



Protégé 101: Protégé Basics

March 23, 2006

Samson Tu
Stanford Medical Informatics
Stanford University

Goal of this session: Introduction to the use of Protégé

- Protégé a complex tool
- Learning use of tool needs repetitive practice
- Focus of the session
 - Get to know Protégé GUI for developing guideline knowledge base
 - Practice basic operations

Outline

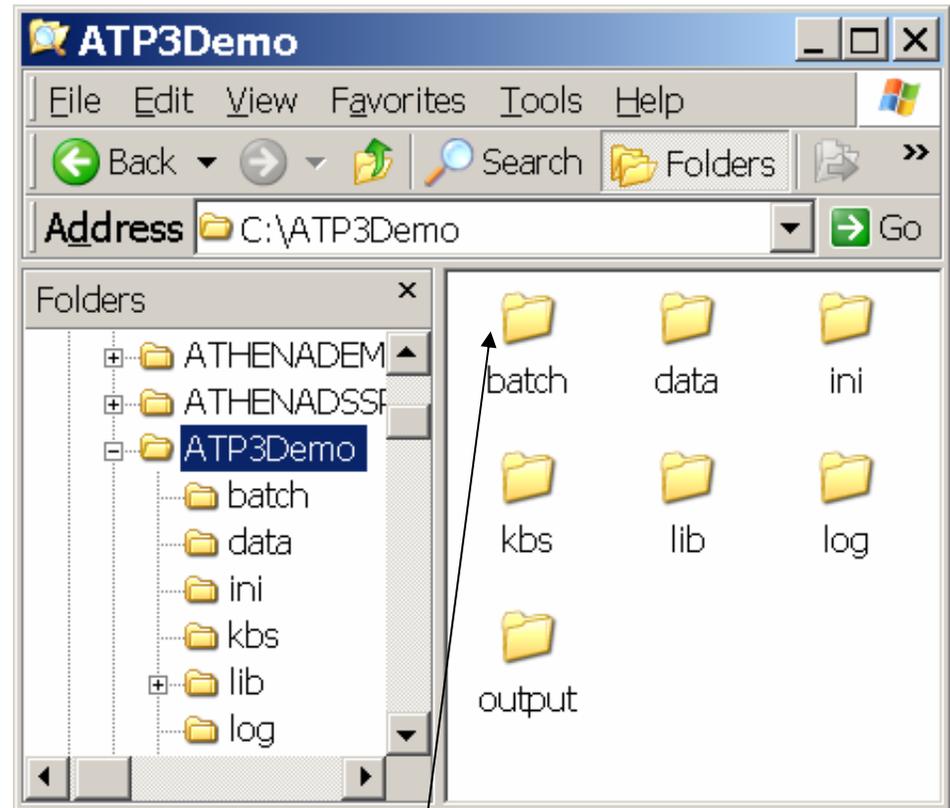
- Distinctions: Protégé/EON/ATHENA
- Installation
- Navigating and working with classes and instances
- Working with Protégé projects

What's the relationship among Protégé, EON, and ATHENA?

- Protégé: a generic ontology/knowledge-base development environment
 - 48,000+ registered users
 - Different representation formalisms
- EON
 - set of models and software
 - for encoding and applying guidelines and protocols
 - to provide patient-specific decision support
- ATHENA: VA project using EON and Protégé for CDSS
 - ATHENA Hypertension
 - ATHENA Opioid Therapy

Installing Protégé for this Workshop

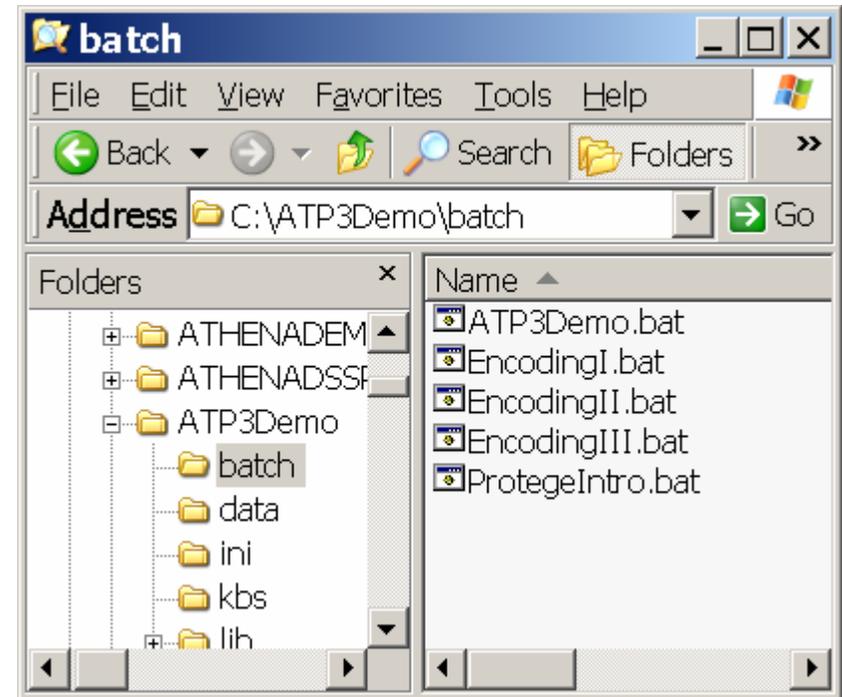
- Copy the “ATP3Demo” folder to the C: drive of your computer
- Protégé is freely available on the web
- This version is custom-tailored for the workshop
 - Contains testing environment
 - Contains commercial software for which we obtained permission for demonstration purpose (Do not re-distribute software)



batch/ contains scripts
for running Protégé

The *batch*/ folder

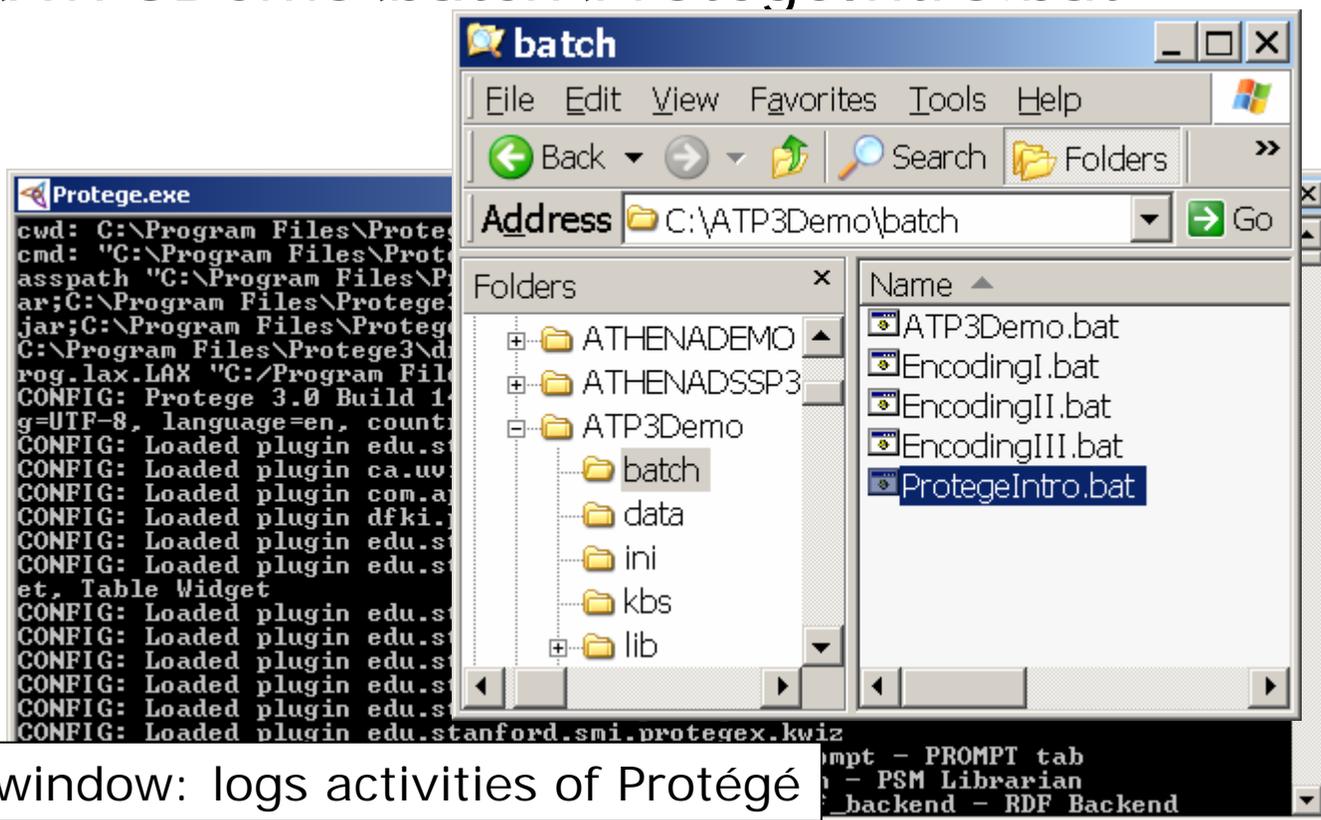
- ProtegeIntro.bat
 - For this session
- EncodingI, II, III
 - For tomorrow's sessions
- ATP3Demo.bat
 - For running a completed encoding of (a portion of) ATP3 guideline



