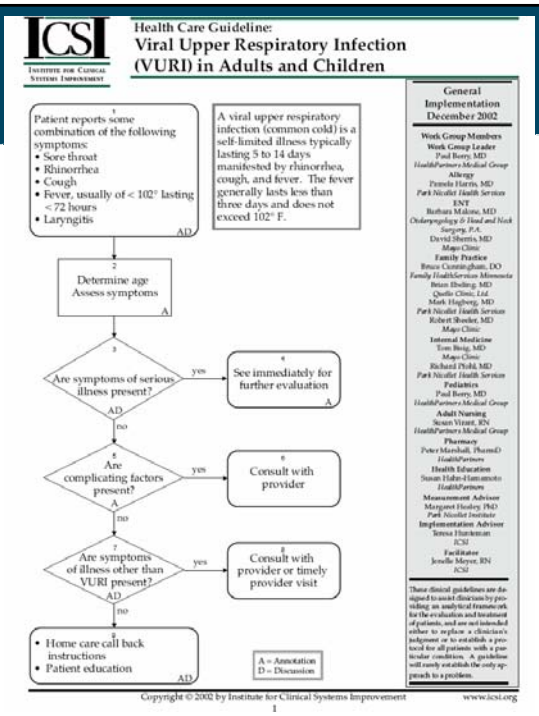
Built-in support for documentation



7

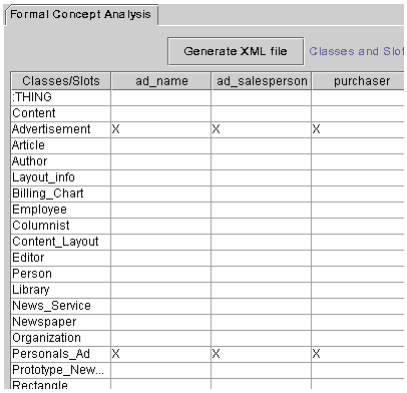
Example scenario

- 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			

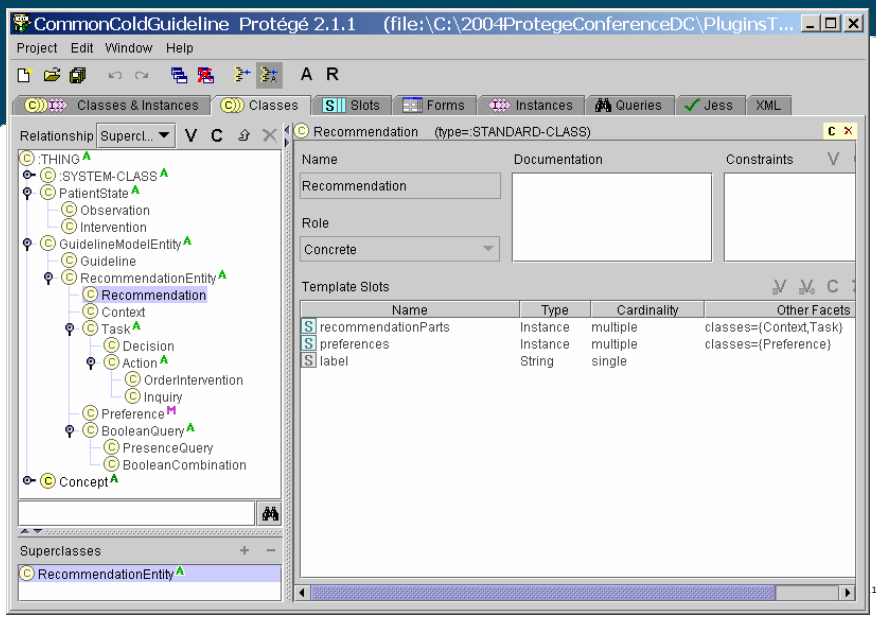
9

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



11

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)

12

Reuse/import of existing resources

Protégé Ontologies Library

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

...

Submissions welcomed and encouraged !!

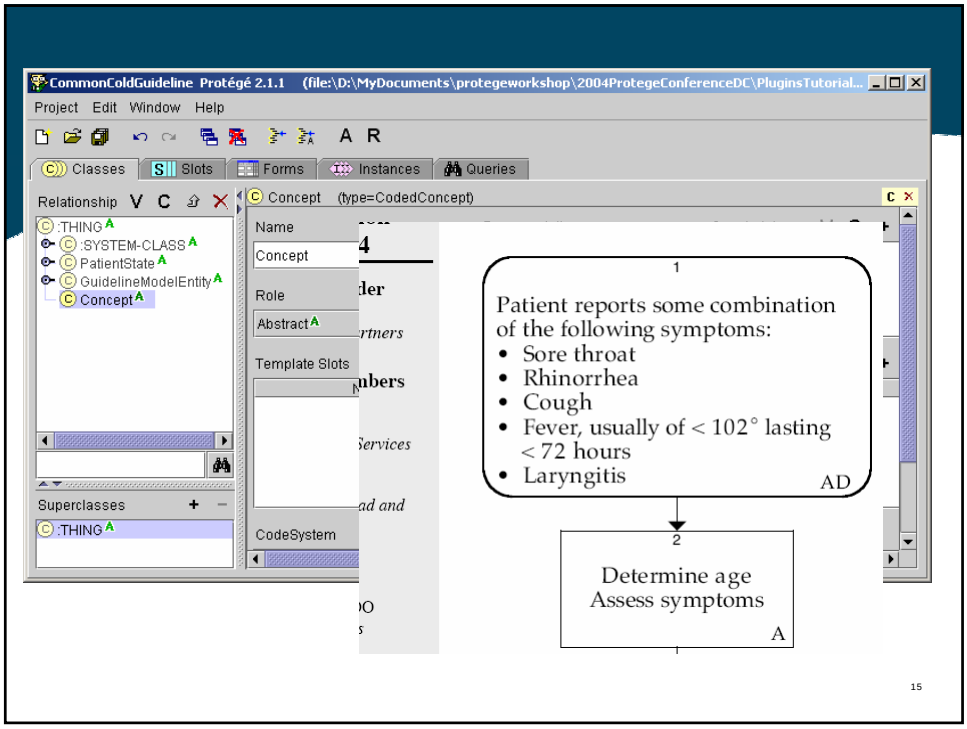
http://protege.stanford.edu/ontologies/ontologies.html

