Managing Ontology Life Cycle: Part I

Jennifer Vendetti
Stanford Medical Informatics
Stanford University

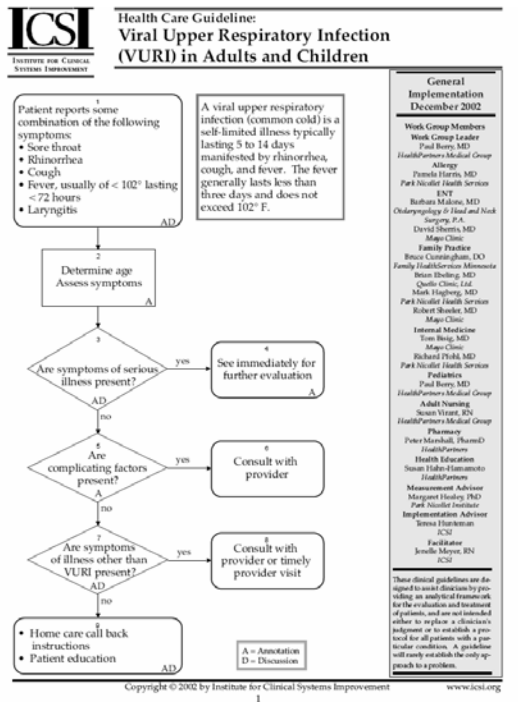
8th International Protégé Conference
Madrid, Spain, July 2005

Tasks in the life cycle of ontology and knowledge base development

- How to reuse or import existing resources?
- How to visualize information in the knowledge base?
- How to manage multiple ontologies using Prompt
- How to query or search knowledge bases?
- How to set up Protégé for multiple users
- How to export to external formats?
- How to add and test integrity constraints?

Example Scenario

- Goal: develop a medical decision-support application that generates recommendations based on clinical practice guideline
- Guideline example: management of common cold
- Disclaimer: tutorial example, no real medicine involved



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
3
Are symptoms of serious illness present?
AD
yes
4
See immediately for further evaluation
A
no
5
Are complicating factors present?
A
yes
6
Consult with provider
no
7
Are symptoms of illness other than VURI present?
AD
yes
8
Consult with provider or timely provider visit
no
9
• Home care call back instructions
• Patient education
AD
A = Annotation
D = Discussion

ICSI
Institute for Clinical Systems Improvement

Health Care Guideline:
Viral Upper Respiratory Infection (VURI) in Adults and Children

A viral upper respiratory infection (common cold) is a self-limited illness typically lasting 5 to 14 days manifested by rhinorrhea, cough, and fever. The fever generally lasts less than three days and does not exceed 102° F.

General Implementation December 2002

Work Group Members
Work Group Leader
Paul Berry, MD
HealthPartners Medical Group
Allergy
Pamela Harris, MD
Park Nicollet Health Services
ENT
Barbara Malone, MD
Otolaryngology & Head and Neck Surgery, P.A.
David Sherris, MD
Mayo Clinic
Family Practice
Bruce Cunningham, DO
Family HealthServices Minnesota
Brian Ebeling, MD
Quello Clinic, Ltd.
Mark Hagberg, MD
Park Nicollet Health Services
Robert Shoeler, MD
Mayo Clinic
Internal Medicine
Tom Bisig, MD
Mayo Clinic
Richard Pfohl, MD
Park Nicollet Health Services
Pediatrics
Paul Berry, MD
HealthPartners Medical Group
Adult Nursing
Susan Virant, RN
HealthPartners Medical Group
Pharmacy
Peter Marshall, PharmD
HealthPartners
Health Education
Susan Hahn-Hamamoto
HealthPartners
Measurement Advisor
Margaret Healey, PhD
Park Nicollet Institute
Implementation Advisor
Teresa Hunteman
ICSI
Facilitator
Jenelle Meyer, RN
ICSI

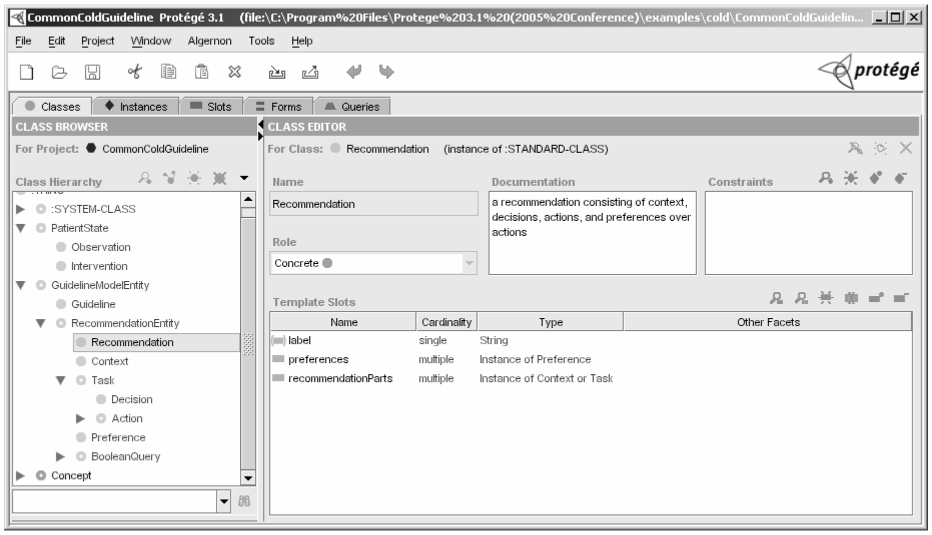
These clinical guidelines are designed to assist clinicians by providing an analytical framework for the evaluation and treatment of patients, and are not intended either to replace a clinician's judgment or to establish a protocol for all patients with a particular condition. A guideline will rarely establish the only approach to a problem.

Copyright © 2002 by Institute for Clinical Systems Improvement www.icsi.org
1

Example Domain

- A **guideline** is a set of **recommendations** consisting of
  - **contexts** (e.g. presentation of symptoms)
  - **tasks**
    - **actions** (e.g. inquiry, 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, etc.)
- **Concepts** represent abstractions of medical conditions (e.g. cough, fever, laryngitis)

Example ontology in Protégé



CommonColdGuideline Protégé 3.1 (file:\C:\Program%20Files\Protege%203.1%20(2005%20Conference)\examples\cold\CommonColdGuidelin...

For Project: CommonColdGuideline

Class Hierarchy

```
▶ :SYSTEM-CLASS
▼ PatientState
    Observation
    Intervention
▼ GuidelineModelEntity
    Guideline
  ▼ RecommendationEntity
      Recommendation
      Context
    ▼ Task
        Decision
      ▶ Action
      Preference
    ▶ BooleanQuery
▶ Concept
```

For Class: Recommendation (instance of :STANDARD-CLASS)

Name: Recommendation

Documentation: a recommendation consisting of context, decisions, actions, and preferences over actions

Constraints

Role: Concrete

Template Slots

Name	Cardinality	Type	Other Facets
label	single	String	
preferences	multiple	Instance of Preference	
recommendationParts	multiple	Instance of Context or Task	

Tasks in the life cycle of ontology and knowledge base development

- How to reuse or import existing resources?
- How to visualize information in the knowledge base?
- How to manage multiple ontologies using Prompt
- How to query or search knowledge bases?
- How to set up Protégé for multiple users
- How to export to external formats?
- How to add and test integrity constraints?

Import & reuse of existing resources

- *Issue:* Get existing resources into Protégé
- *Solutions:* Multiple
- *Things to consider:*
  - What formats are my resources in?
  - Import some or all?
  - Represent or reference?

Example import scenarios

- Import resources already in Protégé-compatible formats (existing Protégé projects, OKBC, RDFS, OWL)
- Import a database
- Import arbitrary XML files
- Import concepts from external servers

Import Protégé-compatible resources

Protégé Ontologies Library

- Download existing Protégé projects and use Protégé's inclusion mechanism
- Submissions welcome !!

OKBC Tab

- Import OKBC-compliant ontologies
- Downside: few exist, unsupported

Demo: DataGenie Tab

Import entire database

- Tables map to classes
- Columns map to slots

Downside: no export

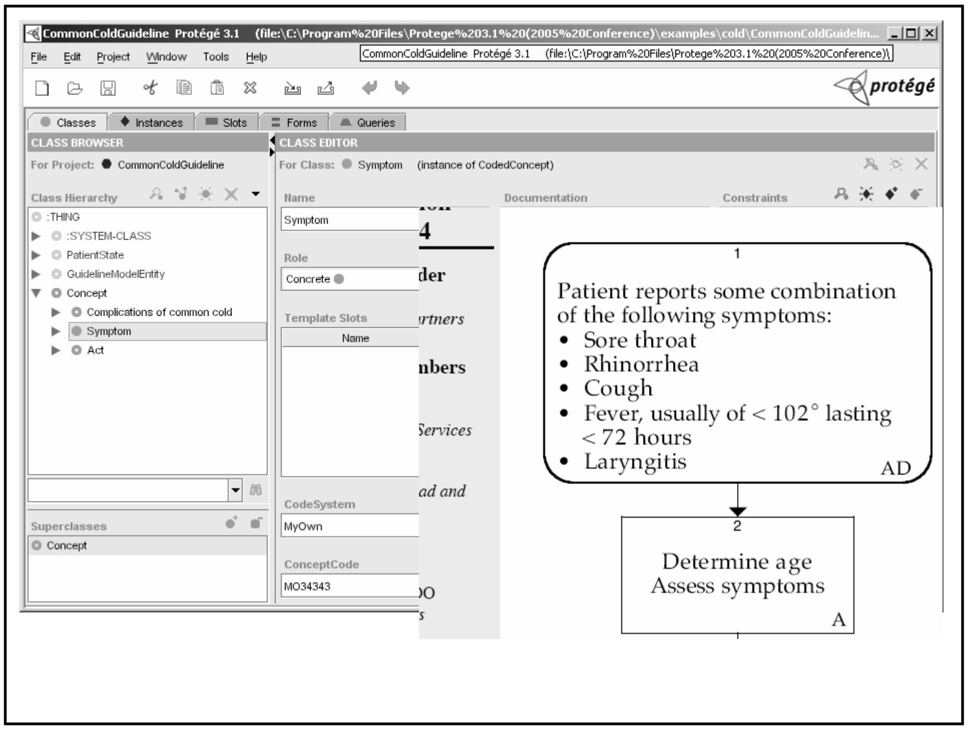
Demo: XML Tab

- Top level elements map to classes
- Contained elements map to slots

Downside: Unsupported, undocumented

UMLS Tab

- National Library of Medicine's Unified Medical Language System
- Biomedicine and health related concept databases (concepts, their names, their relationships)
- Set of software tools to access databases


CommonColdGuideline Protégé 3.1 (file:\C:\Program%20Files\Protege%203.1%20(2005%20Conference)\examples\cold\CommonColdGuidelin...

File Edit Project Window Tools Help
Classes Instances Slots Forms Queries
CLASS BROWSER
CLASS EDITOR

For Project: CommonColdGuideline

For Class: Symptom (instance of CodedConcept)

Class Hierarchy

- :THING
  - :SYSTEM-CLASS
  - PatientState
  - GuidelineModelEntity
  - Concept
    - Complications of common cold
    - Symptom
    - Act

Superclasses: Concept

Name: Symptom

Role: Concrete

Template Slots

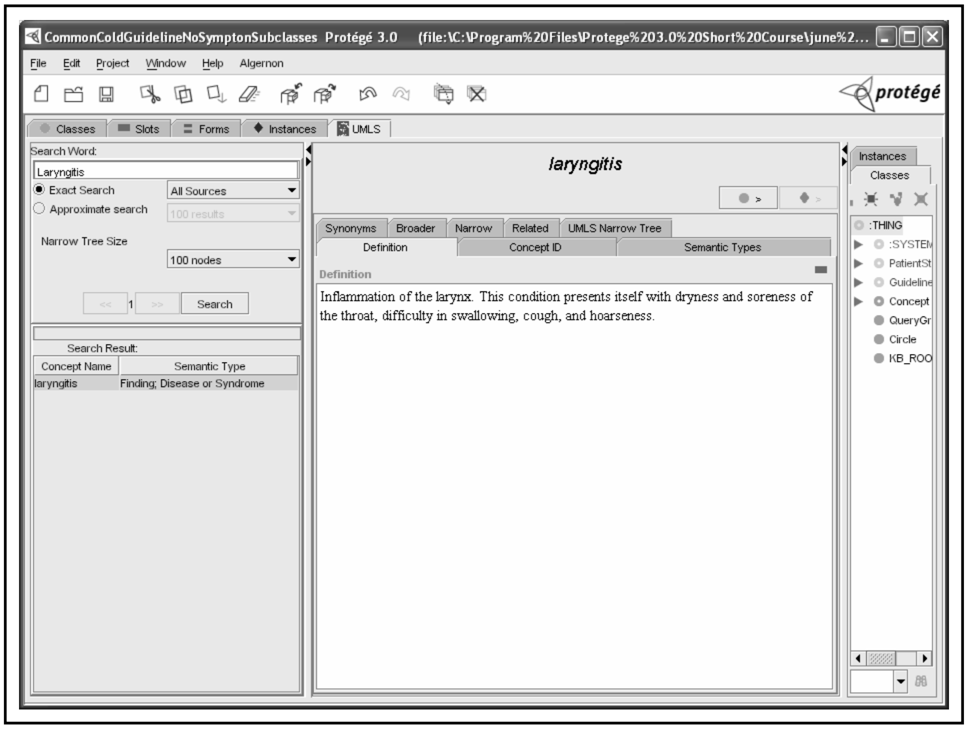
CodeSystem: MyOwn

ConceptCode: MO34343

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


CommonColdGuidelineNoSymptonSubclasses Protégé 3.0 (file:\C:\Program%20Files\Protege%203.0%20Short%20Course\june%2...