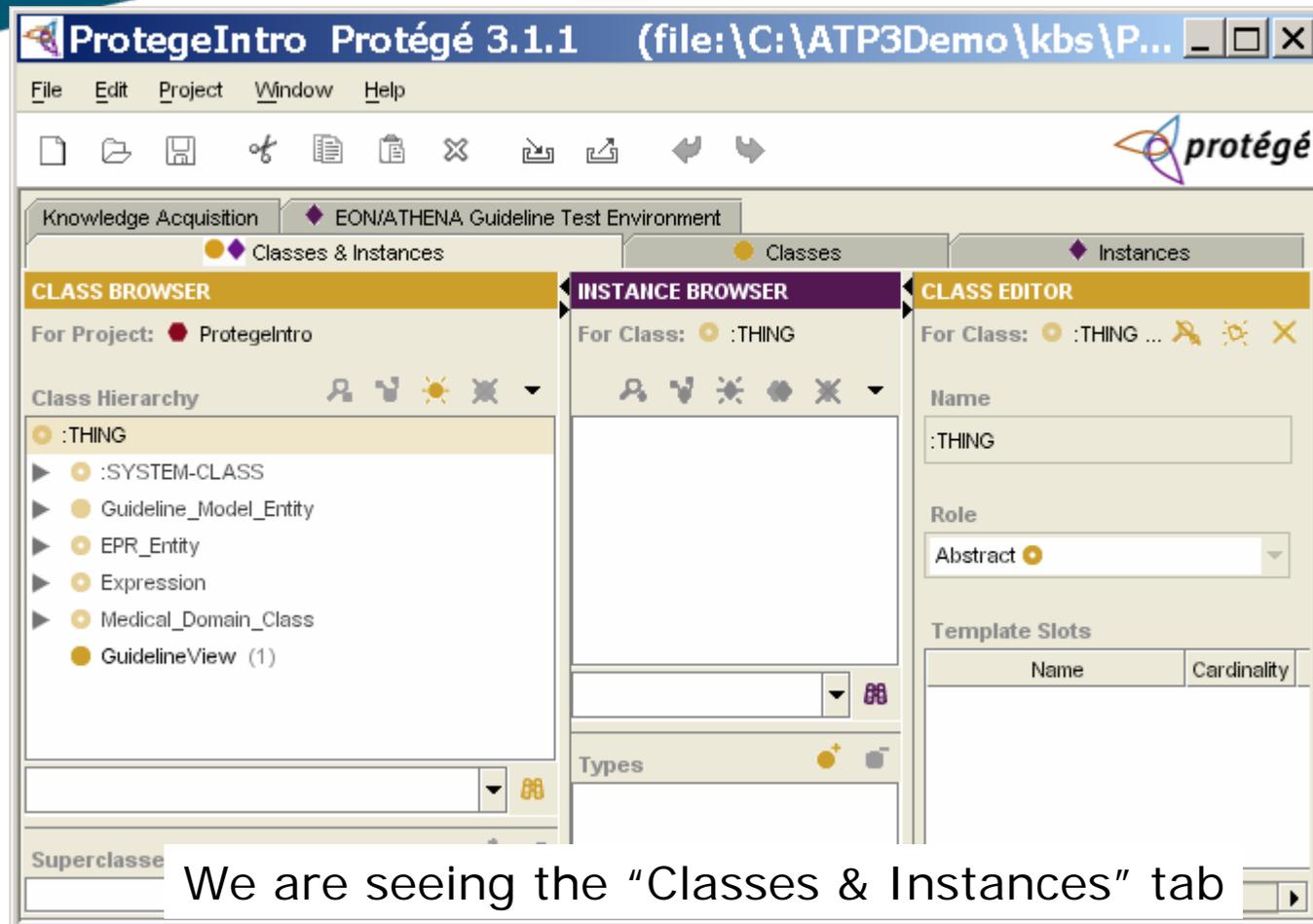
Tip: Create shortcuts for these bat files on your desktop

Starting Protégé for the Session

- Double-click on
C:\ATP3Demo\batch\ProtegeIntro.bat



This is what you see: Protégé user interface organized in “tabs”

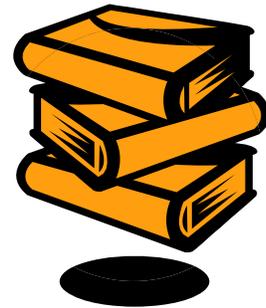


Basic Ideas of Protégé

- Represent things of interest in a domain as
 - **Classes** (e.g., Guideline, Medication)
 - **Instances** (individuals e.g., JNC 7 guideline)
 - **Slots** (properties e.g., author, eligibility criteria)
- Generate **customizable instance forms** from definitions of a class
- Edit details of classes, properties, and instances on forms
- Develop applications that make use of resulting knowledge base