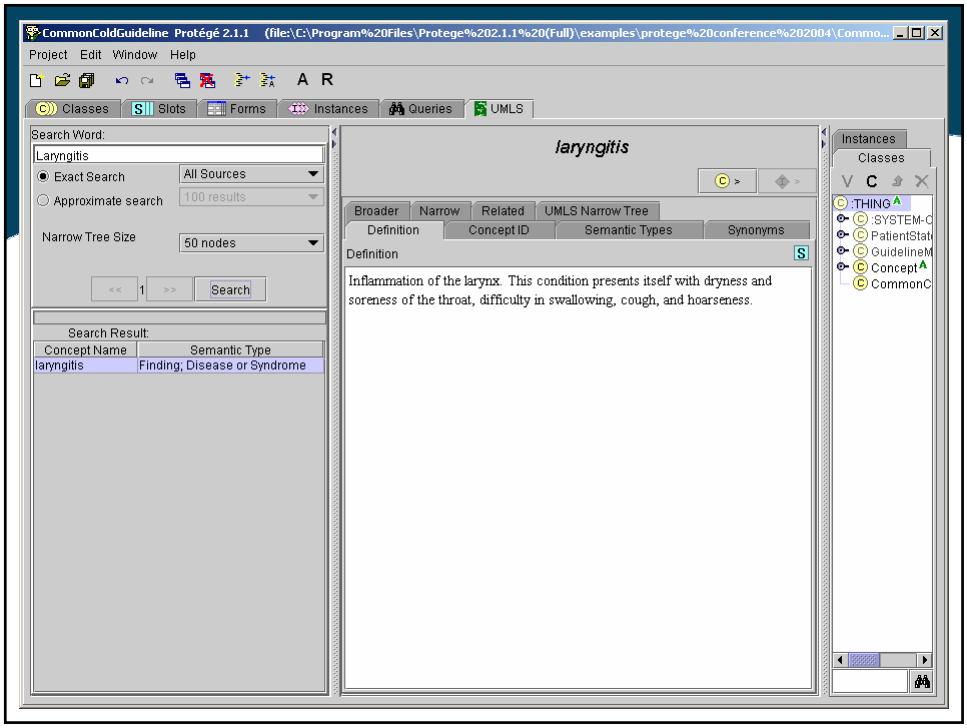
13

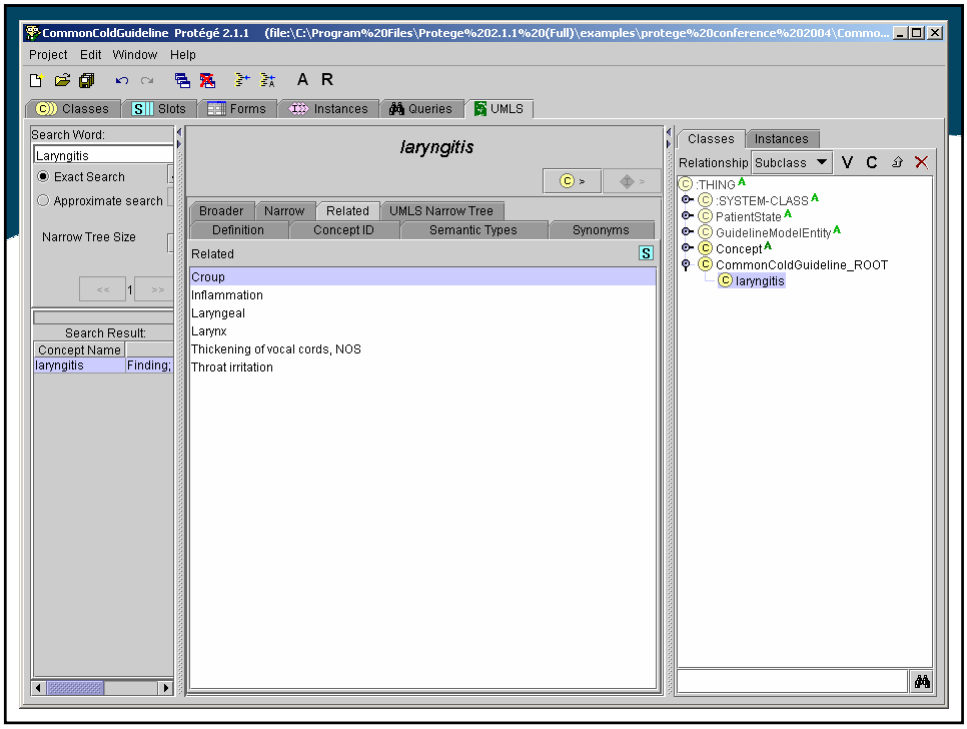
Reuse/import of existing resources

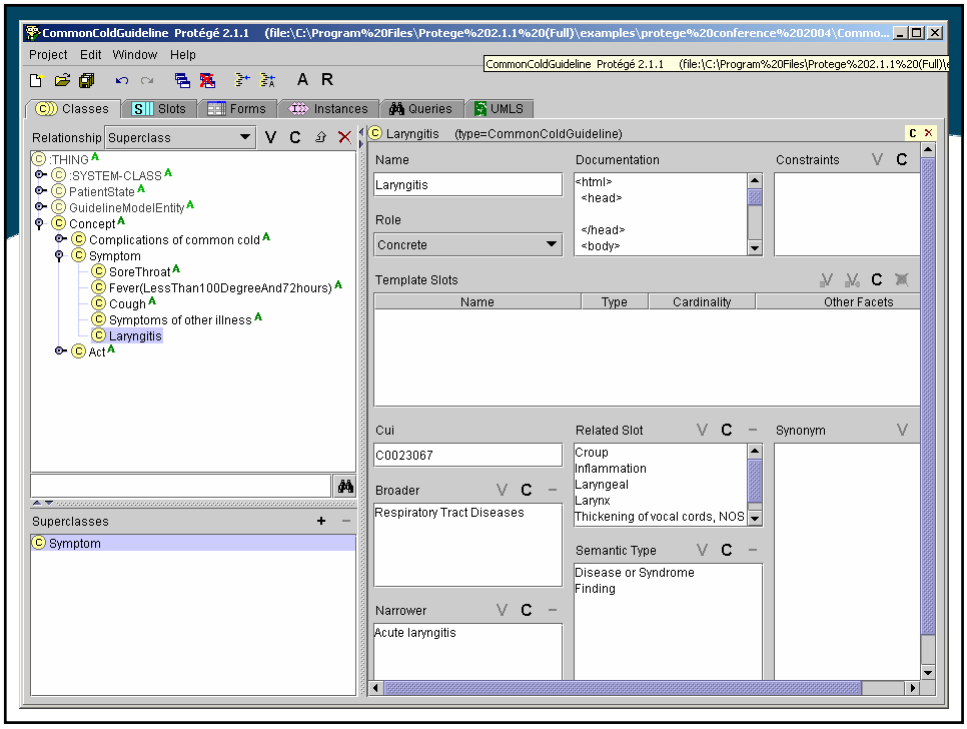
UMLS Tab Demo

14


CommonColdGuideline Protégé 2.1.1 (file:\D:\MyDocuments\protegeworkshop\2004ProtegeConferenceDC\PluginsTutorial...
Project Edit Window Help
Classes Slots Forms Instances Queries
Relationship
:THING
:SYSTEM-CLASS
PatientState
GuidelineModelEntity
Concept
Concept (type=CodedConcept)
Name
Concept
Role
Abstract
Template Slots
CodeSystem
Superclasses
:THING
1
Patient reports some combination of the following symptoms:
Sore throat
Rhinorrhea
Cough
Fever, usually of < 102° lasting < 72 hours
Laryngitis
AD
2
Determine age
Assess symptoms
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
Approximate search
All Sources
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 Concept ID Semantic Types Synonyms
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
Concept
CommonC


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
Approximate search
Narrow Tree Size
Search Result:
Concept Name
laryngitis Finding;
laryngitis
Broader Narrow Related UMLS Narrow Tree
Definition Concept ID Semantic Types Synonyms
Related
Croup
Inflammation
Laryngeal
Larynx
Thickening of vocal cords, NOS
Throat irritation
Classes Instances
Relationship Subclass
:THING
:SYSTEM-CLASS
PatientState
GuidelineModelEntity
Concept
CommonColdGuideline_ROOT
laryngitis


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
Relationship Superclass
:THING
:SYSTEM-CLASS
PatientState
GuidelineModelEntity
Concept
Complications of common cold
Symptom
SoreThroat
Fever(LessThan100DegreeAnd72hours)
Cough
Symptoms of other illness
Laryngitis
Act
Superclasses
Symptom
Laryngitis (type=CommonColdGuideline)
Name
Laryngitis
Role
Concrete
Documentation
<html>
<head>
</head>
<body>
Constraints
Template Slots
Name Type Cardinality Other Facets
Cui
C0023067
Broader
Respiratory Tract Diseases
Narrower
Acute laryngitis
Related Slot
Croup
Inflammation
Laryngeal
Larynx
Thickening of vocal cords, NOS
Semantic Type
Disease or Syndrome
Finding
Synonym