File Edit Project Window Help Algernon
Classes Slots Forms Instances UMLS
Synonyms Broader Narrow Related UMLS Narrow Tree
Definition Concept ID Semantic Types
Search

Search Word: Laryngitis

Exact Search: All Sources

Approximate search: 100 results

Narrow Tree Size: 100 nodes

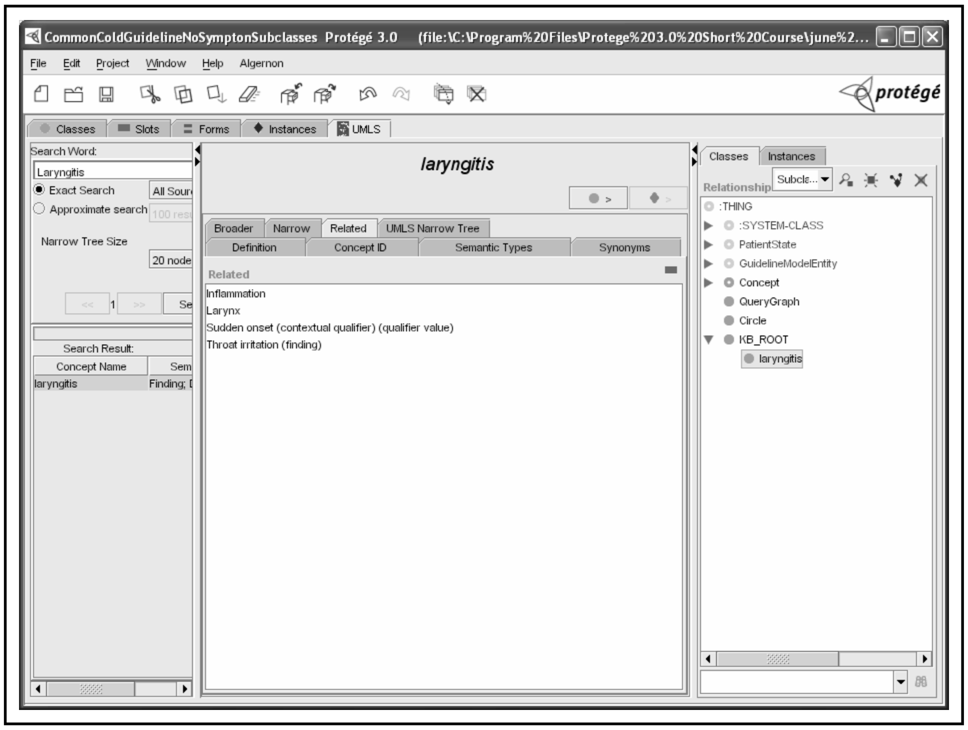
Search Result:

Concept Name	Semantic Type
laryngitis	Finding; Disease or Syndrome

laryngitis

Definition

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


CommonColdGuidelineNoSymptonSubclasses Protégé 3.0 (file:\C:\Program%20Files\Protege%203.0%20Short%20Course\june%2...
File Edit Project Window Help Algernon
protégé
Classes Slots Forms Instances UMLS
Broader Narrow Related UMLS Narrow Tree
Definition Concept ID Semantic Types Synonyms
Classes Instances

Search Word: Laryngitis

● Exact Search All Sour

○ Approximate search

Narrow Tree Size: 20 node

Search Result:

Concept Name	Sem
laryngitis	Finding; I

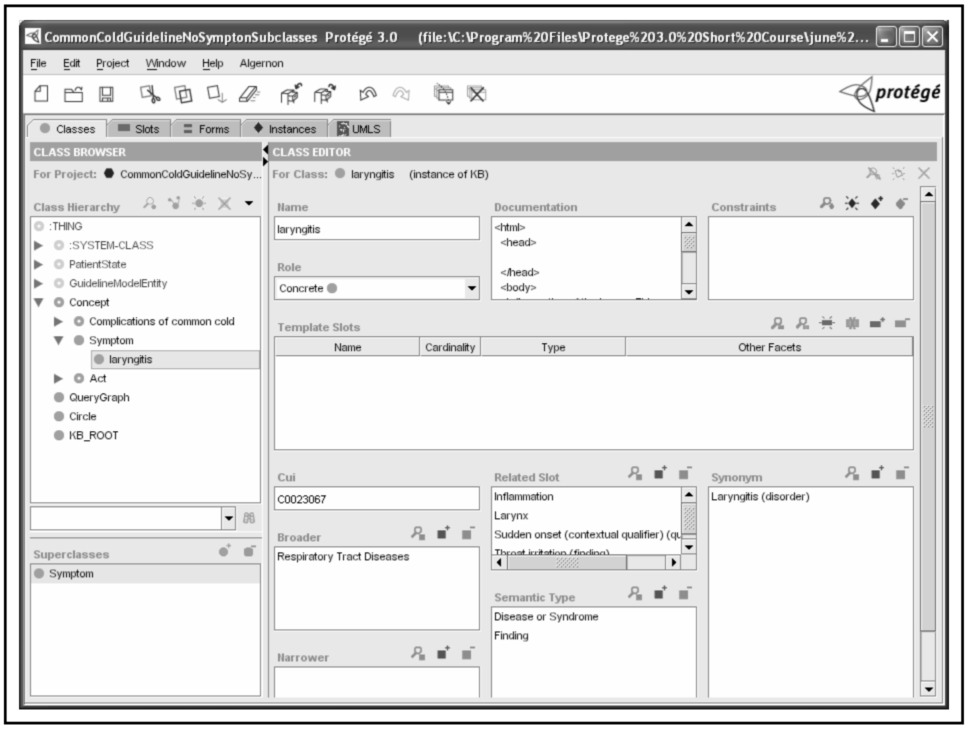
laryngitis

Related

Inflammation
Larynx
Sudden onset (contextual qualifier) (qualifier value)
Throat irritation (finding)

Relationship: Subcla...

- :THING
  - :SYSTEM-CLASS
  - PatientState
  - GuidelineModelEntity
  - Concept
  - QueryGraph
  - Circle
  - KB_ROOT
    - laryngitis


CommonColdGuidelineNoSymptonSubclasses Protégé 3.0 (file:\C:\Program%20Files\Protege%203.0%20Short%20Course\june%2...
File Edit Project Window Help Algernon
protégé
Classes Slots Forms Instances UMLS
CLASS BROWSER
CLASS EDITOR

For Project: CommonColdGuidelineNoSy...

Class Hierarchy

- :THING
  - :SYSTEM-CLASS
  - PatientState
  - GuidelineModelEntity
  - Concept
    - Complications of common cold
    - Symptom
      - laryngitis
    - Act
  - QueryGraph
  - Circle
  - KB_ROOT

Superclasses: Symptom

For Class: laryngitis (instance of KB)

Name: laryngitis

Role: Concrete

Documentation:
```
<html>
  <head>

  </head>
  <body>
```

Constraints

Template Slots

Name	Cardinality	Type	Other Facets

Cui: C0023067

Broader: Respiratory Tract Diseases

Narrower

Related Slot: Inflammation; Larynx; Sudden onset (contextual qualifier) (qu...; Throat irritation (finding)

Semantic Type: Disease or Syndrome; Finding

Synonym: Laryngitis (disorder)

Other import plug-ins

- OntoBase – read, navigate, update arbitrary databases
- WordNet Tab – import lexical content from WordNet
- Scripting Tabs (JessTab, Algernon, Protégé Script Console)
  - Scripting for… well… anything really (covered by Samson)
- Apelon DTS Plug-in
  - Commercial plug-in to browse/reference terminologies from Apelon’s Distributed Terminology Server (SNOMED CT, LOINC, etc.)

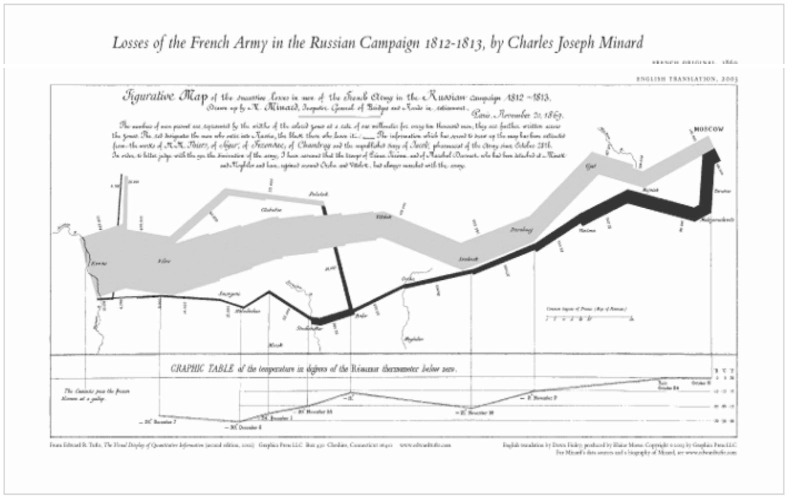
Visualization of knowledge bases

- What do we mean by visualization?
- What are the issues in visualizing ontologies?
- Large-scale visualization
- Visualization of non-standard data types
- Customization of instance display

Visualization of knowledge bases

- Visualization: graphically display data to facilitate better understanding of its meaning *(http://www.twocrows.com/glossary.htm)*
- Reference: E. Tufte, *Envisioning Information*, Graphics Press, 1990
- Principles of good design
  - Appropriateness for the information content
  - Increased number of displayed dimensions
  - Increased data density

Visualization of knowledge bases



From: http://www.edwardtufte.com/tufte/posters

Knowledge base visualization issues

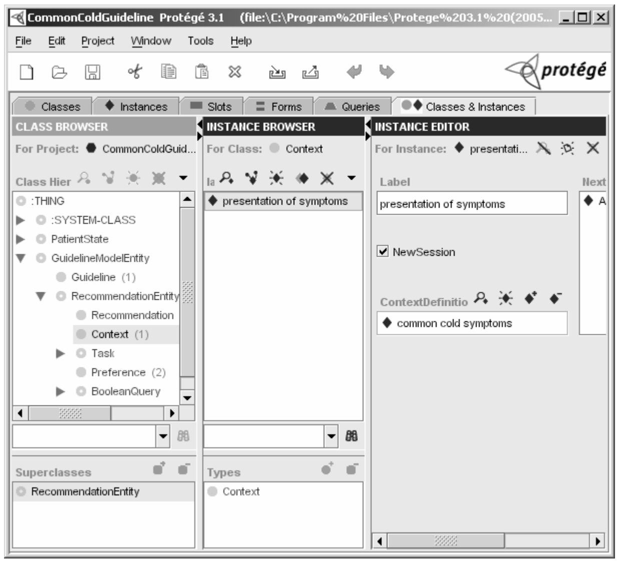
- What are the “meanings” that should be highlighted
- What are the alternative graphical displays for the “meanings”
- What are the appropriate navigation paths?

More specific display issues

- Selection of information
- Visual metaphors
- Scalability
- Domain specificity
- Possibility of multiple views
- Degree of user control

Protégé’s default visualization

- Tabs provides large-scale views
- Slot widgets provide default views of Protégé data types
- Default views highlight is-a (class/superclass), binary (slots), and instance-of (class/instance) relationships



For Project: CommonColdGuid...

For Class: Context

For Instance: presentati...

presentation of symptoms

Label: presentation of symptoms

☑ NewSession

ContextDefinitio: common cold symptoms

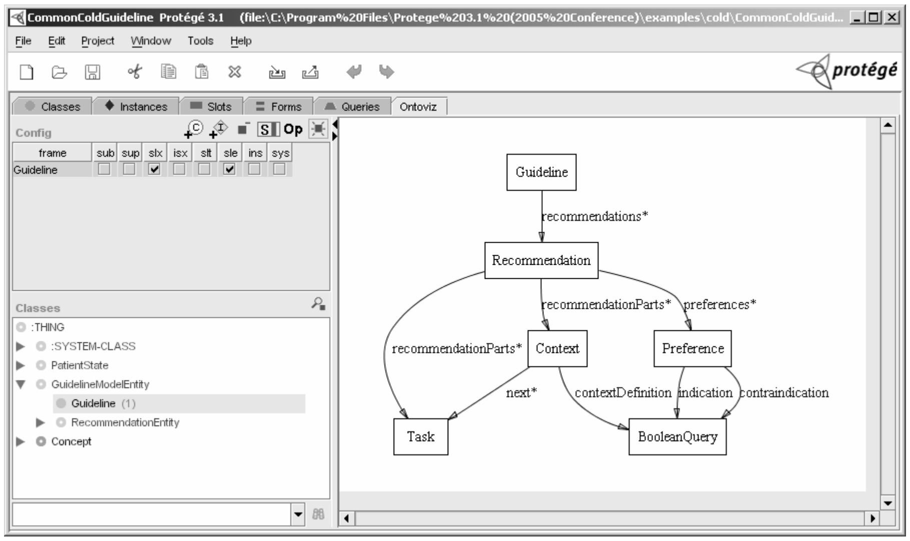
Superclasses: RecommendationEntity

Types: Context

Larger scale visualization

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

Demo: OntoViz Tab


CommonColdGuideline Protégé 3.1 (file:\C:\Program%20Files\Protege%203.1%20(2005%20Conference)\examples\cold\CommonColdGuid...