I. What is a Class?

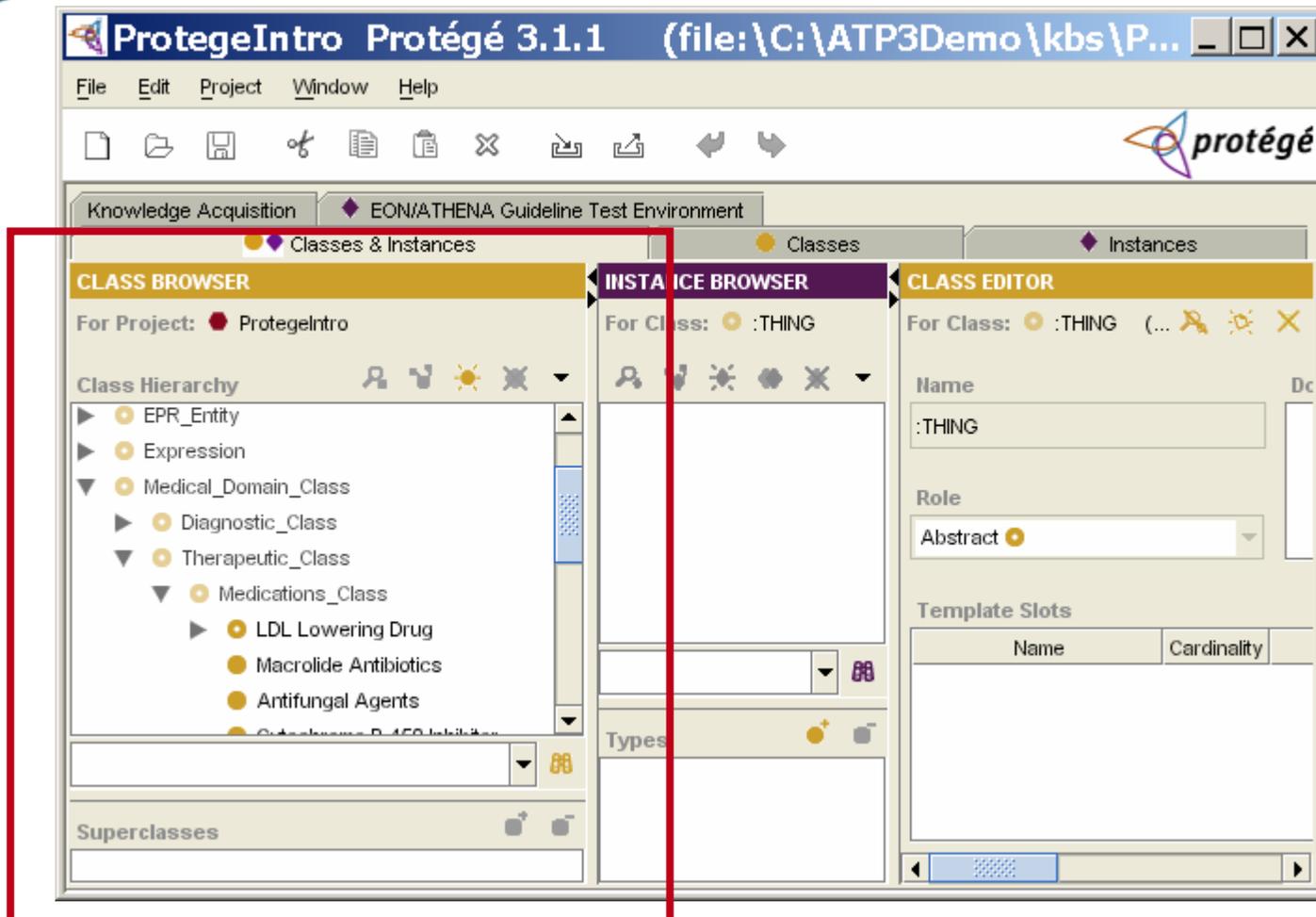
- Class: set of individuals that share characteristics
 - The set of all cats
 - The set of practice guidelines
- Class – Subclass relationship
 - The set of mammals includes the set of cats => the Cat class is a subclass of the Mammal class



II. What is a Class?

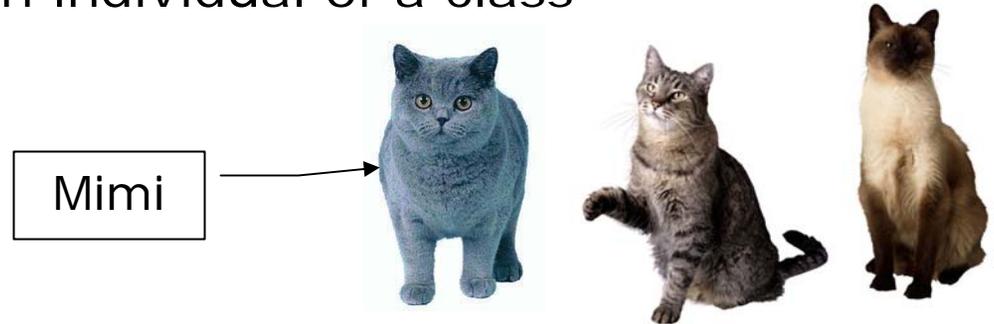
- Class as terminological concept
 - Give me **liberty** or give me **death!**
 - Medication prescription of **lisinopril** at 20 **mg/day**
- Terminological concepts can be arranged in a class-subclass hierarchy
 - ACE inhibitor
 - lisinopril
 - lisinopril 10 mg
 - Unit
 - weight unit
 - mg

Class Hierarchy Shown in Class Browser Pane in Classes & Instances tab



What is an Instance?

- An instance is an individual of a class



- Instances can be described in terms of the properties they have
 - **breed_of** *Mimi* is "British Shorthair"
 - **color** of *Mimi* is blue
 - **age** of *Mimi* is 3 years
- Properties are also called *slots* in Protégé

A Class defines the properties of its instances: Navigate to *Management_Guideline* Class

The screenshot shows the Protege 3.1.1 interface. The title bar reads "ProtegeIntro Protégé 3.1.1 (file:\C:\ATP3Demo\kbs\Pr...". The menu bar includes "File", "Edit", "Project", "Window", and "Help". The toolbar contains various icons for file operations and editing. The main workspace is divided into several panes:

- Class Hierarchy:** A tree view showing the class structure. The "Management_Guideline" class is selected and highlighted in yellow.
- CLASS EDITOR:** A pane for editing the selected class. It shows the class name "Management_Guideline" and its role as "Concrete".
- Template Slots:** A table listing the slots of the class.

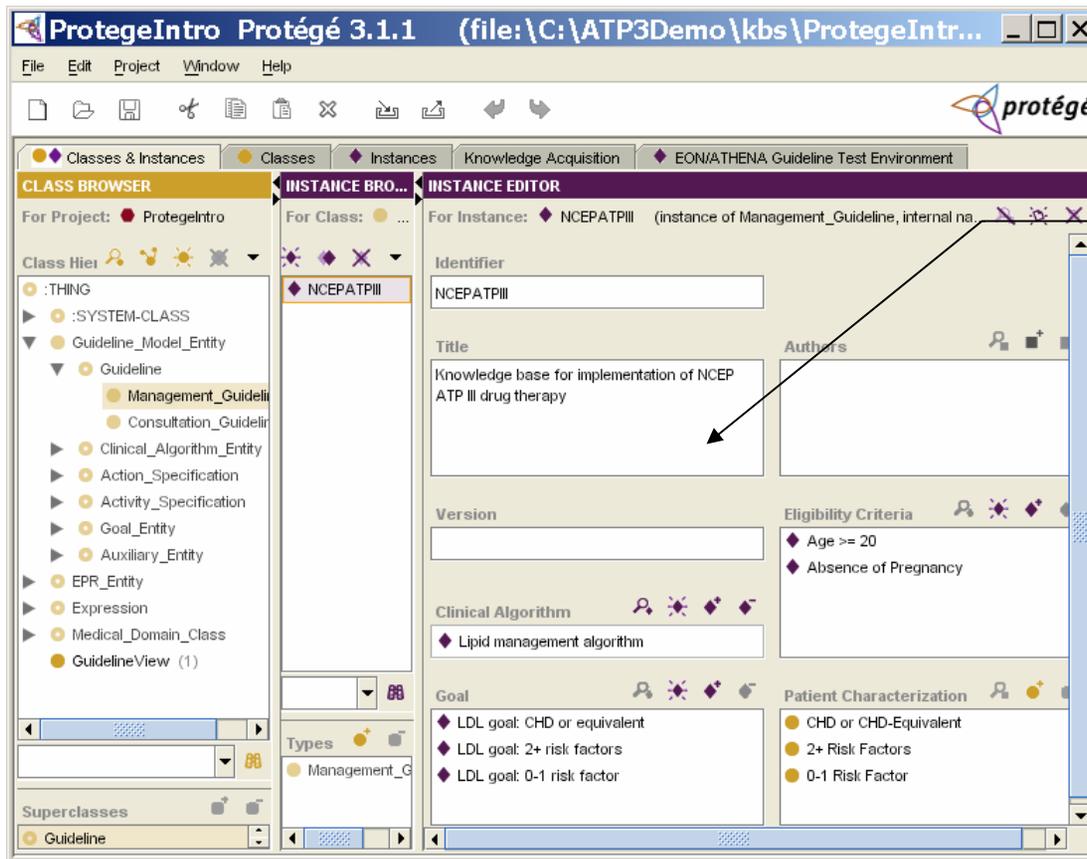
Two callout boxes provide additional information:

- 1. Class hierarchy:** Points to the "Management_Guideline" class in the hierarchy pane.
- 2. Properties for instances of the selected class (e.g., authors of the guideline):** Points to the "authors" slot in the Template Slots table.