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

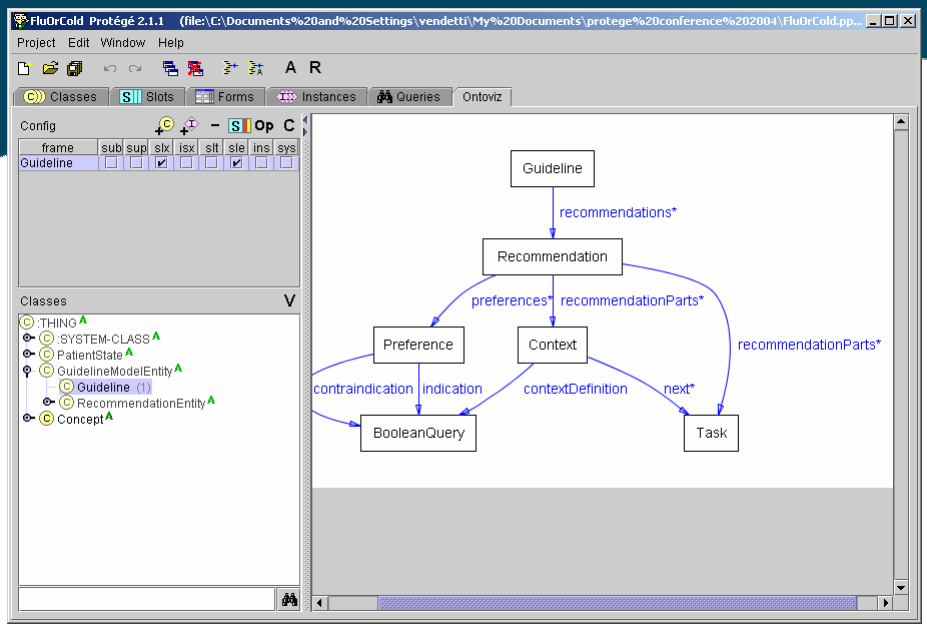
19

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



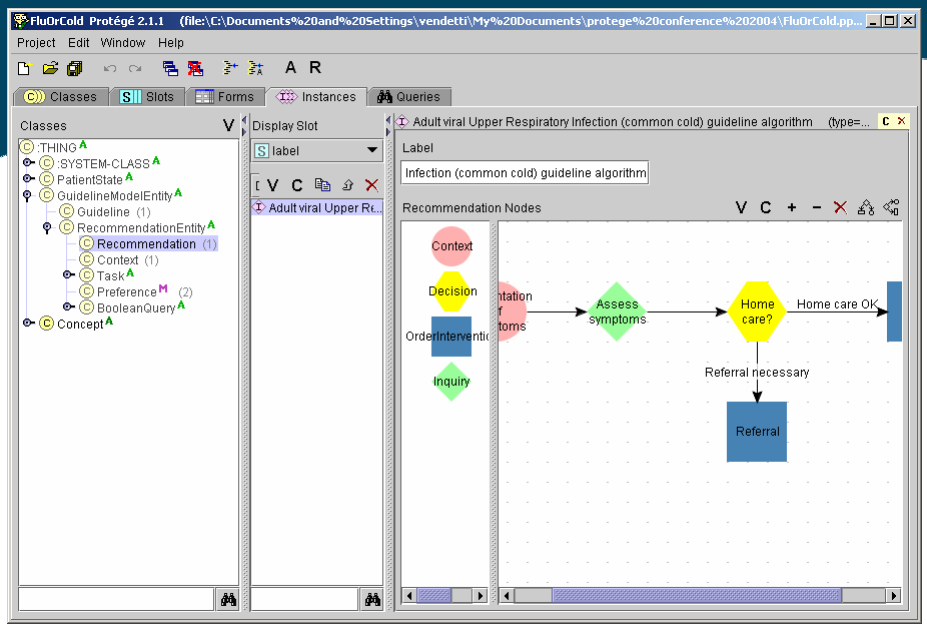
Customizing displays

Graph Widget

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

22

Graph Widget Demo

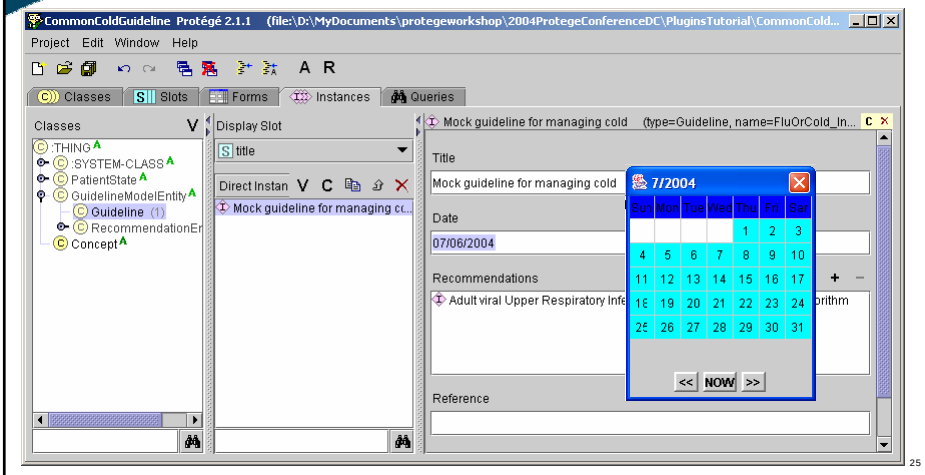


Dealing with non-standard data types

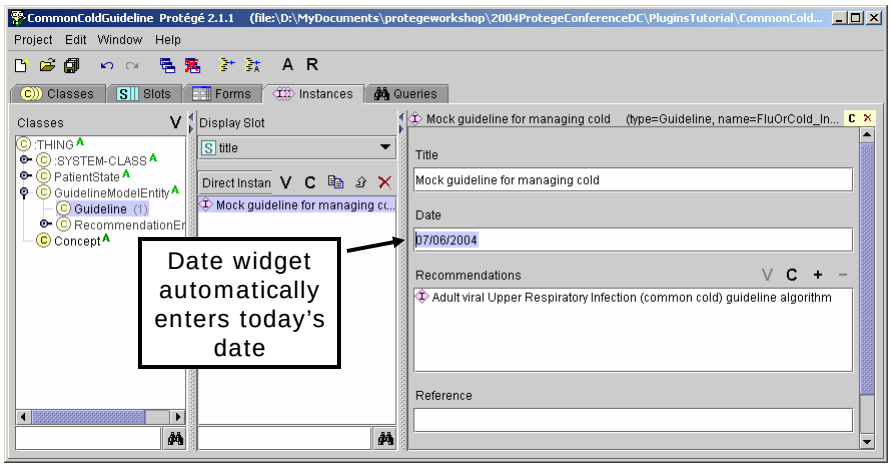
- Calendar Widget
- Date Widget
- URL Widget
- Image Widget
- Indirect Instances