Config

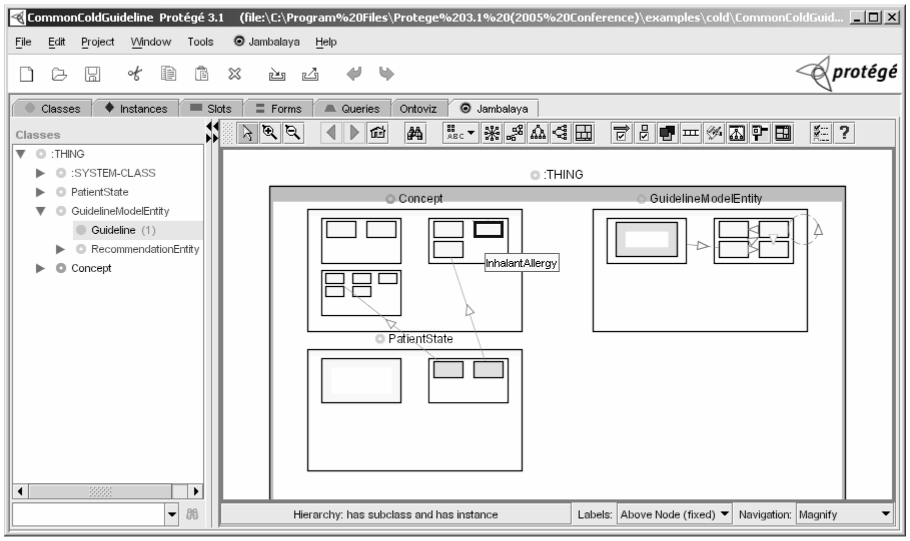
frame	sub	sup	slx	isx	slt	sle	ins	sys
Guideline	☐	☐	✓	☐	☐	✓	☐	☐

Classes
- :THING
  - :SYSTEM-CLASS
  - PatientState
  - GuidelineModelEntity
    - Guideline (1)
    - RecommendationEntity
  - Concept

Jambalaya

- Uses SHriMP (Simple Hierarchical Multi-Perspective)
- SHriMP is designed to help people browse complex information spaces
- Upside: very feature rich
- Downside: bigger learning curve than other tools
- Documentation/tutorials: http://www.thechiselgroup.org/jambalaya

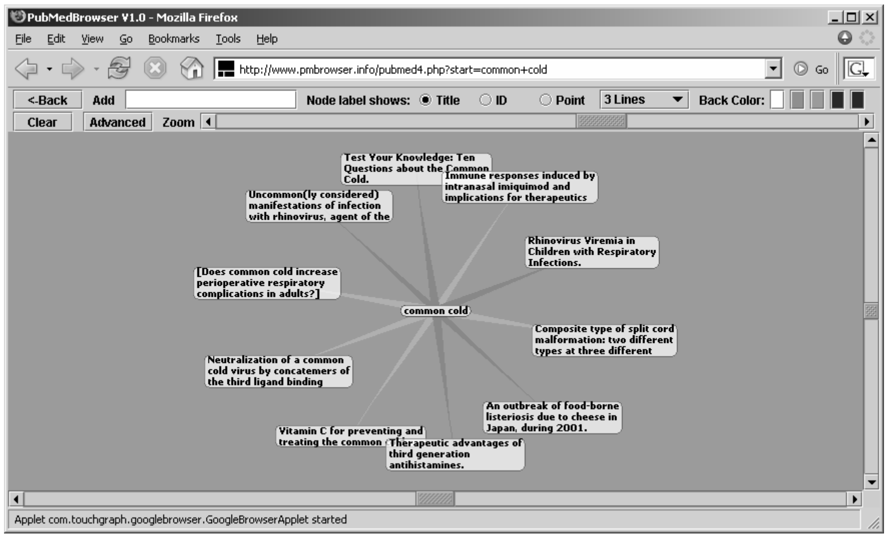
Jambalaya


CommonColdGuideline Protégé 3.1 (file:\C:\Program%20Files\Protege%203.1%20(2005%20Conference)\examples\cold\CommonColdGuid...
File Edit Project Window Tools Jambalaya Help
protégé
Classes Instances Slots Forms Queries Ontoviz Jambalaya
Classes
:THING
:SYSTEM-CLASS
PatientState
GuidelineModelEntity
Guideline (1)
RecommendationEntity
Concept
:THING
Concept
GuidelineModelEntity
InhalantAllergy
PatientState
Hierarchy: has subclass and has instance
Labels: Above Node (fixed)
Navigation: Magnify

TGViz Tab

- Utilizes TouchGraph (renders networks as interactive graphs)
- TouchGraph uses “Spring Layout”
- PubMed uses TouchGraph to visualize graphs of related documents in medical libraries

TGViz Tab


PubMedBrowser V1.0 - Mozilla Firefox

http://www.pmbrowser.info/pubmed4.php?start=common+cold

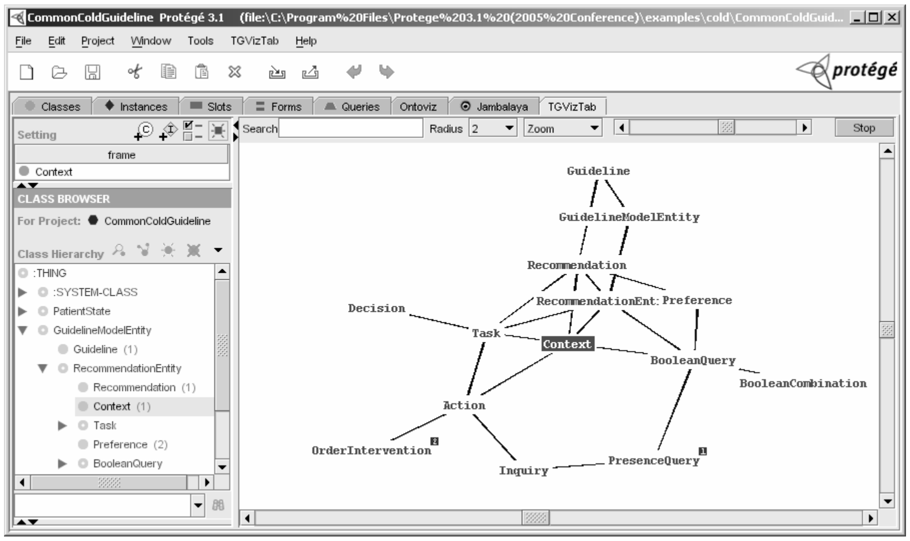
Node label shows: [✓ Title] ID Point

3 Lines

Applet com.touchgraph.googlebrowser.GoogleBrowserApplet started

File Edit View Go Bookmarks Tools Help
<-Back
Add
Back Color:
Clear
Advanced
Zoom
common cold
Test Your Knowledge: Ten Questions about the Common Cold.
Immune responses induced by intranasal imiquimod and implications for therapeutics
Uncommon(ly considered) manifestations of infection with rhinovirus, agent of the
Rhinovirus Viremia in Children with Respiratory Infections.
[Does common cold increase perioperative respiratory complications in adults?]
Composite type of split cord malformation: two different types at three different
Neutralization of a common cold virus by concatemers of the third ligand binding
An outbreak of food-borne listeriosis due to cheese in Japan, during 2001.
Vitamin C for preventing and treating the common
Therapeutic advantages of third generation antihistamines.

TGViz Tab


CommonColdGuideline Protégé 3.1 (file:\C:\Program%20Files\Protege%203.1%20(2005%20Conference)\examples\cold\CommonColdGuid...

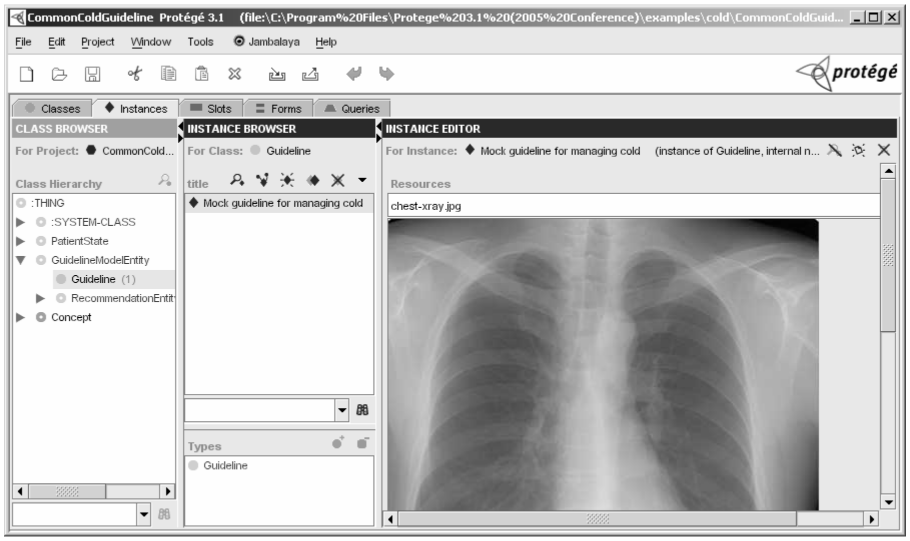
For Project: CommonColdGuideline

Radius 2

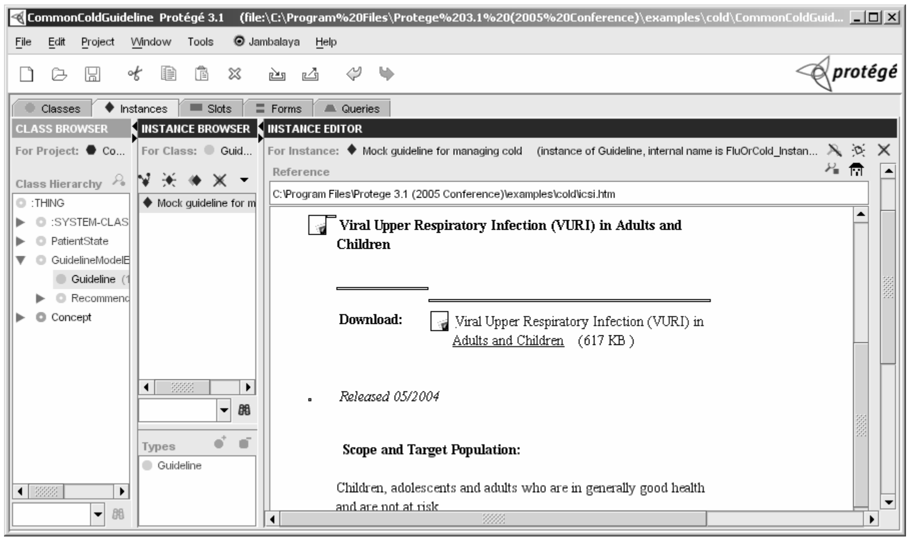
Visualization of non-standard data types

- Protégé’s basic data types
  - integer, float, string, symbol, class, instance
- Non-standard data types
  - Date
  - Image
  - URL
  - Ordered list
  - …

Image Widget



URL Widget


CommonColdGuideline Protégé 3.1 (file:\C:\Program%20Files\Protege%203.1%20(2005%20Conference)\examples\cold\CommonColdGuid...
File Edit Project Window Tools Jambalaya Help
protégé
Classes Instances Slots Forms Queries
CLASS BROWSER
INSTANCE BROWSER
INSTANCE EDITOR
Class Hierarchy
:THING
:SYSTEM-CLAS
PatientState
GuidelineModelE
Guideline
Recommenc
Concept
Types
Guideline

For Project: Co...

For Class: Guid...

Mock guideline for m

For Instance: Mock guideline for managing cold (instance of Guideline, internal name is FluOrCold_Instan...

Reference

C:\Program Files\Protege 3.1 (2005 Conference)\examples\cold\icsi.htm

Viral Upper Respiratory Infection (VURI) in Adults and Children

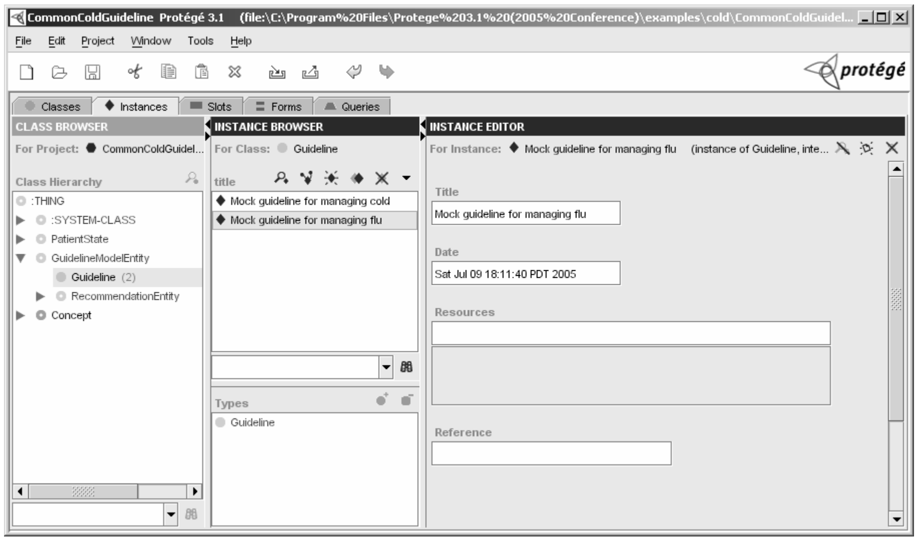
Download: Viral Upper Respiratory Infection (VURI) in Adults and Children (617 KB)

Released 05/2004

Scope and Target Population:

Children, adolescents and adults who are in generally good health and are not at risk

Creation Date Widget


CommonColdGuideline Protégé 3.1 (file:\C:\Program%20Files\Protege%203.1%20(2005%20Conference)\examples\cold\CommonColdGuidel...
File Edit Project Window Tools Help
protégé
Classes Instances Slots Forms Queries
CLASS BROWSER
INSTANCE BROWSER
INSTANCE EDITOR
Class Hierarchy
:THING
:SYSTEM-CLASS
PatientState
GuidelineModelEntity
Guideline (2)
RecommendationEntity
Concept
Types
Guideline

For Project: CommonColdGuidel...

For Class: Guideline

title

Mock guideline for managing cold

Mock guideline for managing flu

For Instance: Mock guideline for managing flu (instance of Guideline, inte...

Title

Mock guideline for managing flu

Date

Sat Jul 09 18:11:40 PDT 2005

Resources

Reference

Customization of instance display

- Custom slot widgets are primary method
- Classes and their instance forms are fixed within a project
- Can override form customizations in including projects

Instance & Knowledge Tree Tabs

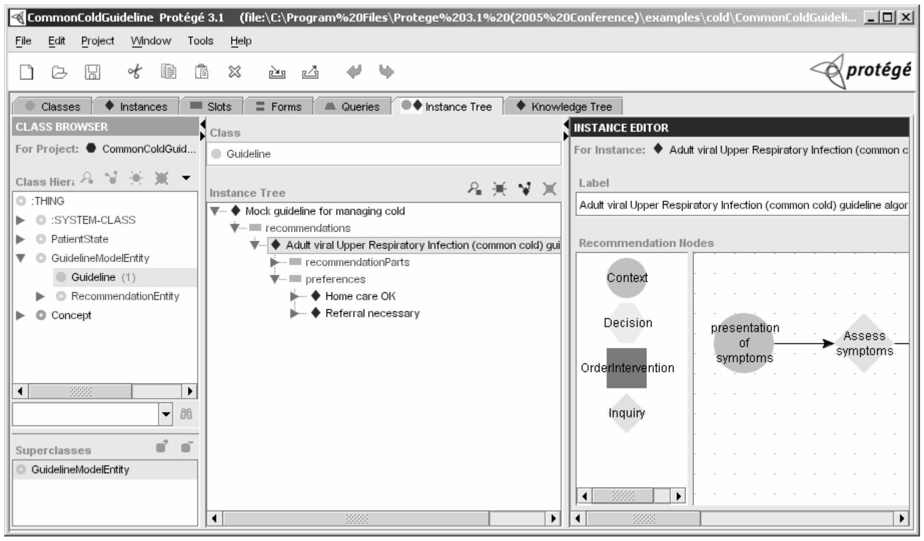
Instance Tree

- 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

InstanceTree Tab


CommonColdGuideline Protégé 3.1 (file:\C:\Program%20Files\Protege%203.1%20(2005%20Conference)\examples\cold\CommonColdGuideli...
File Edit Project Window Tools Help
protégé
Classes Instances Slots Forms Queries Instance Tree Knowledge Tree
CLASS BROWSER
Class
INSTANCE EDITOR
Class Hierarchy
Instance Tree
Superclasses
Label
Recommendation Nodes
Context
Decision
OrderIntervention
Inquiry
presentation of symptoms
Assess symptoms

For Project: CommonColdGuid...