Name	Cardinality	Symbol
assume_if_no_goal_v...	single	Symbol
authors	multiple	String
clinical_algorithm	required ...	Instance of Management...
eligibility_criteria	multiple	Instance of Criterion
goal	multiple	Instance of Goal_Entity
label	required ...	String
patient_characterizati...	multiple	Class with superclass D...
primary	single	Boolean

Protégé generate customizable instance forms from definitions of a class

- Click on *NCEPATPIII* in the middle column

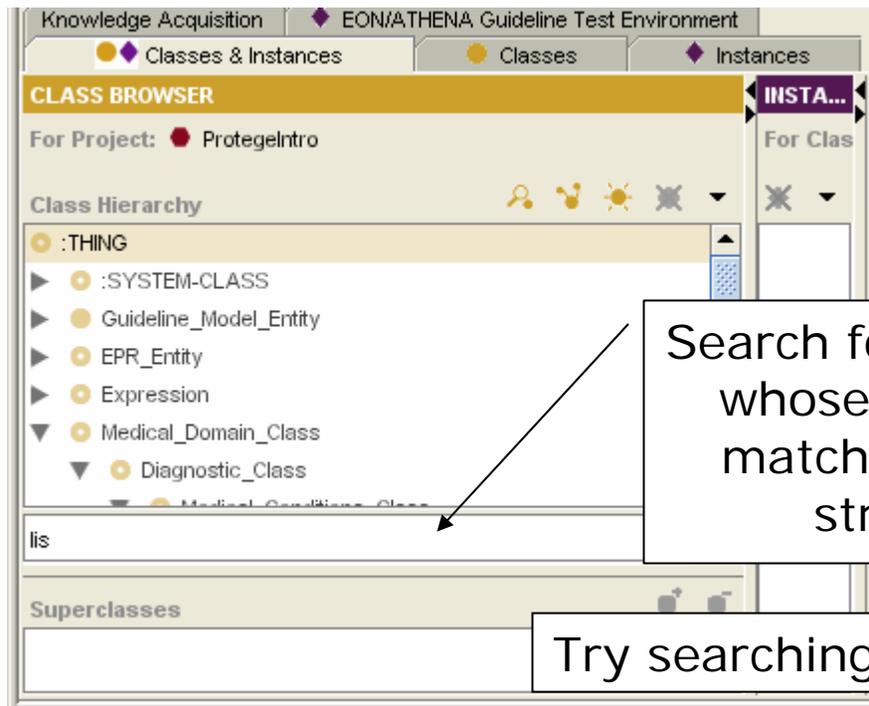


A form for editing details of the NCEPATPIII instance of Management Guideline

Recap: Basic Ideas of Protégé

- Represent things of interest in a domain as
 - **Classes** (e.g., Guideline, Medication)
 - **Instances** (individuals e.g., JNC 7 guideline)
 - **Slots** (properties e.g., author, eligibility criteria)
- Generate **customizable instance forms** from definitions of a class
- Edit details of classes, properties, and instances on forms
- Develop applications that make use of resulting knowledge base

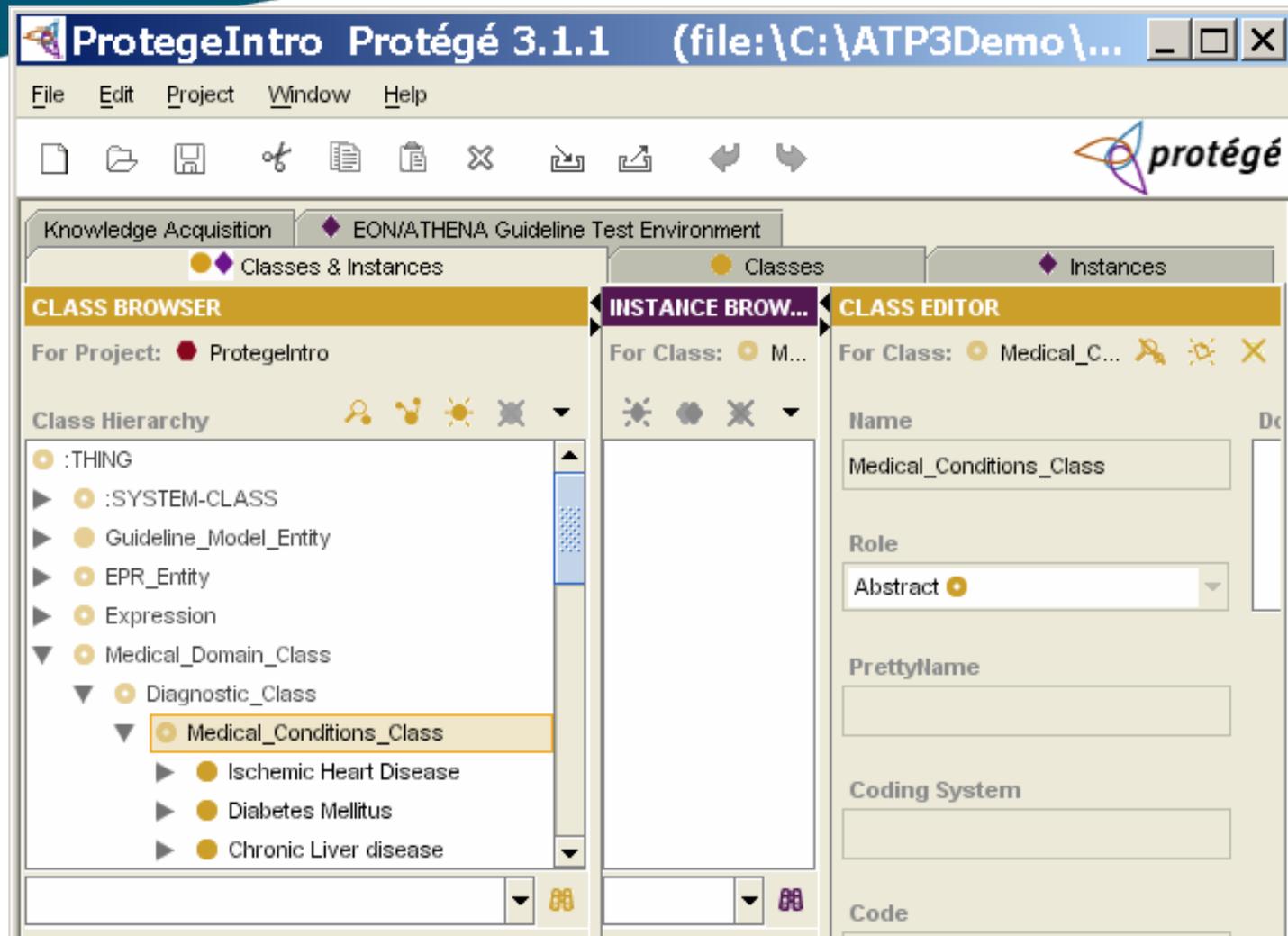
Searching for a class in the hierarchy



Search for classes whose names match search string

Try searching for "lis" and then "*lis"

Operations on classes: First navigate to *Medical_Conditions_Class*



Operations on classes

The screenshot shows the Protege 3.1.1 interface with the following components and callouts:

- Callout 1:** "Delete the selected class" points to the delete icon (an 'X') in the toolbar above the Class Hierarchy panel.
- Callout 2:** "Open a window to view the selected class" points to the "INSTANCE BROW..." button in the toolbar above the Class Editor panel.
- Callout 3:** "Find all places where this class is referenced" points to the search icon (magnifying glass) in the toolbar above the Class Hierarchy panel.
- Callout 4:** "Create a new subclass of the selected class" points to the "Add Class" icon (a sun-like symbol) in the toolbar above the Class Hierarchy panel.