24

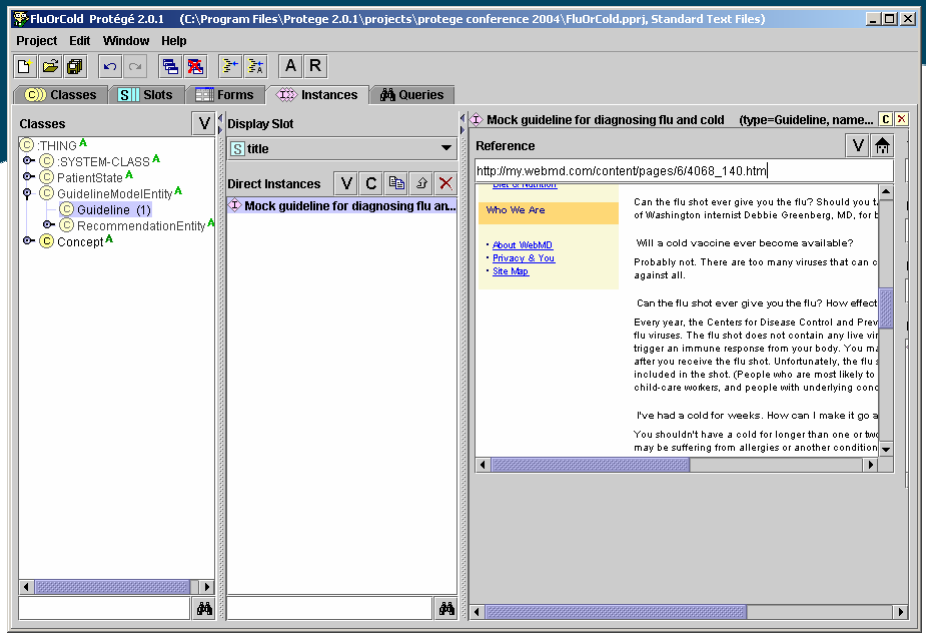
Calendar Widget



Date Widget



URL Widget



URL Widget 2

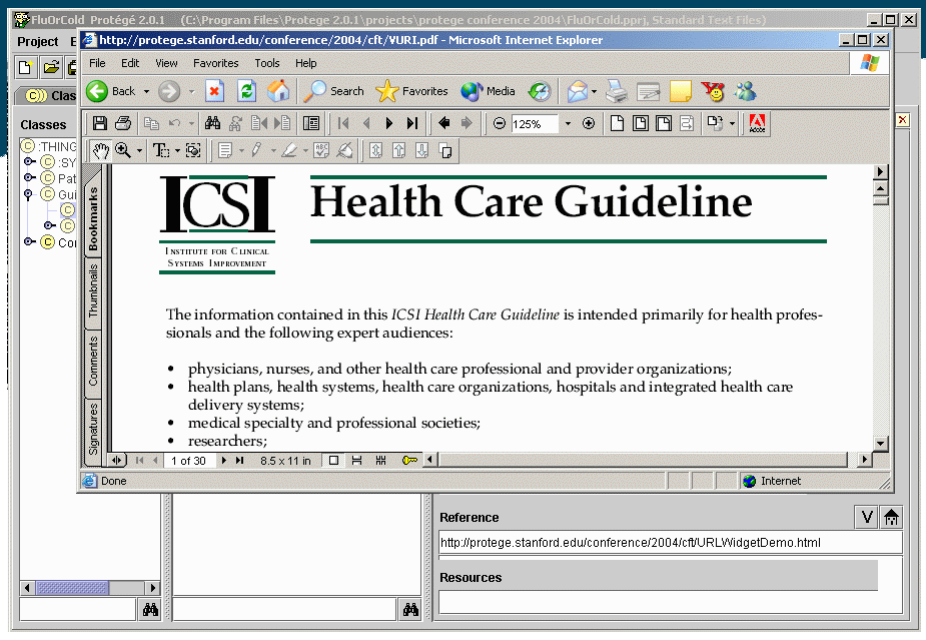
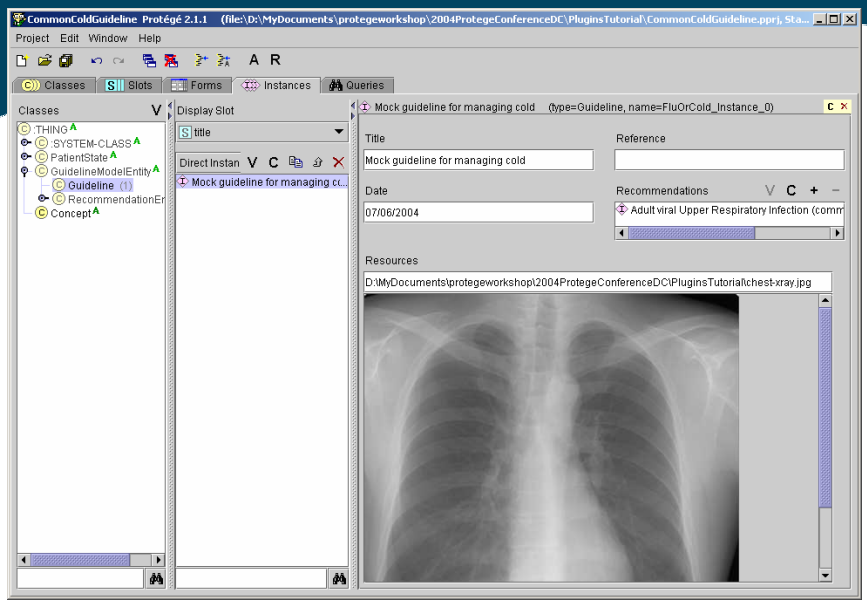


Image Widget



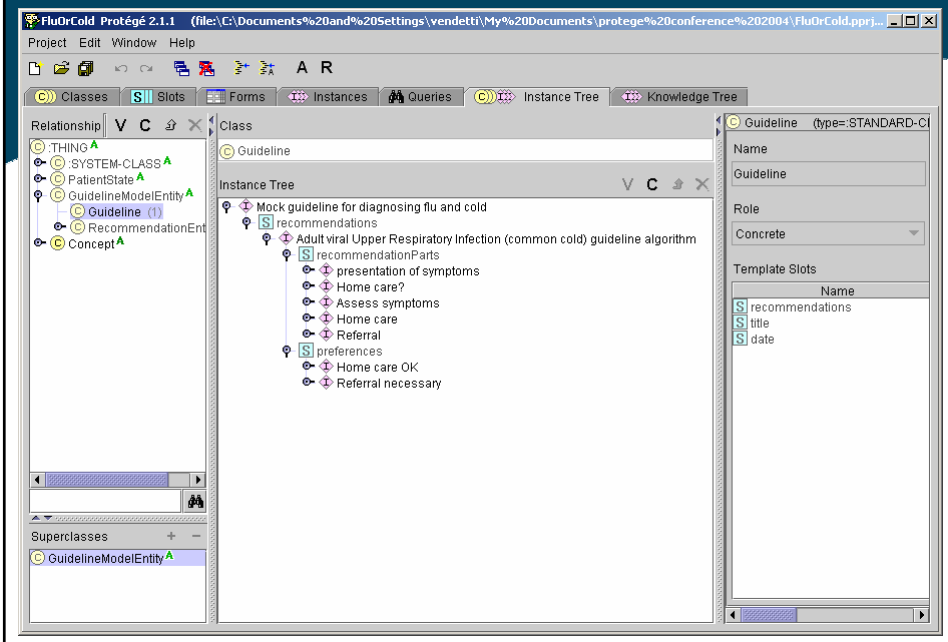
29

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

30

Instance Tree & Knowledge Tree Tabs Demo



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)