Class: Guideline

For Instance: Adult viral Upper Respiratory Infection (common c

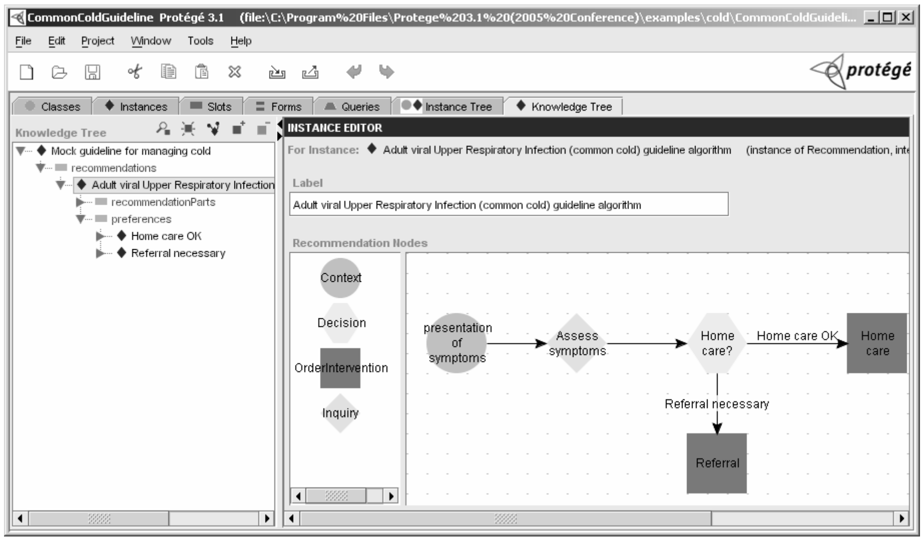
- :THING
  - :SYSTEM-CLASS
  - PatientState
  - GuidelineModelEntity
    - Guideline (1)
    - RecommendationEntity
  - Concept

Superclasses: GuidelineModelEntity

- Mock guideline for managing cold
  - recommendations
    - Adult viral Upper Respiratory Infection (common cold) gui
      - recommendationParts
      - preferences
        - Home care OK
        - Referral necessary

Label: Adult viral Upper Respiratory Infection (common cold) guideline algor

Knowledge Tree Tab


CommonColdGuideline Protégé 3.1 (file:\C:\Program%20Files\Protege%203.1%20(2005%20Conference)\examples\cold\CommonColdGuideli...
File Edit Project Window Tools Help
protégé
Classes Instances Slots Forms Queries Instance Tree Knowledge Tree
Knowledge Tree
INSTANCE EDITOR
Label
Recommendation Nodes
Context
Decision
OrderIntervention
Inquiry
presentation of symptoms
Assess symptoms
Home care?
Home care OK
Home care
Referral necessary
Referral

- Mock guideline for managing cold
  - recommendations
    - Adult viral Upper Respiratory Infection
      - recommendationParts
      - preferences
        - Home care OK
        - Referral necessary

For Instance: Adult viral Upper Respiratory Infection (common cold) guideline algorithm (instance of Recommendation, inte

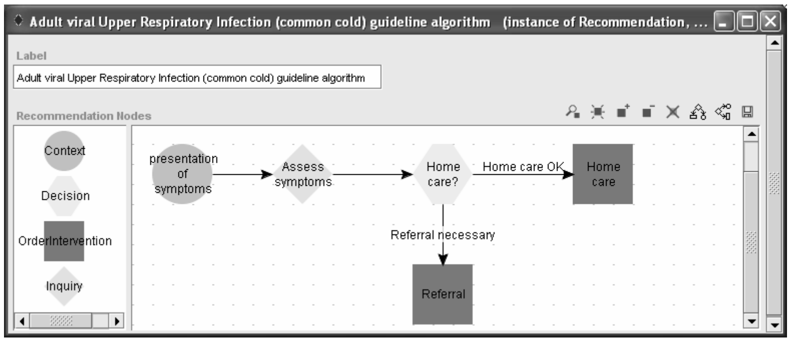
Label: Adult viral Upper Respiratory Infection (common cold) guideline algorithm

Graph Widget

- Focus on one example of a custom slot widget - the graph widget
- Detailed tutorial on our Web site: http://protege.stanford.edu/doc/tutorial/graph_widget/

What is the Graph Widget?

- Allows visual editing of of instances and relationships between instances
- Alternative to Protege's "Forms" for entering instance data


Adult viral Upper Respiratory Infection (common cold) guideline algorithm (instance of Recommendation, ...

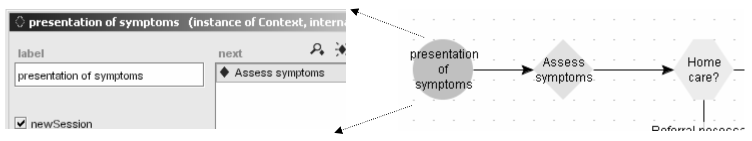
Label

Adult viral Upper Respiratory Infection (common cold) guideline algorithm

Recommendation Nodes

When is the graph widget appropriate?

- When instances of a slot are connected as values of some slots (e.g. a linked list where one instance is links to another through a slot relation)



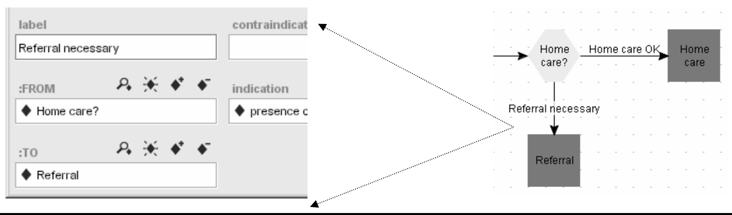
presentation of symptoms (instance of Context, intern

label: presentation of symptoms

next: ◆ Assess symptoms

☑ newSession

- When instances of a slot are related by instances of *Directed-Binary-Relation* class



label: Referral necessary

contraindicat

:FROM: ◆ Home care?

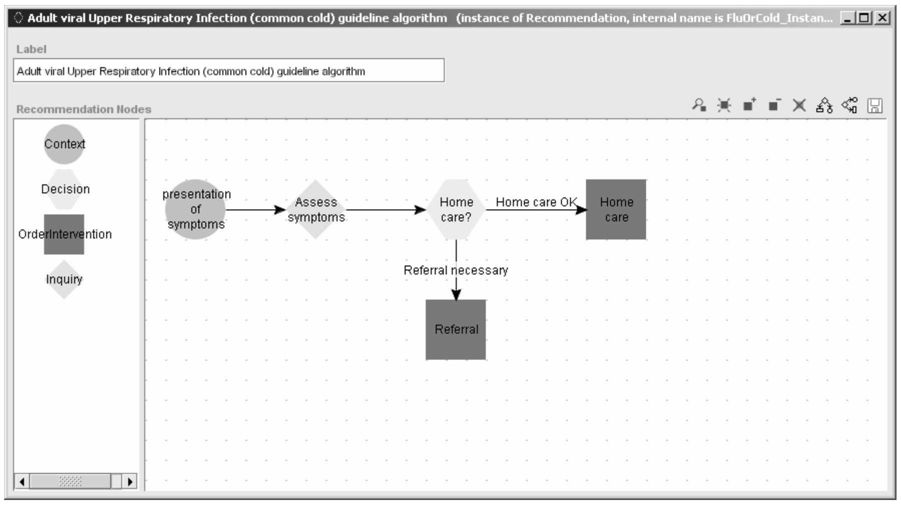
indication: ◆ presence c

:TO: ◆ Referral

When is the graph widget appropriate?

- Speeds knowledge acquisition in ontologies with heavily interconnected concepts.
- Helps convey meaning and organization of acquired knowledge
- Data that resembles process diagrams, flow charts, organizational charts

Graph Widget


Adult viral Upper Respiratory Infection (common cold) guideline algorithm (instance of Recommendation, internal name is FluOrCold_Instan...

Label

Adult viral Upper Respiratory Infection (common cold) guideline algorithm

Recommendation Nodes

What are other custom slot widgets?

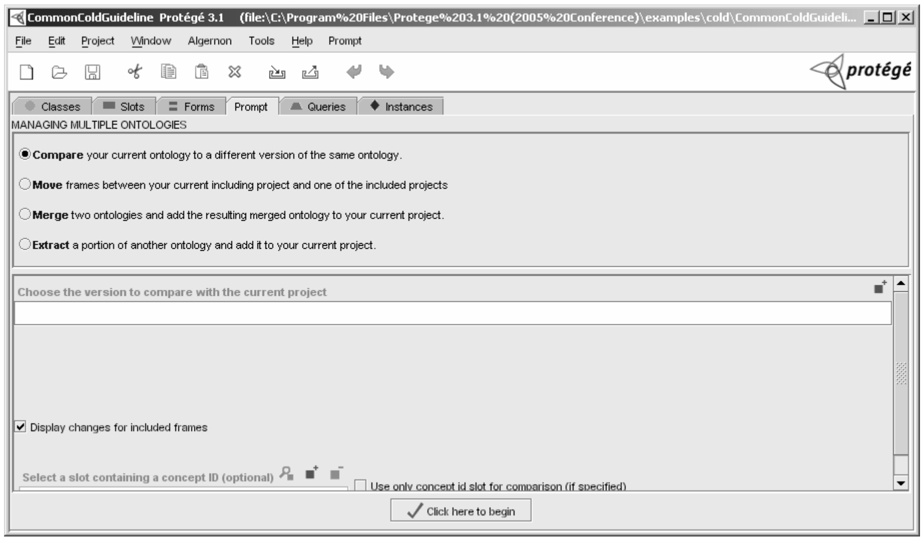
- ContainsWidget
  - Embeds forms for slots of type instance
- InstanceTableWidget
- InstanceListWidget

Managing multiple ontologies

What is Prompt?

- Compare versions of the same ontology (version management)
- Move frames between included and including project
- Merge two ontologies into one
- Extract a part of an ontology

Demo: Prompt


CommonColdGuideline Protégé 3.1 (file:\C:\Program%20Files\Protege%203.1%20(2005%20Conference)\examples\cold\CommonColdGuideli...

File Edit Project Window Algernon Tools Help Prompt
protégé
Classes Slots Forms Prompt Queries Instances

MANAGING MULTIPLE ONTOLOGIES

[✓ **Compare** your current ontology to a different version of the same ontology.]

Move frames between your current including project and one of the included projects

Merge two ontologies and add the resulting merged ontology to your current project.

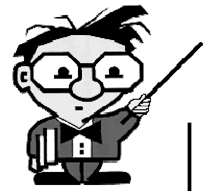
Extract a portion of another ontology and add it to your current project.

Choose the version to compare with the current project

[✓] Display changes for included frames

Select a slot containing a concept ID (optional)

[ ] Use only concept id slot for comparison (if specified)



Managing Ontology Life Cycle: Part II

Samson Tu
Stanford Medical Informatics
Stanford University

8th International Protégé Conference
Madrid, Spain, July 2005

Tasks in the life cycle of ontology and knowledge base development

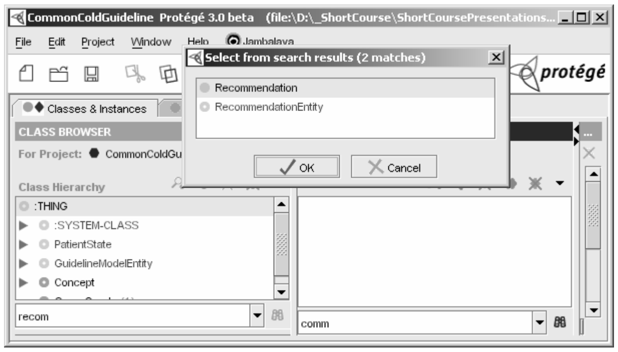
- How to reuse or import existing resources?
- How to visualize information in the knowledge base?
- How to manage multiple ontologies using Prompt
- How to query or search knowledge bases?
- How to set up Protégé for multiple users
- How to export to external formats?
- How to add and test integrity constraints?

4. How to query or search knowledge bases?

- Problem: you have specific information you are looking for in the knowledge base
- Considerations
  - What is the search space?
  - What identifying information do you have?
  - What do you want to do with search result?
  - What is the efficiency of search?

Scenario: Given the name or display name, find the frame in Protégé browsers

- Example: Search for a class whose name begins with "recom"
- Solution: Use Protégé "binocular" search


CommonColdGuideline Protégé 3.0 beta (file:\D:_ShortCourse\ShortCoursePresentations...