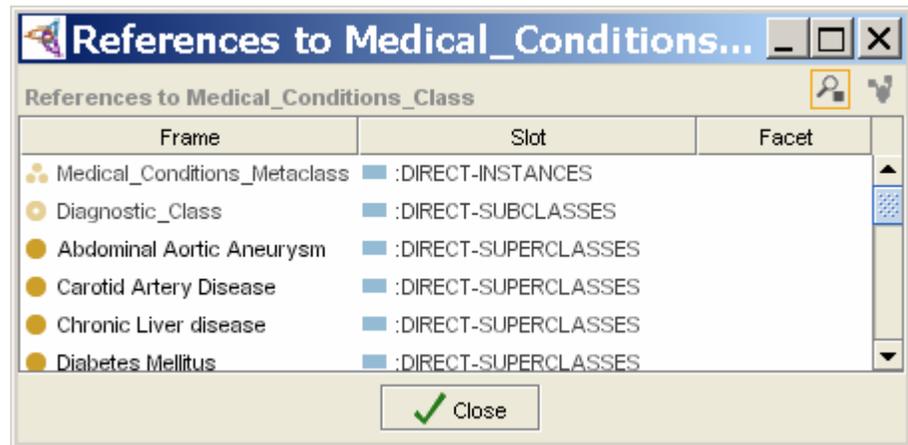
The Class Hierarchy panel shows a tree structure with the following classes:

- :THING
 - :SYSTEM-CLASS
 - Guideline_Model_Entity
 - EPR_Entity
 - Expression
 - Medical_Domain_Class
 - Diagnostic_Class
 - Medical_Conditions_Class
 - Ischemic Heart Disease
 - Diabetes Mellitus
 - Chronic Liver disease

Let's try them!

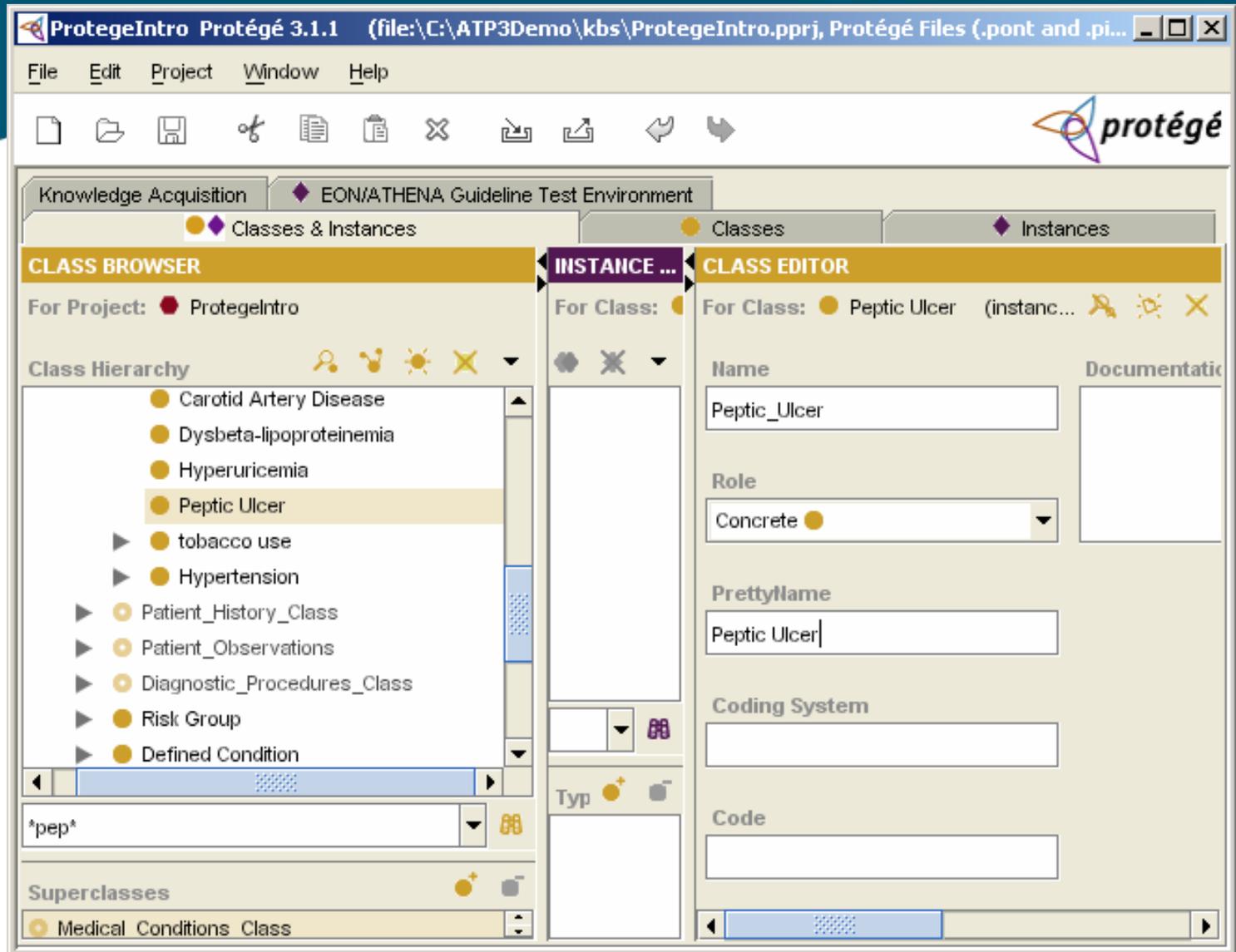
Let's try it!

- Open `Medical_Conditions_Class` class editor
- Find all places where `Medical_Conditions_Class` is referenced