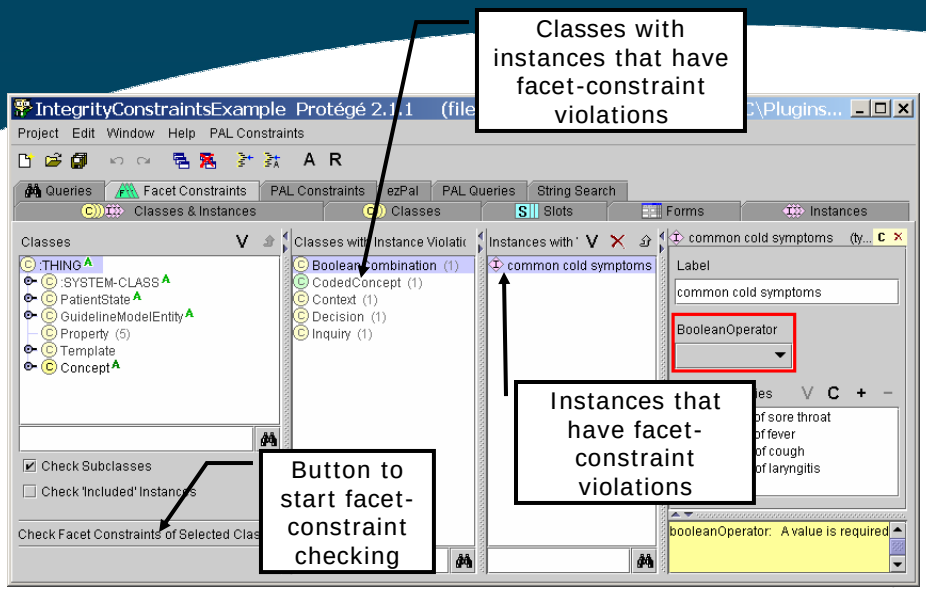
32

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

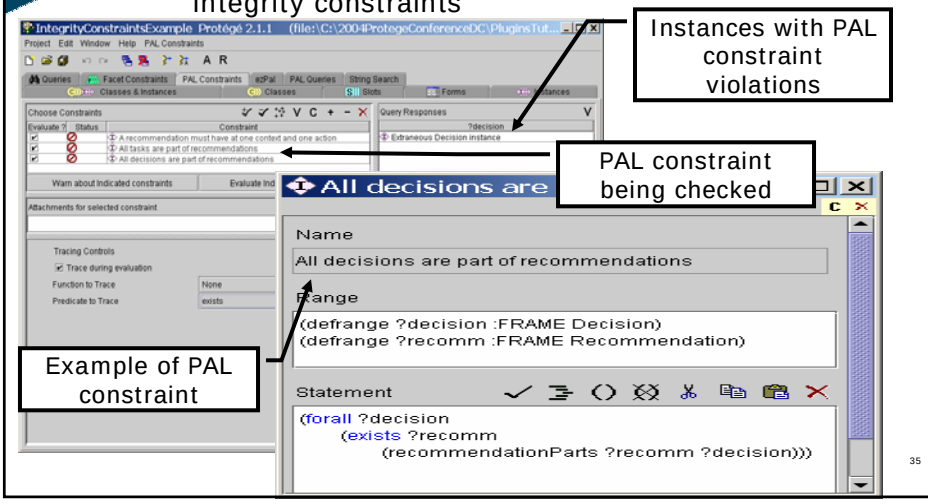
33

Facet-constraint tab



PAL-constraint tab

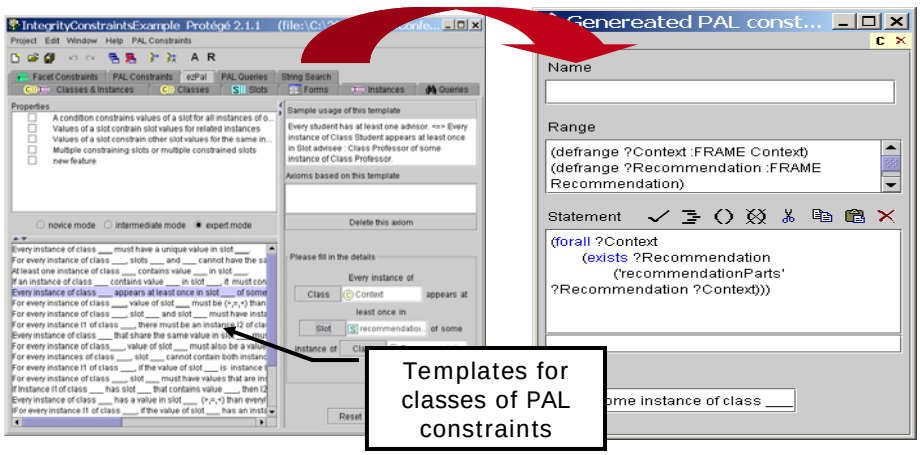
- Allows specification and checking of complex integrity constraints