Select from search results (2 matches)

Recommendation
RecommendationEntity

For Project: CommonColdGu

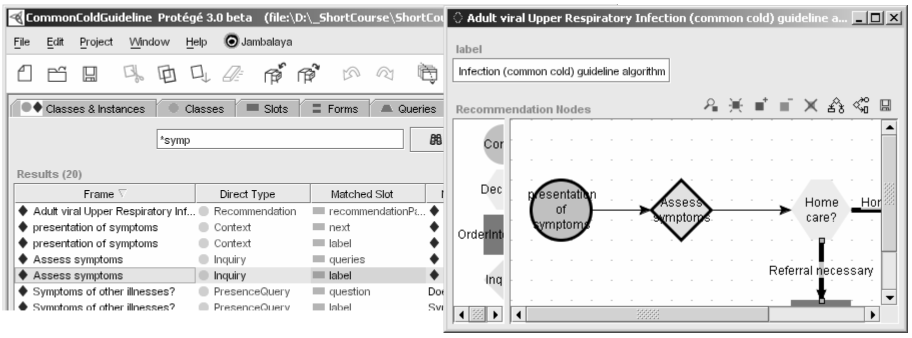
:THING
:SYSTEM-CLASS
PatientState
GuidelineModelEntity
Concept

recom

comm

Scenario: Given a string, search its occurrences in the KB

- Example: Search occurrences of the string “symp”
- Solution: Use the StringSearch tab



CommonColdGuideline Protégé 3.0 beta (file:\D:_ShortCourse\ShortCou

Search string: *symp

Results (20)

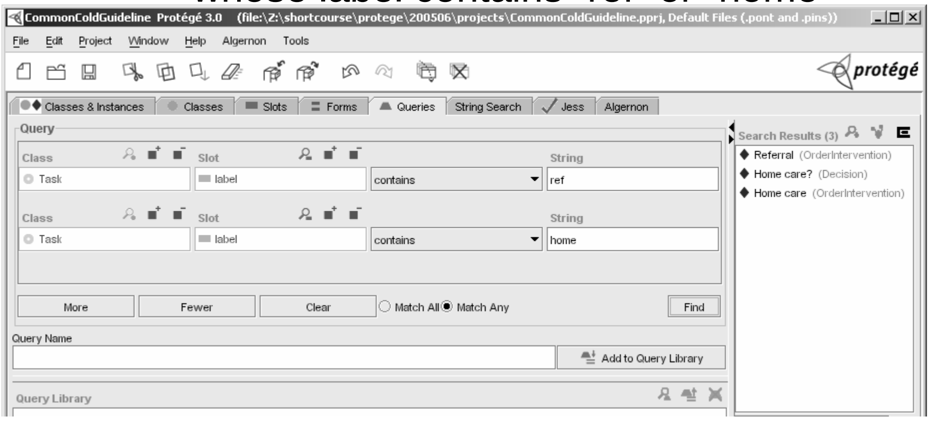
Frame	Direct Type	Matched Slot	
Adult viral Upper Respiratory Inf...	Recommendation	recommendationP...	◆
presentation of symptoms	Context	next	◆
presentation of symptoms	Context	label	◆
Assess symptoms	Inquiry	queries	◆
Assess symptoms	Inquiry	label	◆
Symptoms of other illnesses?	PresenceQuery	question	Do
Symptoms of other illnesses?	PresenceQuery	label	Sy

Adult viral Upper Respiratory Infection (common cold) guideline a...

label: Infection (common cold) guideline algorithm

Scenario: Search for instances of classes whose slot values satisfy simple constraints

- Example: Search instances of Task whose label contains “ref” or “home”



CommonColdGuideline Protégé 3.0 (file:\Z:\shortcourse\protege\200506\projects\CommonColdGuideline.pprj, Default Files (.pont and .pins))

Query

Class	Slot		String
Task	label	contains	ref
Task	label	contains	home

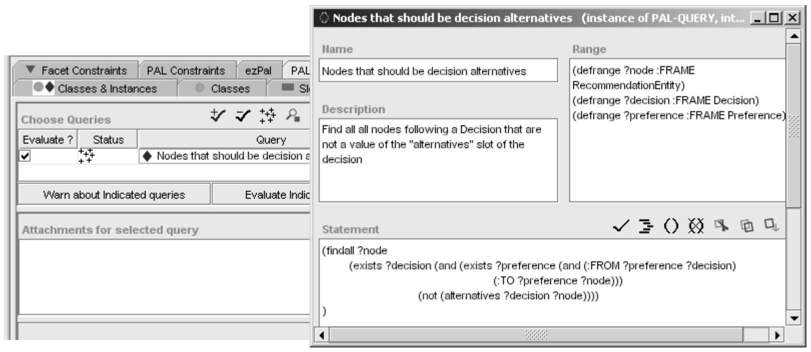
( ) Match All (●) Match Any

Search Results (3)

- Referral (OrderIntervention)
- Home care? (Decision)
- Home care (OrderIntervention)

Scenario: Find instances of a class that satisfy constraints involving other instances

- Example: Find nodes A in a directed graph such that A follows a decision and A is not the value of the decision's "alternatives" slot
- Solution: Use PAL query



Nodes that should be decision alternatives (instance of PAL-QUERY, int...

Name: Nodes that should be decision alternatives

Range:
```
(defrange ?node :FRAME
RecommendationEntity)
(defrange ?decision :FRAME Decision)
(defrange ?preference :FRAME Preference)
```

Description: Find all all nodes following a Decision that are not a value of the "alternatives" slot of the decision

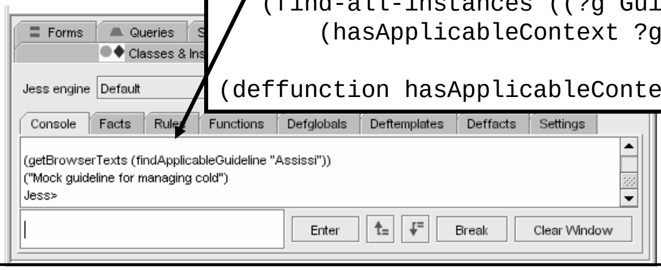
Statement:
```
(findall ?node
       (exists ?decision (and (exists ?preference (and (:FROM ?preference ?decision)
                                                       (:TO ?preference ?node)))
                         (not (alternatives ?decision ?node))))
)
```

Scenario: Programming to search in the KB (scripting languages)

- Problem: You need to perform search not covered in available GUI tools
- Solution 1: Use interface to scripting languages (Jess, Algernon, Python, etc.)

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

(deffunction hasApplicableContext (…)…
```



```
(getBrowserTexts (findApplicableGuideline "Assissi"))
("Mock guideline for managing cold")
Jess>
```

Scenario: Programming to search in the KB (search API)

- “High-level” Java API for different classes of searches
- Search based on “context” (KB, class tree, instance tree, …) and “conditions” (constraints on slot values)

Example: InstanceTree Search for Instances

InstanceTree: all instances that are referenced directly or indirectly from a given instance

In instanceTree “Mock guideline for managing cold”, search for instances that have browser text name “*home” and whose “code” slot has value class `HomeCare`. This search can be done at only 1 level, or recursively.

See Protégé search API documentation

Scenario: Programming to search in the KB (Protégé API)

- “match” methods in Protégé Java API *KnowledgeBase* interface
- Optimized for search database backend

Tasks in the life cycle of ontology and knowledge base development

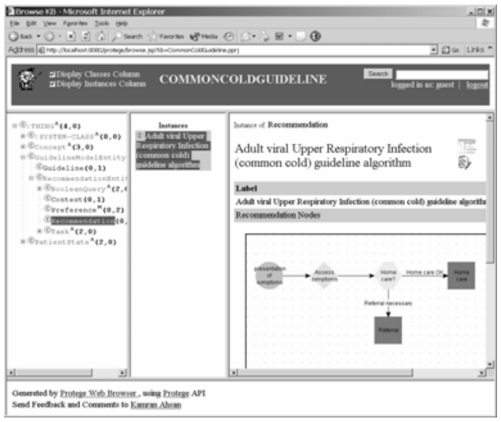
1. How to reuse or import existing resources?
2. How to visualize information in the knowledge base?
    1. How to customize display of instances?
    2. How to display non-standard data types?
    3. What options are there to display portions of knowledge bases?
    4. What options are available for navigation and browsing?
3. How to manage multiple ontologies using Prompt
4. How to query or search knowledge bases?
5. How to set up Protégé for multiple users
6. How to export to external formats?
7. How to add and test integrity constraints?

5. How to set up Protégé for multiple users

- Problem: You have a Protégé knowledge base that you want multiple users (human and programs) to access or edit it simultaneously
- Considerations:
    - installation constraints (availability of network, thin or thick clients)
    - plug-in requirements
    - communication among users

Protégé web browser: Thin clients using mostly standard plug-ins

- Requires installation servlet-capable web server (e.g. Apache Tomcat) on server side
- Changes saved only with database backend
- Possible to add annotations
- Possible to get screen shots of Protégé GUI associated with each instance
- Possible to download projects
- Changes not propagated to different client browsers
- Configuration through metaproject

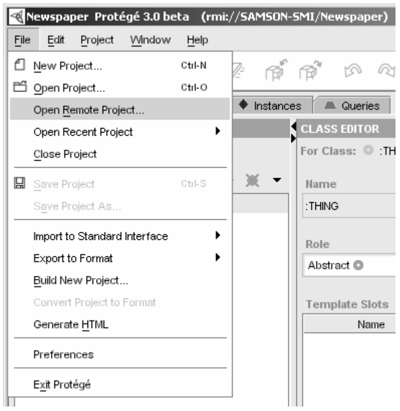


Adult viral Upper Respiratory Infection (common cold) guideline algorithm

Generated by Protege Web Browser, using Protege API
Send Feedback and Comments to Kamran Ahsan

Multi-user Protégé: Thick client using all available plug-ins

- Protégé installations on both server and client side
- Changes saved with database backend and in-memory backends
- Configuration through metaproject
- Firewalls an issue
- Included projects an issue



Detailed documentation at http://protege.stanford.edu/doc/multiuser/index.html

Scenario: Centralized server not an option

- Example: Individuals or groups work separately with no guaranteed access to central server
- Approach: Use PROMPT to compare and merge projects

Tasks in the life cycle of ontology and knowledge base development

1. How to reuse or import existing resources?
2. How to visualize information in the knowledge base?
3. How to manage multiple ontologies using Prompt
4. How to query or search knowledge bases?
5. How to set up Protégé for multiple users
6. How to export to external formats?
7. How to add and test integrity constraints?

6. How to export to external formats?

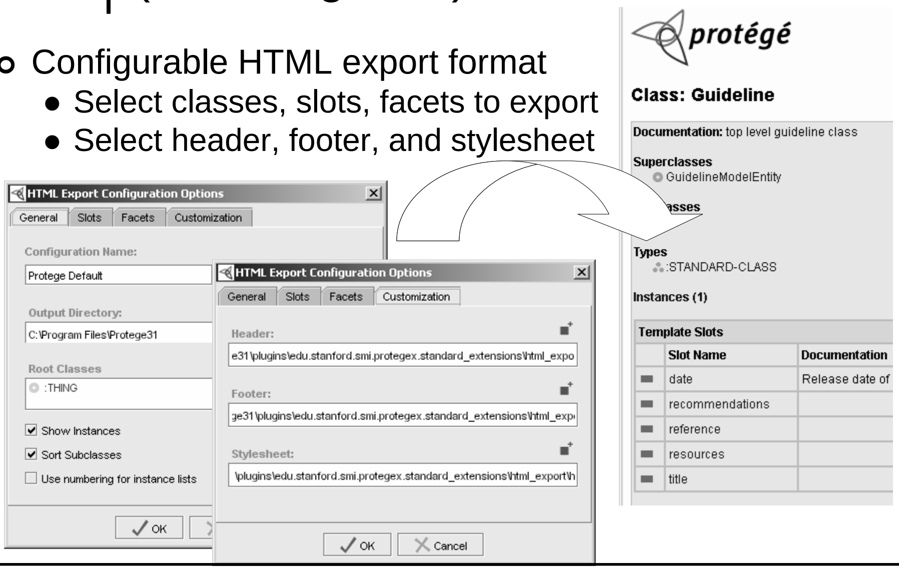
- Problem: Your users (and applications) require formats different from Protégé's backend formats
- Consideration
  - Entire Protégé knowledge base or just selected portions
  - One-way export or two-way conversions
  - The relative representational power of external formats
    - possible use of annotations to represent information that would be lost otherwise

Scenario: Entire Protégé knowledge base in two-way conversions

- Example: Request to make Protégé an RDF editor
- Protégé backends
  - Alternatives
    - 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
  - Not always possible
    - Often mismatch in knowledge models
  - Difficult to build and maintain
    - Requires updates as Protégé evolves

Export ontology or KB as HTML (In Protégé 3.1)

- Configurable HTML export format
  - Select classes, slots, facets to export
  - Select header, footer, and stylesheet



HTML Export Configuration Options

General | Slots | Facets | Customization

Configuration Name: Protege Default

Output Directory: C:\Program Files\Protege31

Root Classes: :THING

[✓] Show Instances

[✓] Sort Subclasses

[ ] Use numbering for instance lists

HTML Export Configuration Options

General | Slots | Facets | Customization

Header: e31\plugins\edu.stanford.smi.protegex.standard_extensions\html_expo

Footer: ge31\plugins\edu.stanford.smi.protegex.standard_extensions\html_exp

Stylesheet: \plugins\edu.stanford.smi.protegex.standard_extensions\html_export\h

protégé

Class: Guideline

Documentation: top level guideline class

Superclasses
GuidelineModelEntity

asses

Types
:STANDARD-CLASS

Instances (1)

Template Slots

	Slot Name	Documentation
▬	date	Release date of
▬	recommendations	
▬	reference	
▬	resources	
▬	title	

Scenario: Export to UML (Unified Modeling Language)

- UML: Dominant modeling standard in software engineering
- Solution: UML "backend"
  - Not a true backend: lose information (e.g., slot overrides)
  - Export in XMI 1.4 format
  - Readable in UML tool such as Poseidon

Scenario: Export to database

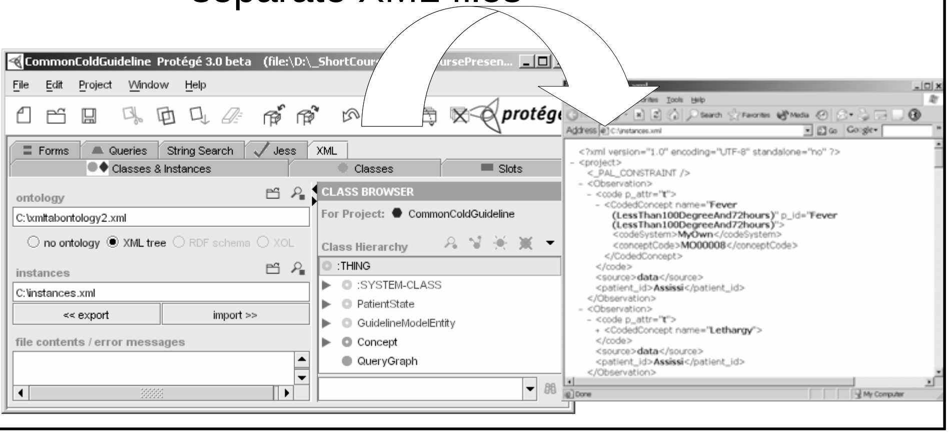
- Use case: Application programs use relational database management systems in multi-tier architecture
- No generic mapping of Protégé knowledge model to relational schema
  - e.g., Terminological classes akin to data
- Protégé stance
  - Protégé database backend optimized for Protégé user interface
  - Responsibility of local developers to create application-specific database export

Scenario: Export to XML

- Use case: Available tools for manipulating XML-formatted content
  - e.g., Use of XSLT to publish in alternative formats
- Solutions
  - OWL: external conversion tool
  - Specialized approaches
    - XML tab
    - Experimental XML file format

XML tab as exporter

- Export all classes and instances in separate XML files



CommonColdGuideline Protégé 3.0 beta (file:\D:_ShortCours…ursePresen…

ontology: C:\xmltabontology2.xml

○ no ontology [✓ XML tree] ○ RDF schema ○ XOL

instances: C:\instances.xml

For Project: CommonColdGuideline

:THING
- :SYSTEM-CLASS
- PatientState
- GuidelineModelEntity
- Concept
- QueryGraph

Address: C:\instances.xml

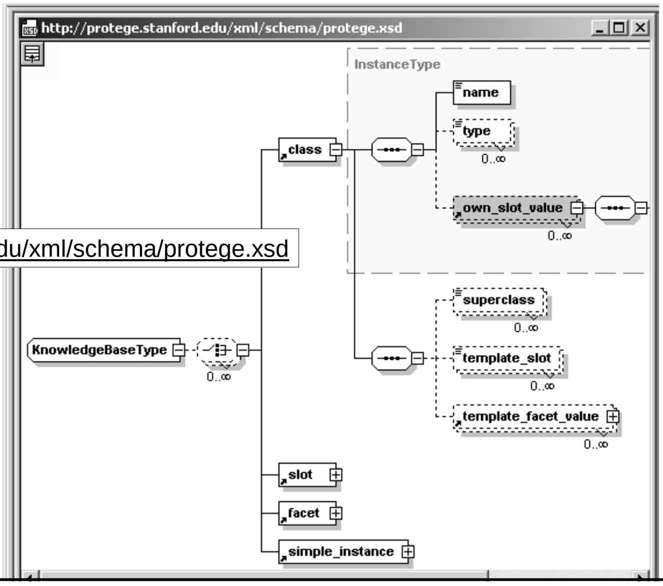
```
  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <project>
    <_PAL_CONSTRAINT />
  - <Observation>
    - <code p_attr="t">
      - <CodedConcept name="Fever
          (LessThan100DegreeAnd72hours)" p_id="Fever
          (LessThan100DegreeAnd72hours)">
          <codeSystem>MyOwn</codeSystem>
          <conceptCode>MO00008</conceptCode>
        </CodedConcept>
      </code>
      <source>data</source>
      <patient_id>Assissi</patient_id>
    </Observation>
  - <Observation>
    - <code p_attr="t">
      + <CodedConcept name="Lethargy">
      </code>
      <source>data</source>
      <patient_id>Assissi</patient_id>
    </Observation>
```

Experimental XML file format

- Uses Protégé XML schema

http://protege.stanford.edu/xml/schema/protege.xsd

- Will be default file format for frame-based projects



Tasks in the life cycle of ontology and knowledge base development

- How to reuse or import existing resources?
- How to visualize information in the knowledge base?
  - How to customize display of instances?
  - How to display non-standard data types?
  - What options are there to display portions of knowledge bases?
  - What options are available for navigation and browsing?
- How to manage multiple ontologies using Prompt
- How to query or search knowledge bases?
- How to set up Protégé for multiple users
- How to export to external formats?
- How to add and test integrity constraints?

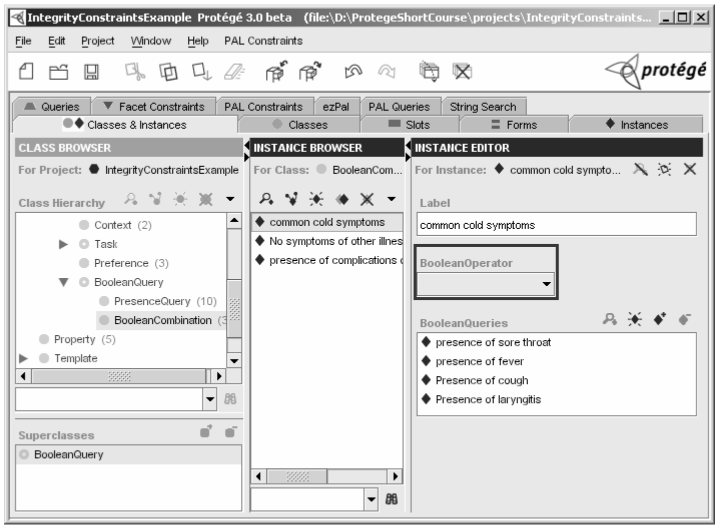
7. How to add and test integrity constraints?

- Problem: You want to verify that the statements encoded in your ontology and knowledge base satisfy some properties
- Considerations
  - What is the knowledge model (logic) of your ontology/KB
    - OWL: Java tests in Protégé OWL
    - Frame: slot constraints and PAL constraints

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

Violations of facet constraint are shown as slot widgets with red border



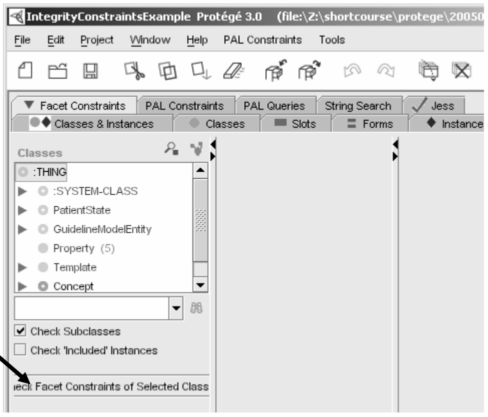
For Project: IntegrityConstraintsExample
Context (2)
Task
Preference (3)
BooleanQuery
PresenceQuery (10)
BooleanCombination (3
Property (5)
Template
Superclasses: BooleanQuery

For Class: BooleanCom...
common cold symptoms
No symptoms of other illnes
presence of complications c

For Instance: common cold sympto...
Label: common cold symptoms
BooleanOperator:
BooleanQueries:
presence of sore throat
presence of fever
Presence of cough
Presence of laryngitis

Facet-constraint tab brings together instances with facet-constraint violations

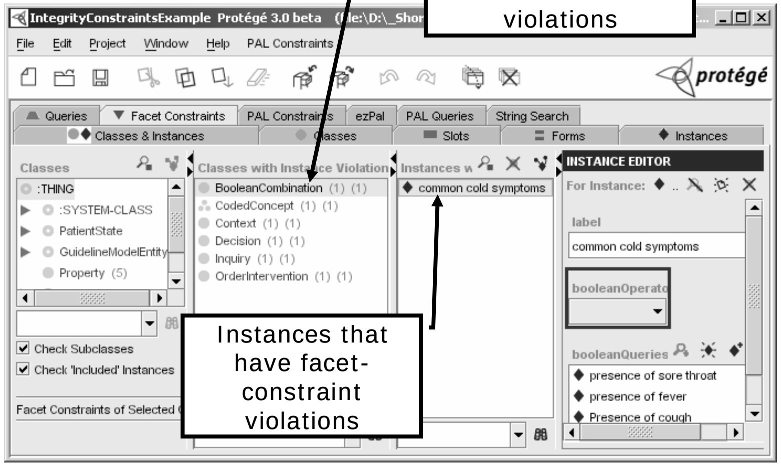
- Example: open any project
- Select Facet Constraints tab



Button to start facet-constraint checking

Select a class and the instance that have facet-constraint violations

Classes with instances that have facet-constraint violations



Instances that have facet-constraint violations

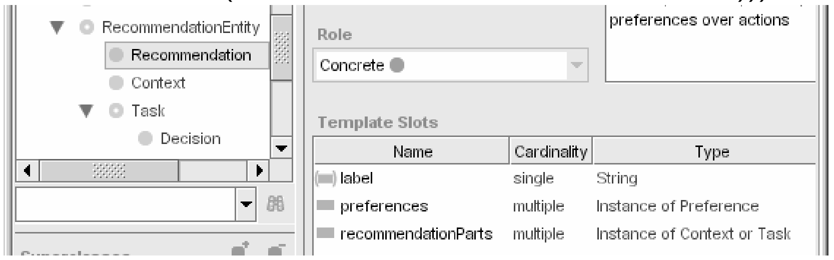
PAL constraints

- Overview
  - What PAL is and what it can be used for
  - How PAL is integrated into the Protégé framework
- ezPAL: fill-in-template method to write PAL constraints
- Mechanics: How to write PAL constraints

Example

- All decisions are part of recommendations