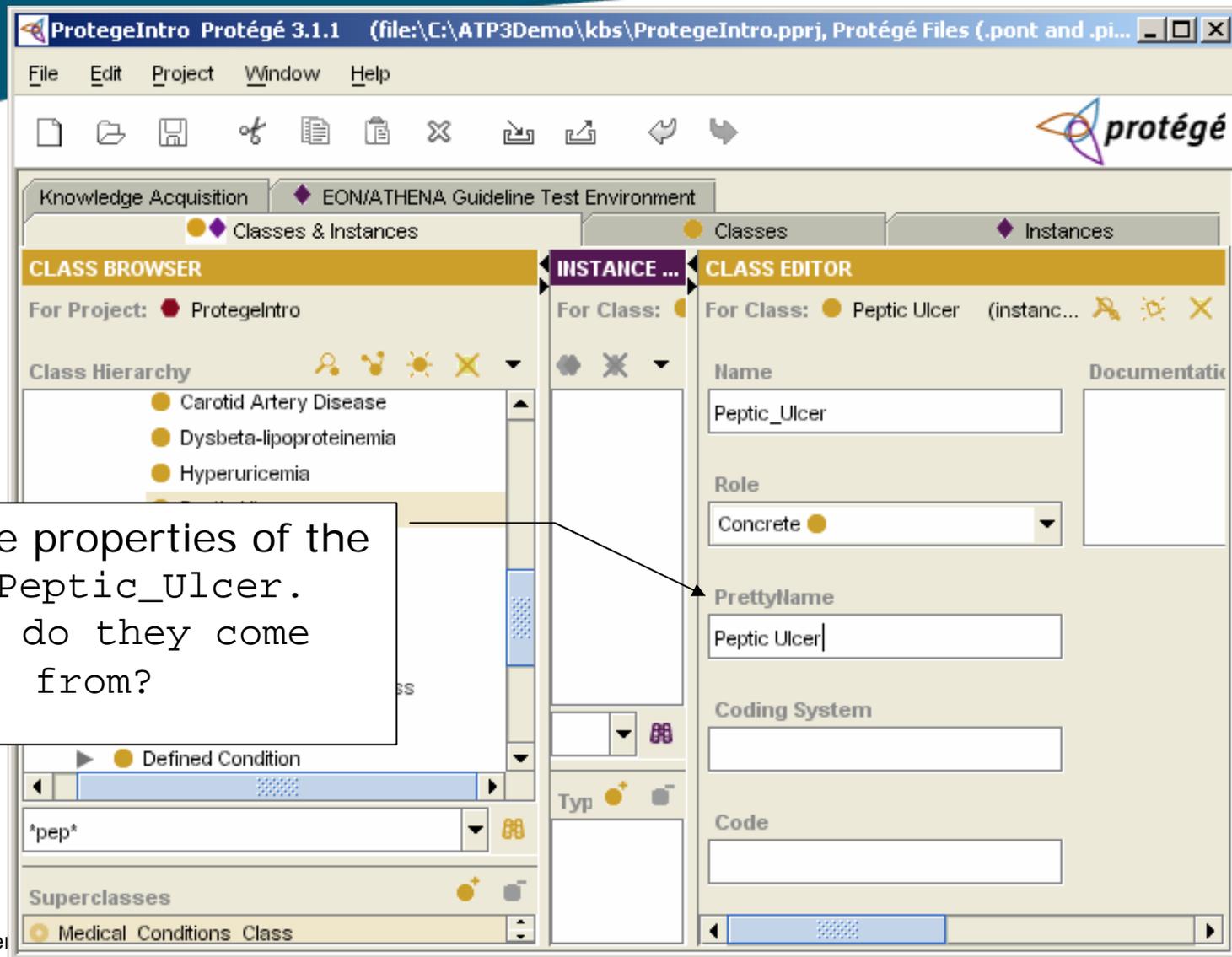


- Create `Peptic_Ulcer` as a subclass of `Medical_Conditions_Class`

Add "Peptic Ulcer" in PrettyName field: You should have



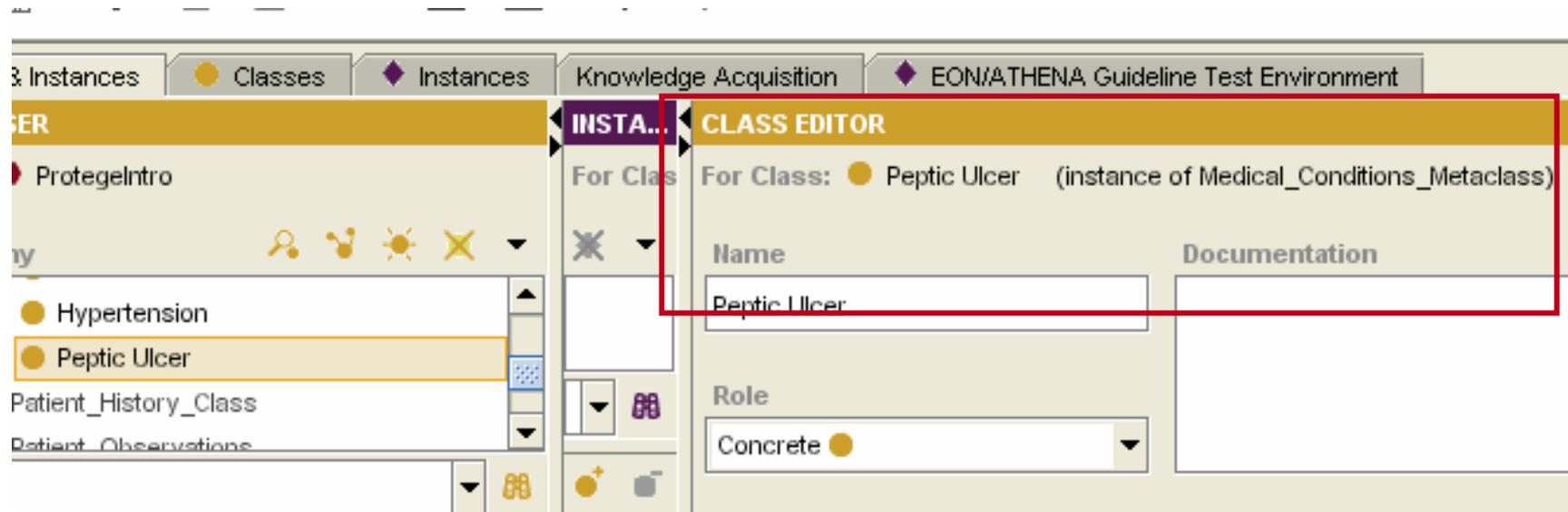
Properties of the Peptic Ulcer class



These are properties of the class Peptic_Ulcer. Where do they come from?

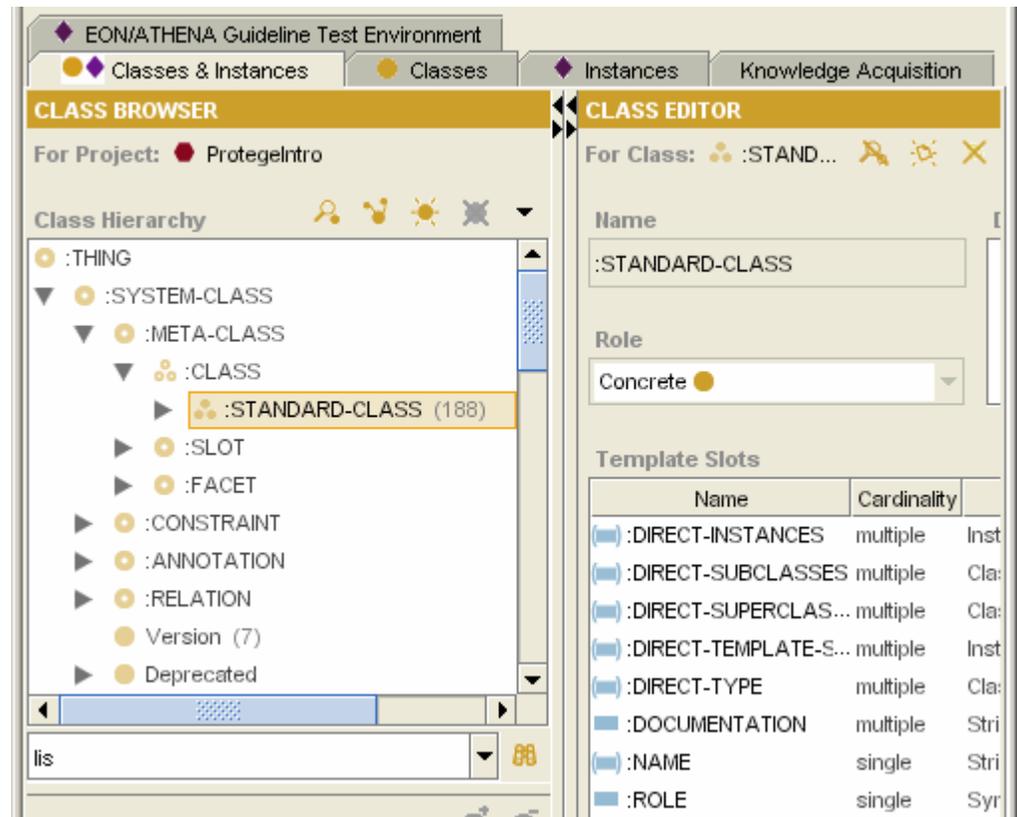
A class defines the properties that its instance may have...

- Therefore, for the `PepticUlcer` class to have properties such as pretty name, coding system, code
- The `PepticUlcer` class is an instance of a *metaclass* in Protégé



Metaclasses: Templates for class definitions

- Metaclasses enable us to add properties to class definitions
- By default, we have:
 - Class name
 - Documentation
 - Slots
 - ...
- Protégé defines a system `:STANDARD-CLASS`
- Other metaclasses are added as subclasses of `:STANDARD-CLASS`



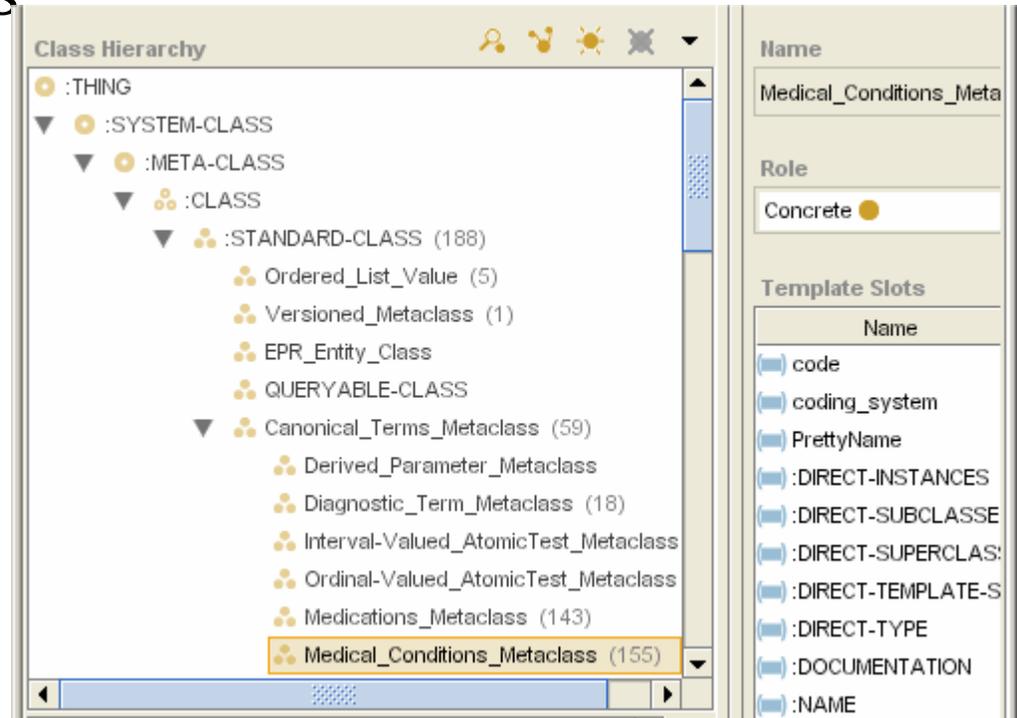
Metaclass example: Medical_Condition_Metaclass