35

EZPAL tab

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

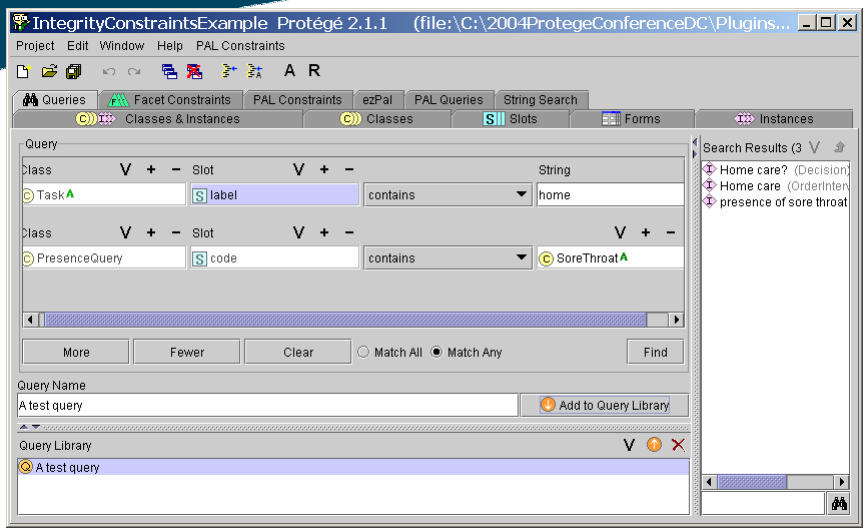


Querying knowledge bases

- **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

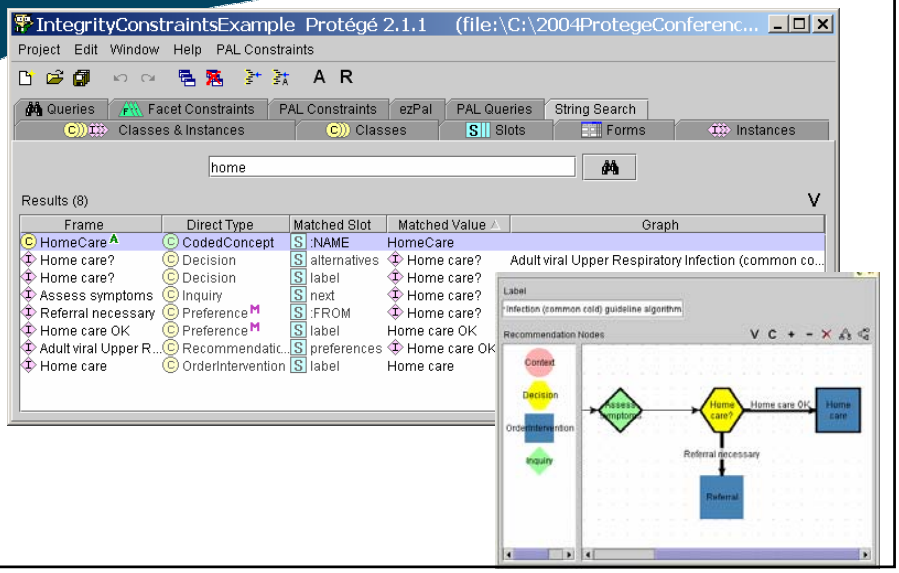
37

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



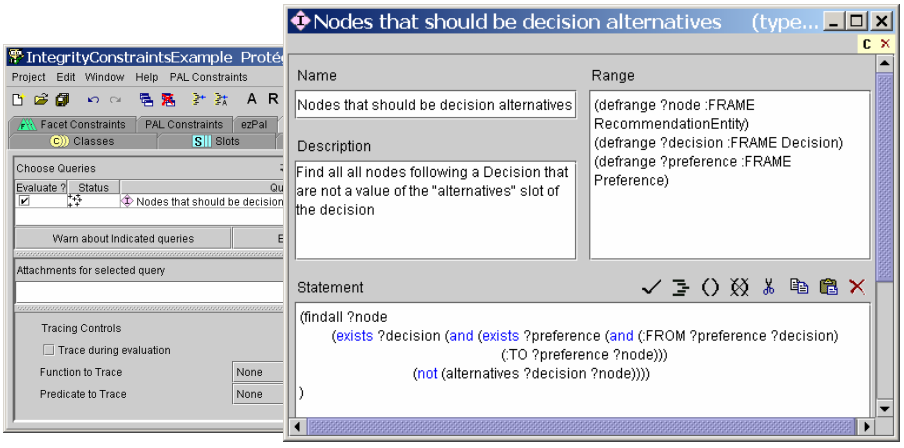
38

String search: search for a string anywhere in the project



PAL Query

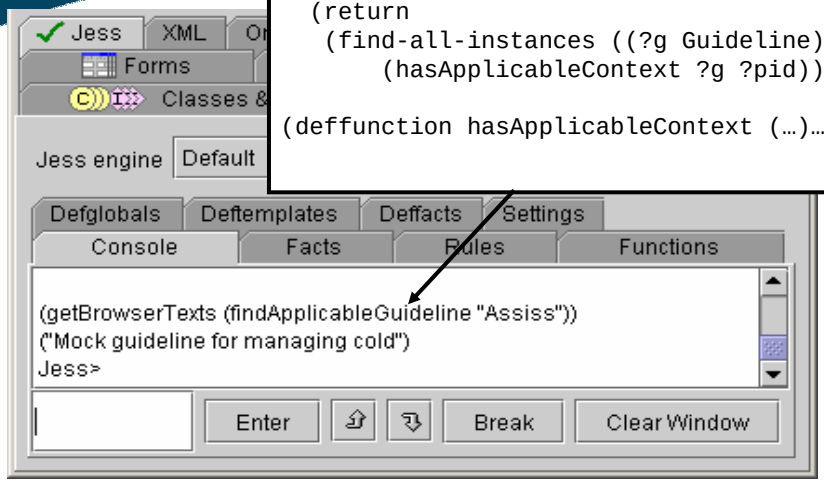
- Allows search with complex search criteria



JessTab as a query interface

```
(deffunction findApplicableGuideline(?pid)
  (return
   (find-all-instances ((?g Guideline))
       (hasApplicableContext ?g ?pid))))

(deffunction hasApplicableContext (…)…
```



41

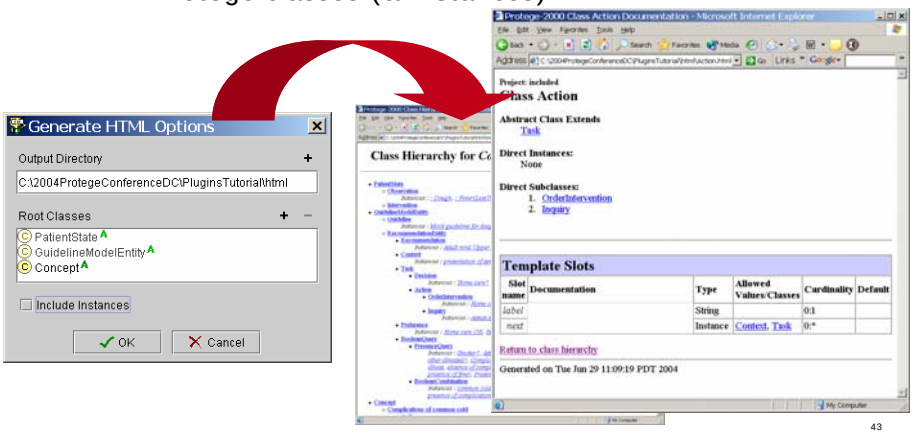
Exporting and publishing and knowledge bases

- 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 instanes
  - **XMLtab** – XML documents for classes and/or instances
  - **JessTab / 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