```
(defrange ?decision :FRAME Decision)
(defrange ?recomm :FRAME Recommendation)
(forall ?decision
      (exists ?recomm
            (recommendationParts ?recomm ?decision)))
```



Name	Cardinality	Type
label	single	String
preferences	multiple	Instance of Preference
recommendationParts	multiple	Instance of Context or Task

PAL: What's It?

- What it is:
  - A constraint language that helps to enforce the semantic properties of knowledge bases encoded in Protégé
  - A query language for searching instances that satisfy certain relationships
- What it is not:
  - A general predicate-logic language
  - A way to do write rules in Protégé
  - Another way to write definitions of concepts modeled in Protégé

A limited first-order logic extension of Protégé

- We decided on a variant of Knowledge Interchange Format (KIF)
- We use the KIF connectives and the KIF syntax
  - =, /=, not, and, or, =>, forall, exists
- Not all the KIF constants and predicates are included
  - (defrelation ...), (deffunction…) are omitted
  - added Protégé-specific predicates

```
(forall ?decision
      (exists ?recomm
            (recommendationParts ?recomm ?decision)))
```

Difference between constraints and axioms

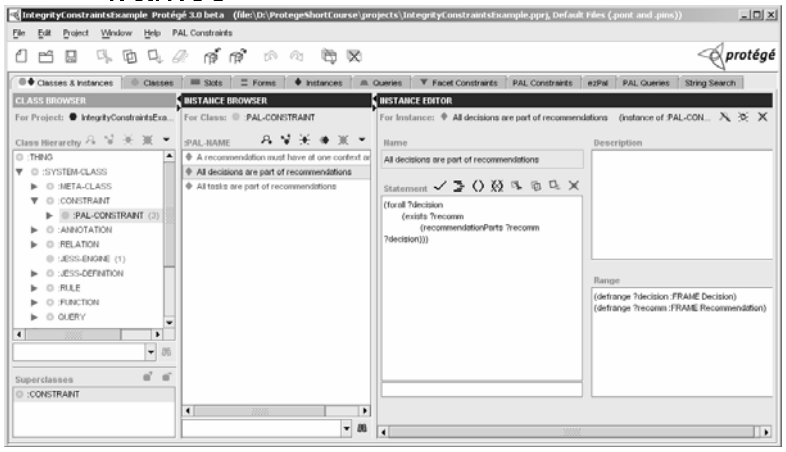
- Use the syntax of logic but have different semantics
  - Axioms are necessarily true
  - Constraints may be violated
- (exists ?y (mother-of John ?y))
  - Asserted as an axiom: John has a mother, even though there is no explicit object in KB
  - Asserted as a constraint: constraint is violated if no existing instance ?y satisfies the relation
- PAL: Protégé Constraint Language

PAL constraints

- Overview
  - What PAL is and what it can be used for
  - How PAL is integrated into the Protégé framework
- ezPAL: fill-in-template method to write PAL constraints
- Mechanics: How to write PAL constraints

Compatibility with Protégé framework (1)

- PAL constraints are themselves frames



IntegrityConstraintsExample Protégé 3.0 beta (file:\D:\ProtegeShortCourse\projects\IntegrityConstraintsExample.pprj, Default Files (.pont and .pins))

File Edit Project Window Help PAL Constraints
protégé
Classes & Instances | Classes | Slots | Forms | Instances | Queries | Facet Constraints | PAL Constraints | ezPal | PAL Queries | String Search
CLASS BROWSER
INSTANCE BROWSER
INSTANCE EDITOR

For Project: IntegrityConstraintsExa...

Class Hierarchy

```
:THING
▼ :SYSTEM-CLASS
    ▶ :META-CLASS
    ▼ :CONSTRAINT
        ▶ :PAL-CONSTRAINT (3)
    ▶ :ANNOTATION
    ▶ :RELATION
      :JESS-ENGINE (1)
    ▶ :JESS-DEFINITION
    ▶ :RULE
    ▶ :FUNCTION
    ▶ QUERY
```

Superclasses: :CONSTRAINT

For Class: :PAL-CONSTRAINT

:PAL-NAME

- A recommendation must have at one context ar
- All decisions are part of recommendations
- All tasks are part of recommendations

For Instance: All decisions are part of recommendations (instance of :PAL-CON...

Name: All decisions are part of recommendations

Description:

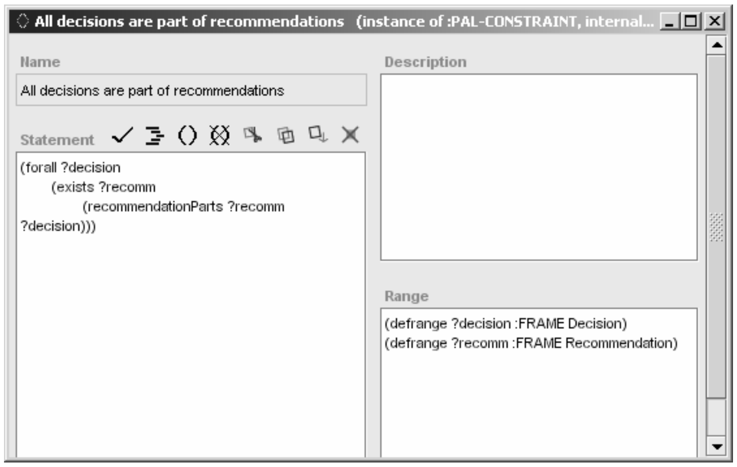
Statement

```
(forall ?decision
        (exists ?recomm
                (recommendationParts ?recomm
?decision)))
```

Range