- Additional properties:
 - PrettyName
 - code_system
 - code

CLASS EDITOR

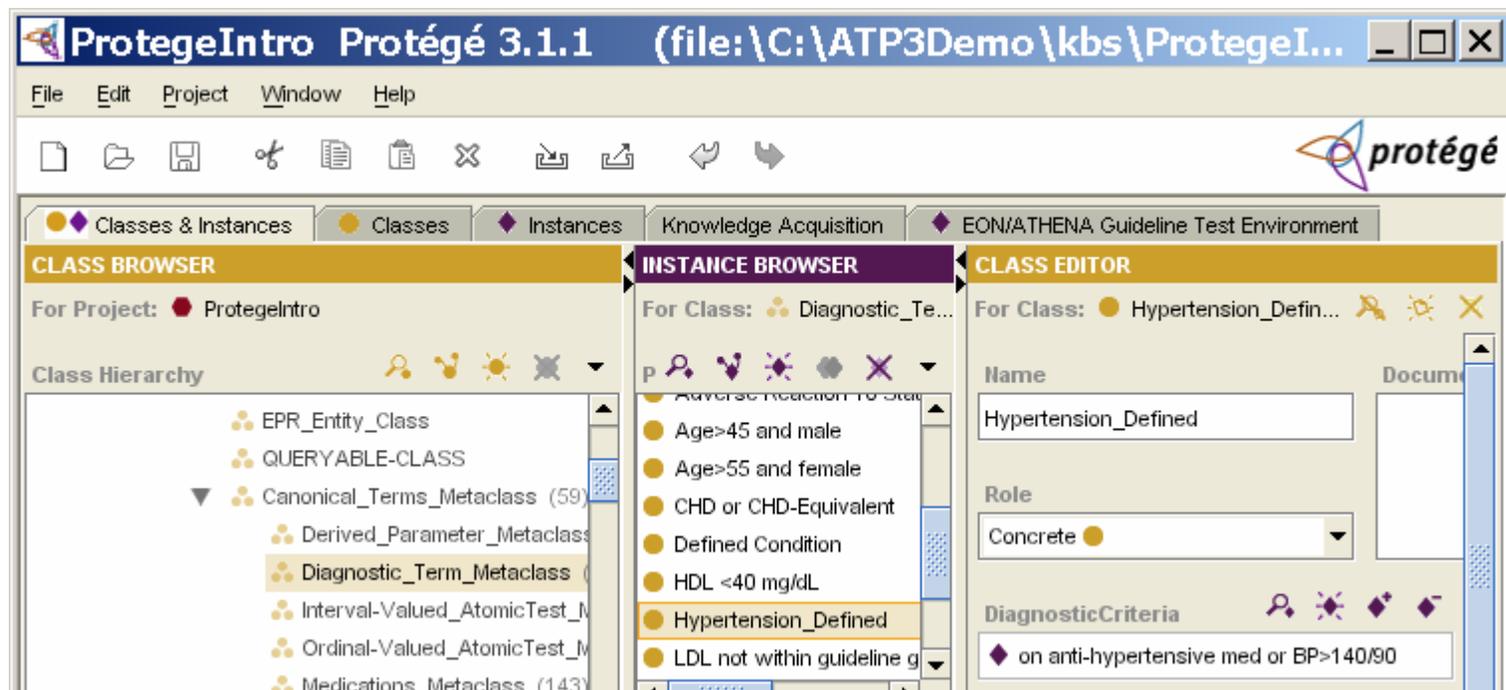
For Class: ● Peptic Ulcer (instance of Medical_Conditions..)

Name	Documentation
<input type="text" value="Peptic Ulcer"/>	
Role	
<input type="text" value="Concrete"/> ●	
PrettyName	
<input type="text"/>	
Coding System	
<input type="text"/>	
Code	
<input type="text"/>	



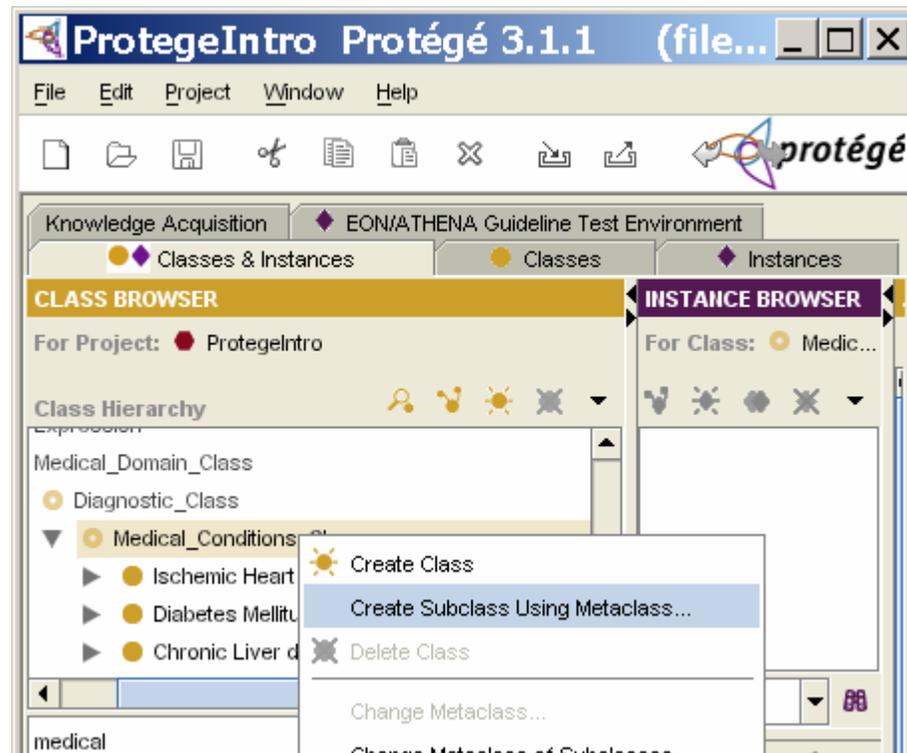
Metaclass example: Diagnostic_Term_Metaclass

- We want to define a concept in terms of an expression: Hypertension_Defined as *on anti-hypertensive med or BP \geq 140/90*



How to create a class of a specific metaclass?