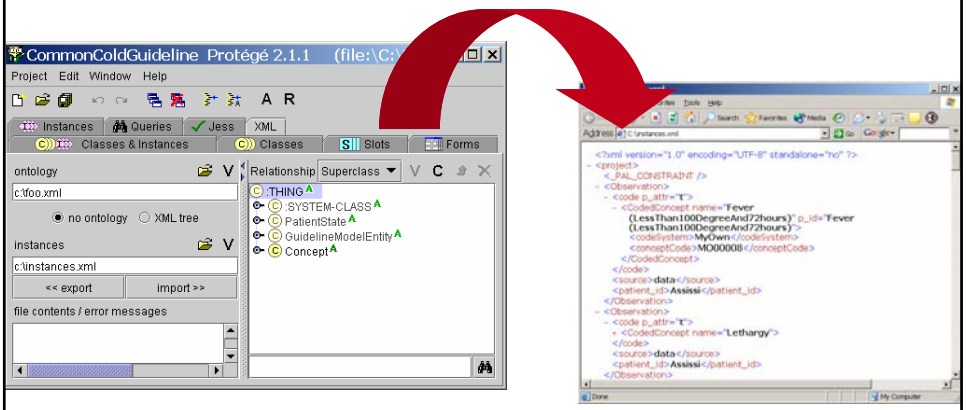
- Creates “Javadoc-style” HTML pages for Protégé classes (& instances)



43

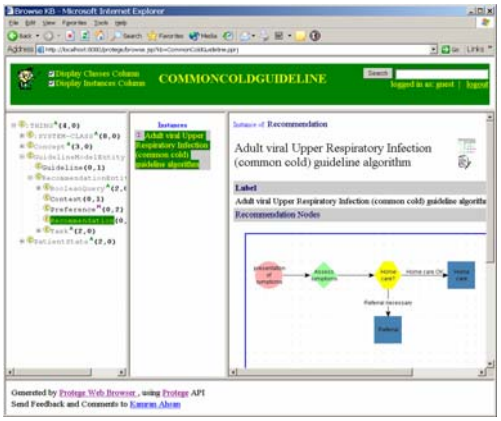
XML tab as exporter

- 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

JessTab: 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)
         (code ?code)
         (patient_id ?pid))
?super<-(object(is-a CodedConcept))
  (test (superclassp ?super ?code))
=>
  (make-instance of Observation
    (code ?super)
    (patient_id ?pid)
    (source guideline)
))
```

protégé class

protégé slots

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)

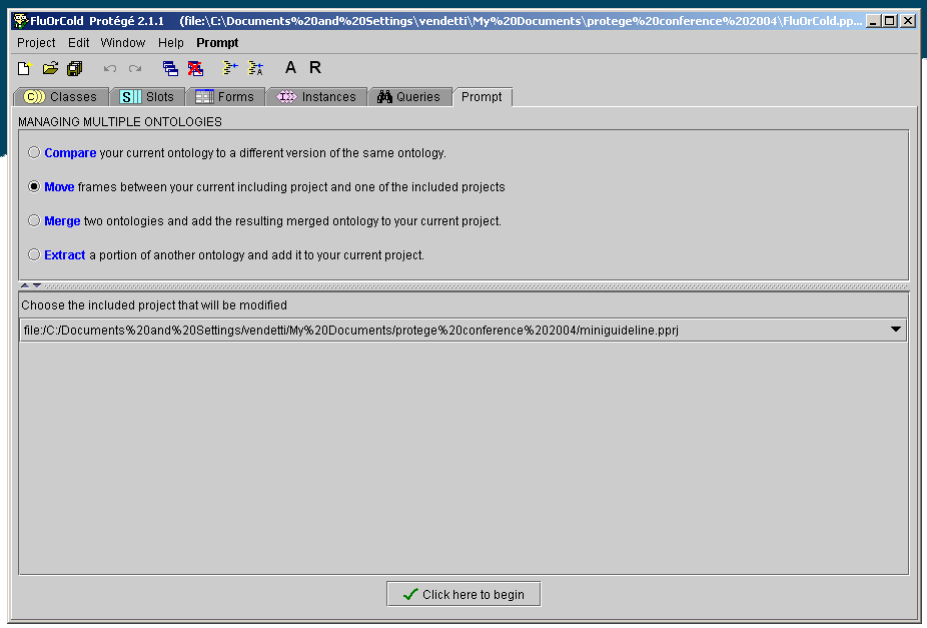
50

Managing Projects

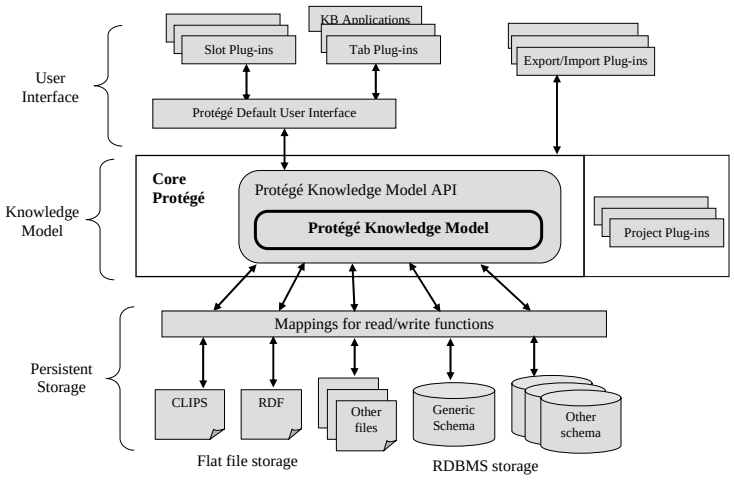
- Project Browser – manage libraries of Protégé projects
- Prompt
  - Move classes/instances up and down inclusion lattices
  - Extract portions of ontologies
  - compare versions of the same ontology

51

Prompt Tab Demo



Summary


User Interface
Knowledge Model
Persistent Storage
Slot Plug-ins
KB Applications
Tab Plug-ins
Export/Import Plug-ins
Protégé Default User Interface
Core Protégé
Protégé Knowledge Model API
Protégé Knowledge Model
Project Plug-ins
Mappings for read/write functions
CLIPS
RDF
Other files
Generic Schema
Other schema
Flat file storage
RDBMS storage

53