```
(defrange ?decision :FRAME Decision)
(defrange ?recomm :FRAME Recommendation)
```

Specialized editor for viewing and editing PAL constraints



All decisions are part of recommendations (instance of :PAL-CONSTRAINT, internal...

Name: All decisions are part of recommendations

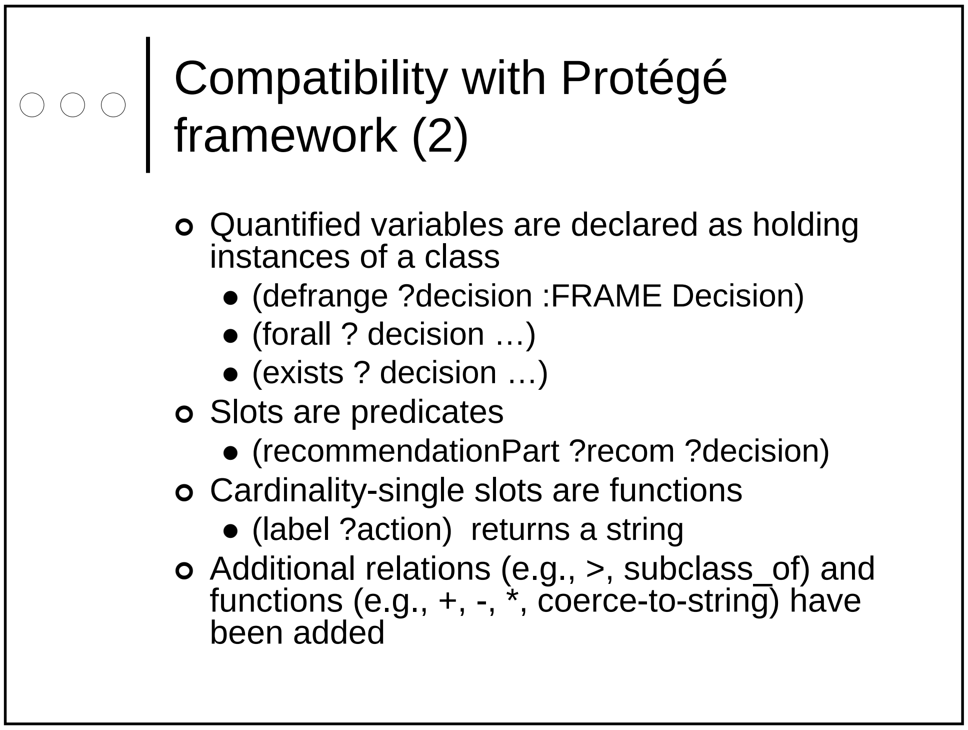
Description:

Statement

```
(forall ?decision
        (exists ?recomm
                (recommendationParts ?recomm
?decision)))
```

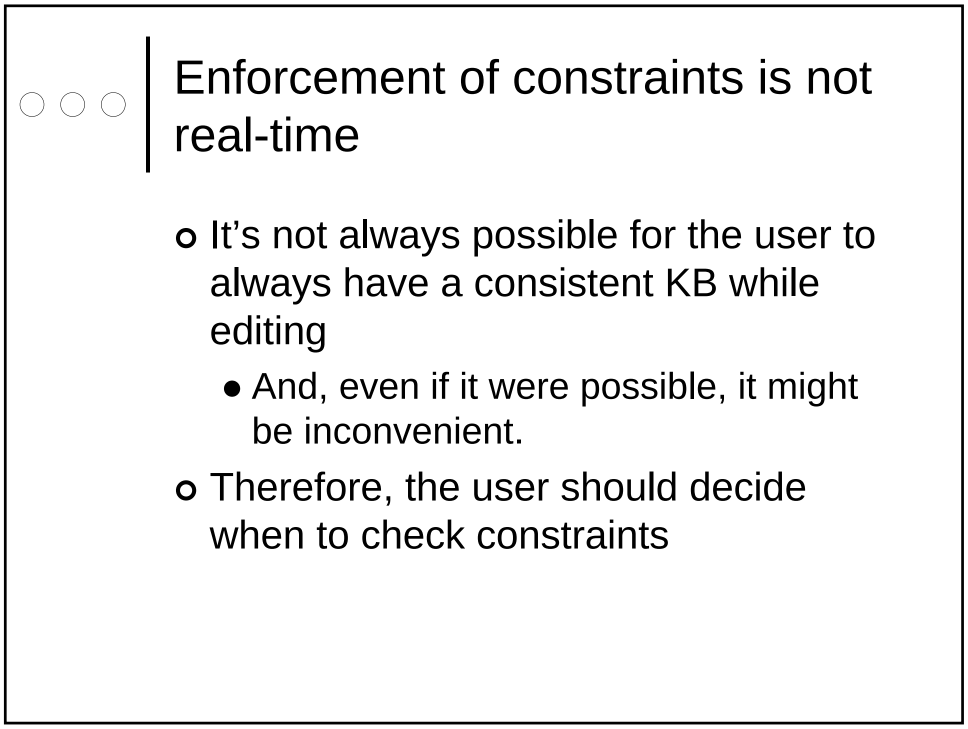
Range

```
(defrange ?decision :FRAME Decision)
(defrange ?recomm :FRAME Recommendation)
```



Compatibility with Protégé framework (2)

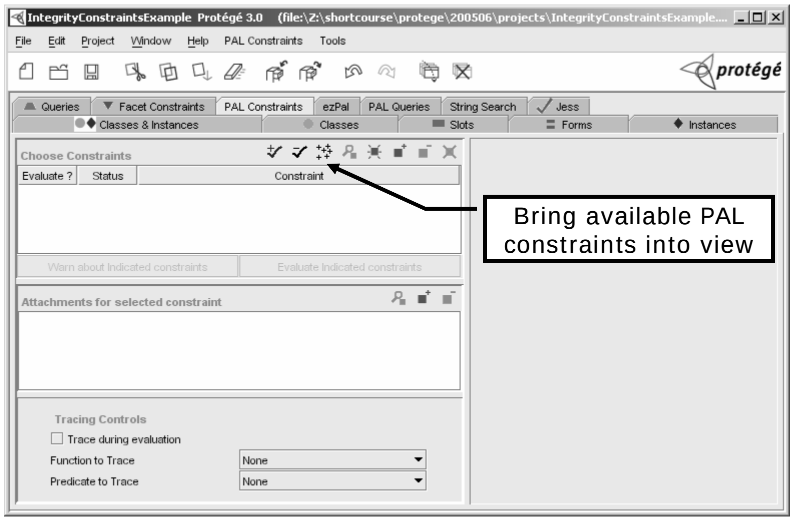
- Quantified variables are declared as holding instances of a class
  - (defrange ?decision :FRAME Decision)
  - (forall ? decision …)
  - (exists ? decision …)
- Slots are predicates
  - (recommendationPart ?recom ?decision)
- Cardinality-single slots are functions
  - (label ?action) returns a string
- Additional relations (e.g., >, subclass_of) and functions (e.g., +, -, *, coerce-to-string) have been added



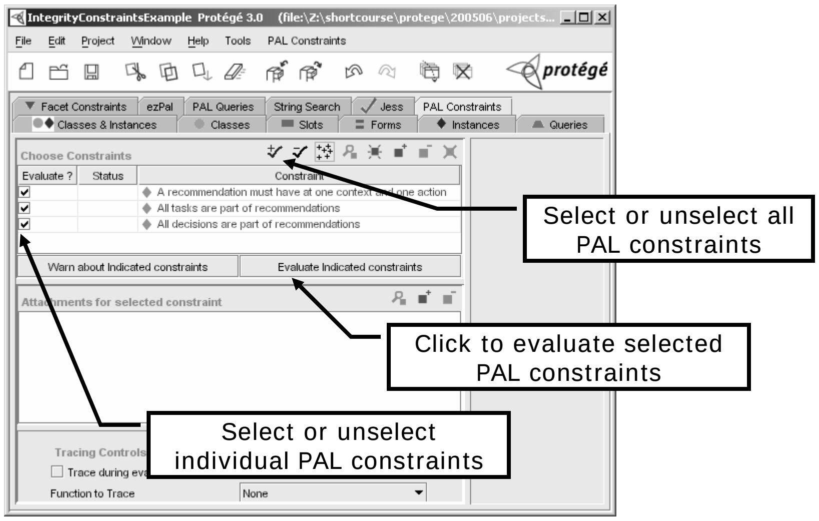
Enforcement of constraints is not real-time

- It’s not always possible for the user to always have a consistent KB while editing
  - And, even if it were possible, it might be inconvenient.
- Therefore, the user should decide when to check constraints

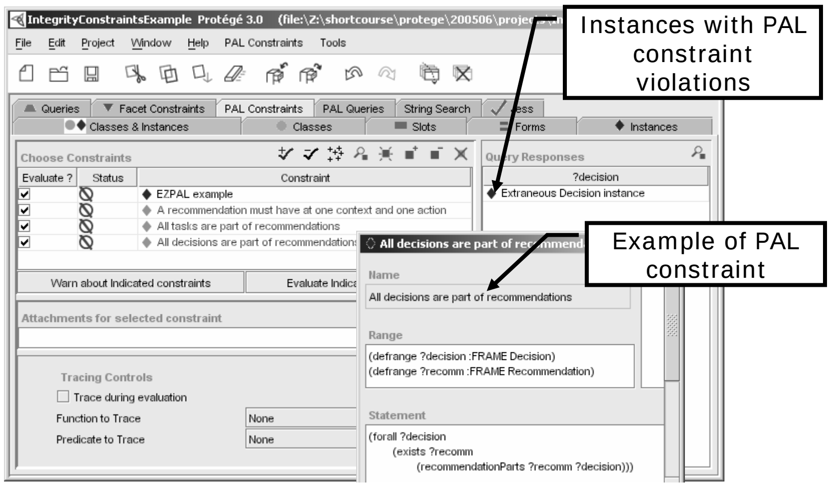
PALConstraints Tab: Allows specification and checking of complex integrity constraints


IntegrityConstraintsExample Protégé 3.0 (file:\Z:\shortcourse\protege\200506\projects\IntegrityConstraintsExample....
File Edit Project Window Help PAL Constraints Tools
protégé
Queries Facet Constraints PAL Constraints ezPal PAL Queries String Search Jess
Classes & Instances Classes Slots Forms Instances
Choose Constraints
Evaluate ? Status Constraint
Warn about Indicated constraints Evaluate Indicated constraints
Attachments for selected constraint
Tracing Controls
Trace during evaluation
Function to Trace None
Predicate to Trace None
Bring available PAL constraints into view

Operations on selected PAL constraint


IntegrityConstraintsExample Protégé 3.0 (file:\Z:\shortcourse\protege\200506\projects...
File Edit Project Window Help Tools PAL Constraints
protégé
Facet Constraints ezPal PAL Queries String Search Jess PAL Constraints
Classes & Instances Classes Slots Forms Instances Queries
Choose Constraints
Evaluate ? Status Constraint
A recommendation must have at one context and one action
All tasks are part of recommendations
All decisions are part of recommendations
Warn about Indicated constraints Evaluate Indicated constraints
Attachments for selected constraint
Tracing Controls
Trace during evaluation
Function to Trace None
Select or unselect all PAL constraints
Click to evaluate selected PAL constraints
Select or unselect individual PAL constraints

Evaluate selected constraints


Instances with PAL constraint violations
Example of PAL constraint
IntegrityConstraintsExample Protégé 3.0 (file:\2:\shortcourse\protege\200506\proje
File Edit Project Window Help PAL Constraints Tools
Queries
Facet Constraints
PAL Constraints
PAL Queries
String Search
Jess
Classes & Instances
Classes
Slots
Forms
Instances
Choose Constraints
Query Responses
Warn about Indicated constraints
Evaluate Indica
Attachments for selected constraint
Tracing Controls
All decisions are part of recommend

Evaluate ?	Status	Constraint
✓	⦸	EZPAL example
✓	⦸	A recommendation must have at one context and one action
✓	⦸	All tasks are part of recommendations
✓	⦸	All decisions are part of recommendations

?decision
Extraneous Decision instance

☐ Trace during evaluation

Function to Trace: None

Predicate to Trace: None

Name: All decisions are part of recommendations

Range:

```
(defrange ?decision :FRAME Decision)
(defrange ?recomm :FRAME Recommendation)
```

Statement:

```
(forall ?decision
      (exists ?recomm
            (recommendationParts ?recomm ?decision)))
```

PAL Query Language

- Taking a constraint and finding instances that violate it is much like finding instances that satisfy a statement
  - (not (exists ?x (…))
- Query engine introduces two keywords
  - find
  - findall

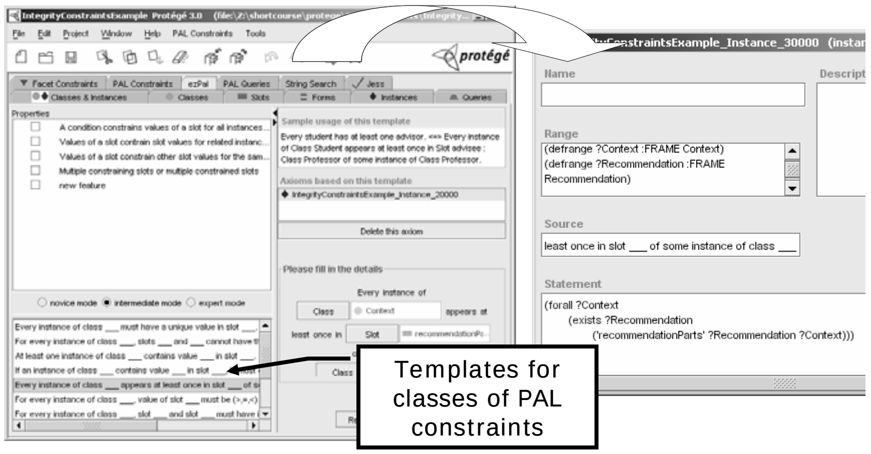
```
(findall ?node
   (exists ?decision (and
      (exists ?preference (and (:FROM ?preference ?decision)
                               (:TO ?preference ?node)))
                (not (alternatives ?decision ?node)))))
```

PAL constraints

- Overview
  - What PAL is and what it can be used for
  - How PAL is integrated into the Protégé framework
- ezPAL: fill-in-template method to write PAL constraints
- Mechanics:How to write PAL constraints

EZPAL tab

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


IntegrityConstraintsExample Protégé 3.0 (file:\Z:\shortcourse\protege\
File Edit Project Window Help PAL Constraints Tools
protégé
Facet Constraints
PAL Constraints
ezPal
PAL Queries
String Search
Jess
Classes & Instances
Classes
Slots
Forms
Instances
Queries
Properties
Sample usage of this template
Axioms based on this template
Delete this axiom
Please fill in the details
Class
Slot
Name
Descript
Range
Source
Statement

Properties:
- ☐ A condition constrains values of a slot for all instances...
- ☐ Values of a slot contrain slot values for related instanc...
- ☐ Values of a slot constrain other slot values for the sam...
- ☐ Multiple constraining slots or multiple constrained slots
- ☐ new feature

○ novice mode ● intermediate mode ○ expert mode

Every instance of class ___ must have a unique value in slot ___
For every instance of class ___, slots ___ and ___ cannot have t
At least one instance of class ___ contains value ___ in slot ___
If an instance of class ___ contains value ___ in slot ___ must
Every instance of class ___ appears at least once in slot ___ of s
For every instance of class ___, value of slot ___ must be (>,=,<)
For every instance of class ___, slot ___ and slot ___ must have

Sample usage of this template: Every student has at least one advisor. <=> Every instance of Class Student appears at least once in Slot advisee : Class Professor of some instance of Class Professor.

Axioms based on this template: IntegrityConstraintsExample_Instance_20000

Please fill in the details: Every instance of Class Context appears at least once in Slot recommendationP...

ityConstraintsExample_Instance_30000 (instan

Range:
```
(defrange ?Context :FRAME Context)
(defrange ?Recommendation :FRAME
Recommendation)
```

Source: least once in slot ___ of some instance of class ___

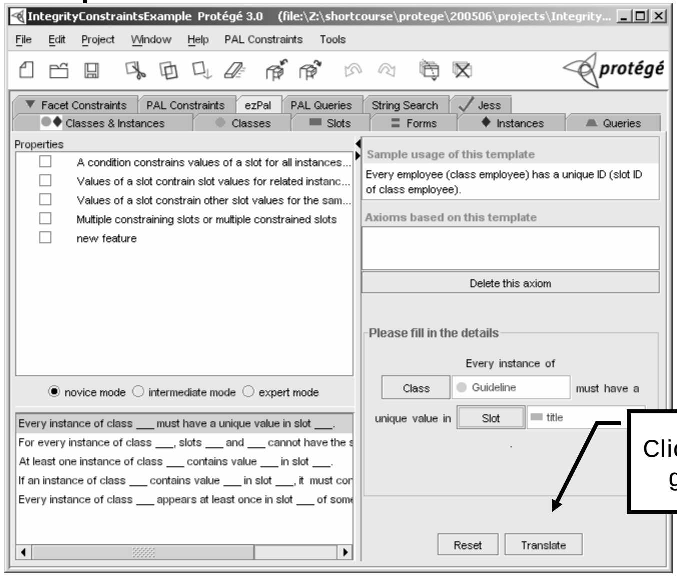
Statement:
```
(forall ?Context
     (exists ?Recommendation
          ('recommendationParts' ?Recommendation ?Context)))
```

Templates for classes of PAL constraints

Exercise: Use of ezPAL

- Make sure that template.pprj is included
- Select property and mode
- Want to write constraint
  - Every instances of class Guideline must have a unique value for the *title* slot
- Select template, and fill in details

You should have



IntegrityConstraintsExample Protégé 3.0 (file:\Z:\shortcourse\protege\200506\projects\Integrity...
File Edit Project Window Help PAL Constraints Tools
protégé
Facet Constraints PAL Constraints ezPal PAL Queries String Search Jess
Classes & Instances Classes Slots Forms Instances Queries
Properties
Delete this axiom
Reset Translate

A condition constrains values of a slot for all instances...
Values of a slot contrain slot values for related instanc...
Values of a slot constrain other slot values for the sam...
Multiple constraining slots or multiple constrained slots
new feature

[✓ novice mode] intermediate mode expert mode

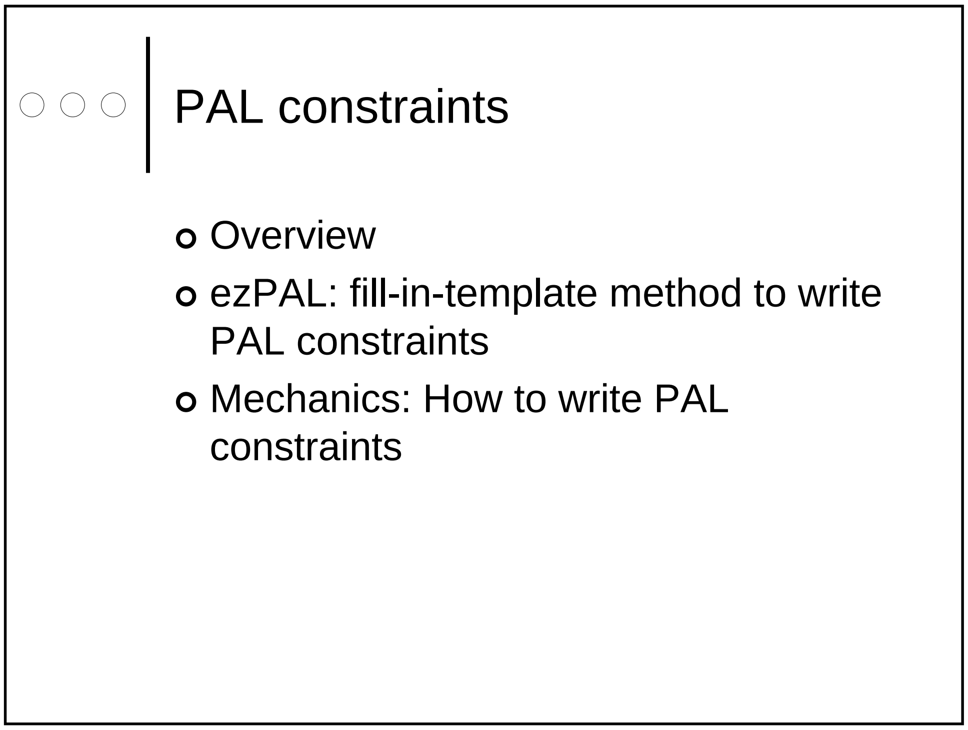
Every instance of class ___ must have a unique value in slot ___.
For every instance of class ___, slots ___ and ___ cannot have the s
At least one instance of class ___ contains value ___ in slot ___.
If an instance of class ___ contains value ___ in slot ___, it must cor
Every instance of class ___ appears at least once in slot ___ of some

Sample usage of this template
Every employee (class employee) has a unique ID (slot ID of class employee).

Axioms based on this template

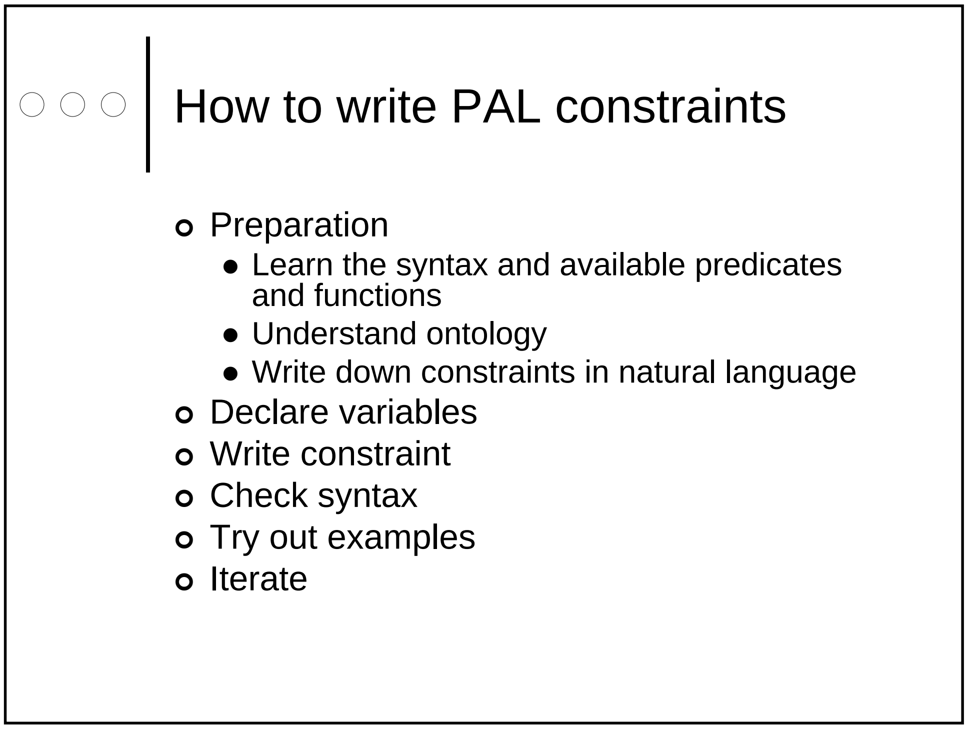
Please fill in the details
Every instance of Class: Guideline must have a unique value in Slot: title .

Click Translate to generate PAL



PAL constraints

- Overview
- ezPAL: fill-in-template method to write PAL constraints
- Mechanics: How to write PAL constraints



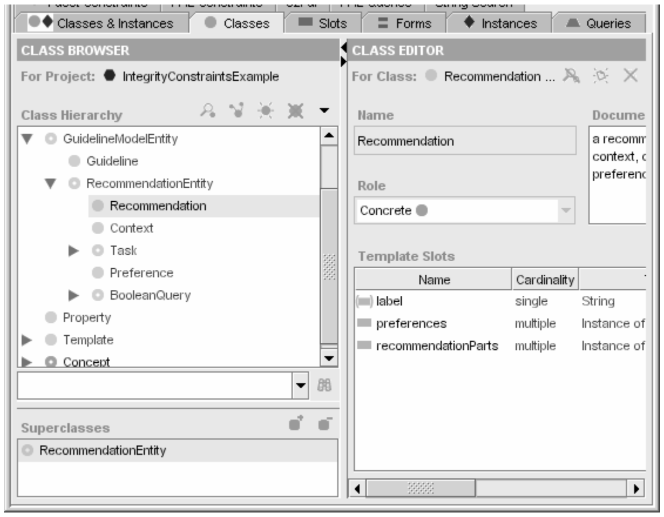
How to write PAL constraints

- Preparation
  - Learn the syntax and available predicates and functions
  - Understand ontology
  - Write down constraints in natural language
- Declare variables
- Write constraint
- Check syntax
- Try out examples
- Iterate

Learn the syntax and available predicates and functions

- http://protege.stanford.edu/plugins/paltabs/pal-documentation/index.html
- Best to re-write examples
- Use PAL editor
  - shows available predicates for selection
  - syntax checking

Understand ontology



For Project: IntegrityConstraintsExample

- GuidelineModelEntity
  - Guideline
  - RecommendationEntity
    - Recommendation
    - Context
    - Task
    - Preference
    - BooleanQuery
- Property
- Template
- Concept

Superclasses: RecommendationEntity

For Class: Recommendation ...

Name: Recommendation

Role: Concrete

Docume
a recomm
context, c
preferenc

Name	Cardinality	
label	single	String
preferences	multiple	Instance of
recommendationParts	multiple	Instance of

Write constraint in natural language

- A recommendation must have at least one context and one action

Declare Variables

- Quantified variables refer to frames
- Syntax for explicit declaration
  - local variable (defrange ?action :FRAME Action)
  - global variable %action: Don’t use

```
(defrange ?recomm :FRAME Recommendation)
(defrange ?context :FRAME Context)
(defrange ?act :FRAME Action)
```

Protégé allows the use of *free variable* (undeclared variable) in a PAL constraint if the constraint is attached to a class. However this practice has unintended consequences. Always declare variables.

Write Constraint

```
(defrange ?recomm :FRAME Recommendation)
(defrange ?context :FRAME Context)
(defrange ?act :FRAME Action)

(forall ?recomm
      (and (exists ?context
                (recommendationParts ?recomm ?context))
          (exists ?act
                (recommendationParts ?recomm ?act))
      )
)
```

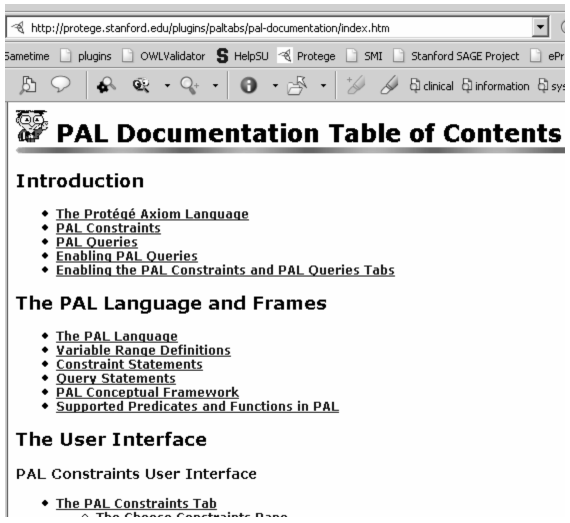
Debug PAL Constraints

- Use tools that help with balancing of parenthesis and with indentations
  - PAL editor does these and more!
- Try small examples
- Test components
- Trace built-in predicates and functions
- Write existence clauses as queries

Summary: PAL

- Integrity constraints on knowledge bases help you to catch errors
- Frame-based Protégé allows use of two types of constraints
  - Facet constraints: constraints on properties of a slot
  - PAL constraints: constraints on instances and their relationships
- Tools like EZPal tab, facet-constraint tab, and PAL-constraint tab help you to write and check constraints

Comprehensive documentation is available http://protege.stanford.edu/plugins/paltabs/pal-documentation/



PAL Documentation Table of Contents

Introduction

- The Protégé Axiom Language
- PAL Constraints
- PAL Queries
- Enabling PAL Queries
- Enabling the PAL Constraints and PAL Queries Tabs

The PAL Language and Frames

- The PAL Language
- Variable Range Definitions
- Constraint Statements
- Query Statements
- PAL Conceptual Framework
- Supported Predicates and Functions in PAL

The User Interface

PAL Constraints User Interface

- The PAL Constraints Tab
  - The Choose Constraints Pane

Summary

- Protégé plugin architecture allows functionalities to be added to core Protégé
- Tasks in the life cycle of ontology and knowledge base development
  - How to reuse or import existing resources?
  - How to visualize information in the knowledge base?
  - How to manage multiple ontologies using Prompt
  - How to query or search knowledge bases?
  - How to set up Protégé for multiple users
  - How to export to external formats?
  - How to add and test integrity constraints?