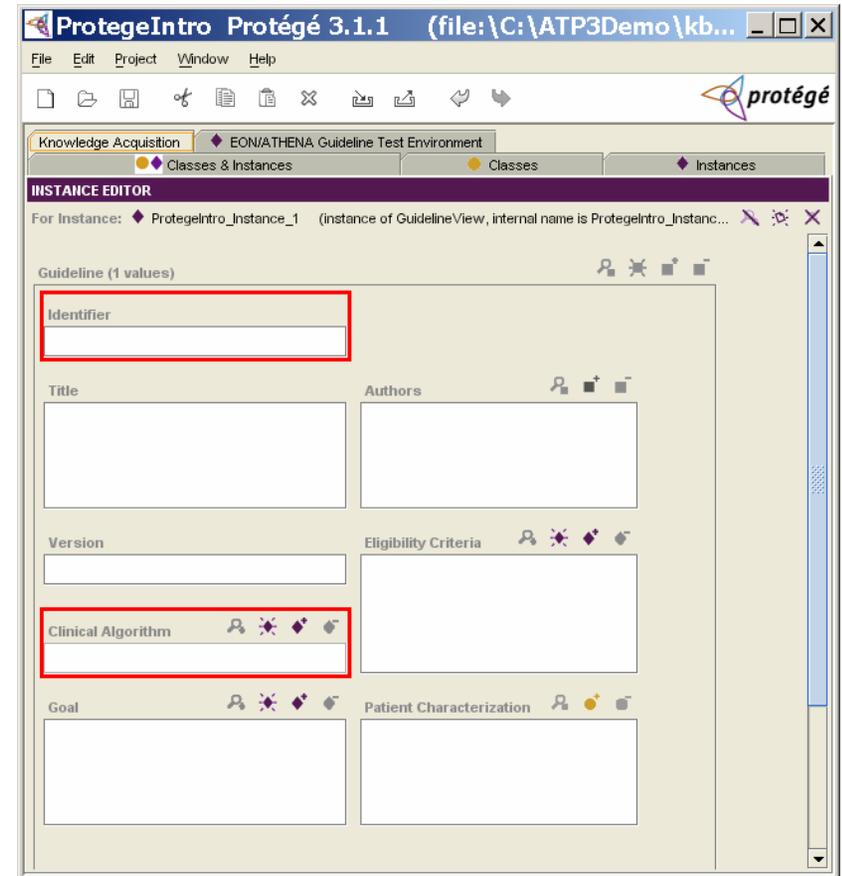
- Right-click on a class to bring up a menu for creating a subclass that's an instance of a specific metaclass



Questions?

Knowledge-Acquisition Tab

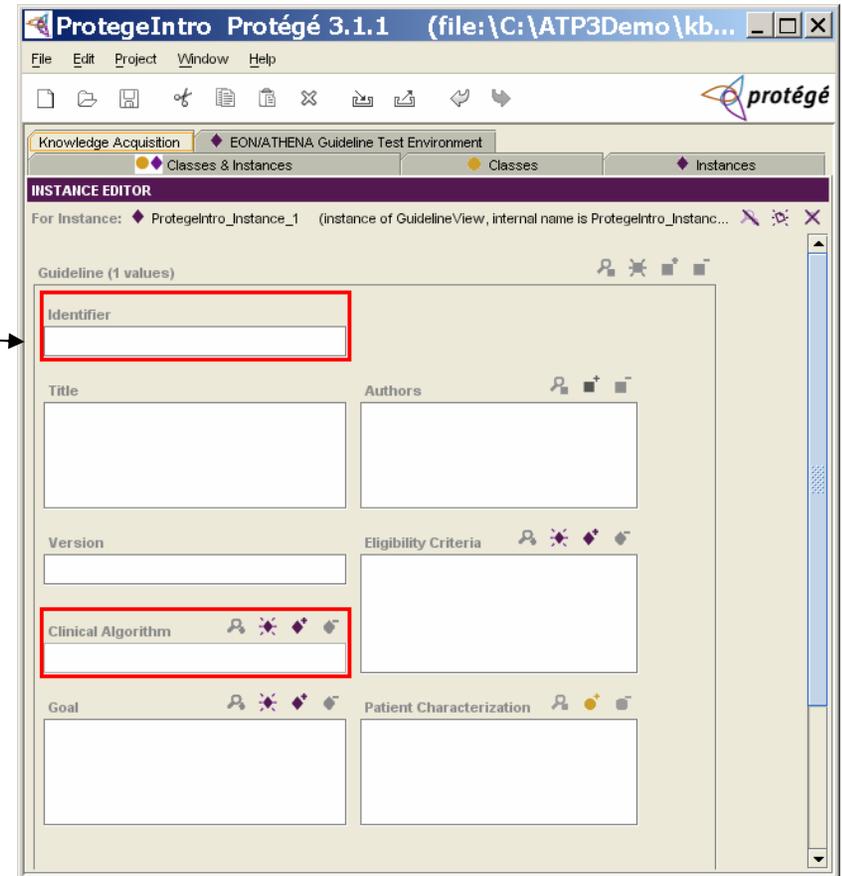
- Use it to work with instances
- View top-level instance
- Create or navigate to other instances



Working with an instance

- Type in a string for the identifier property

Red border means something is not right about this property; Usually it means a value is missing



Creating an instance to be a property value

Classes & Instances Classes Instances

INSTANCE EDITOR

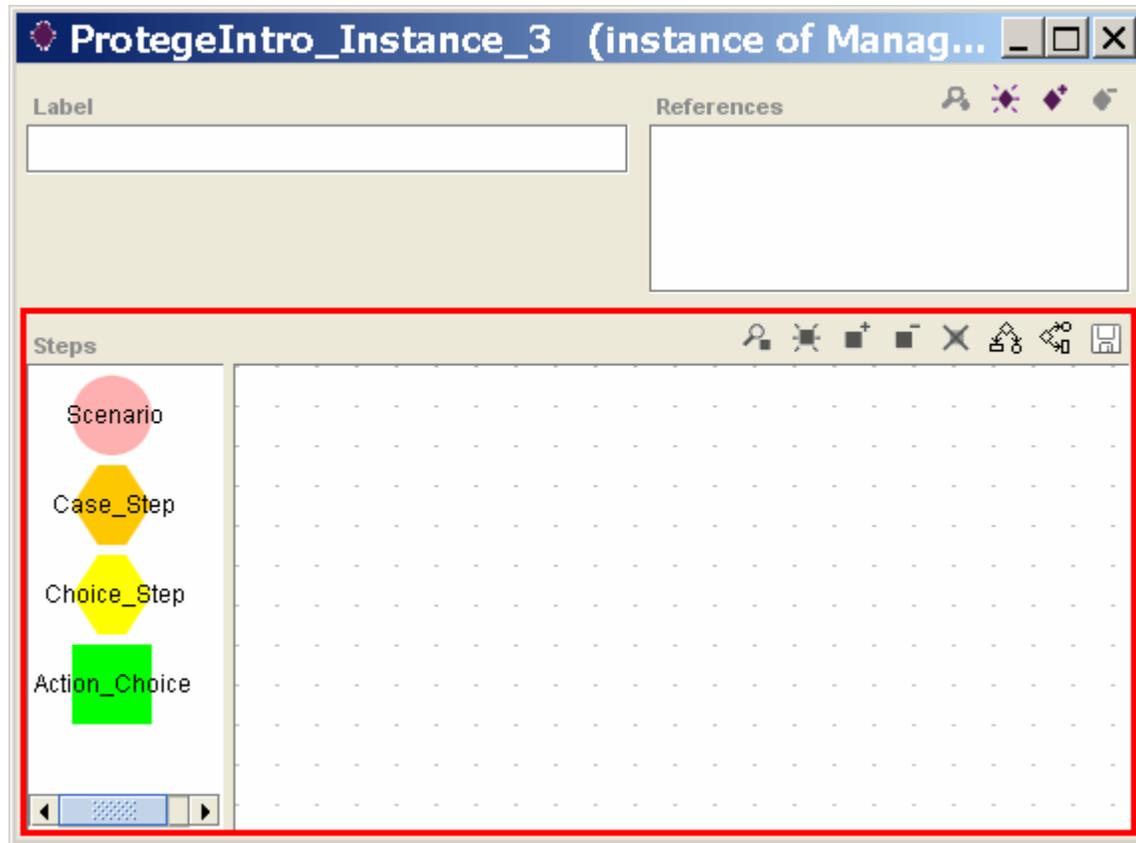
For Instance: ◆ Protegelntro_Instance_1 (instance of Guideline)

Version	Eligibility Crit
Clinical Algorithm	
Goal	Patient Chara

Click on the shiny button

Create Instance

Protégé presents a pre-configure graphical form for adding information about this instance

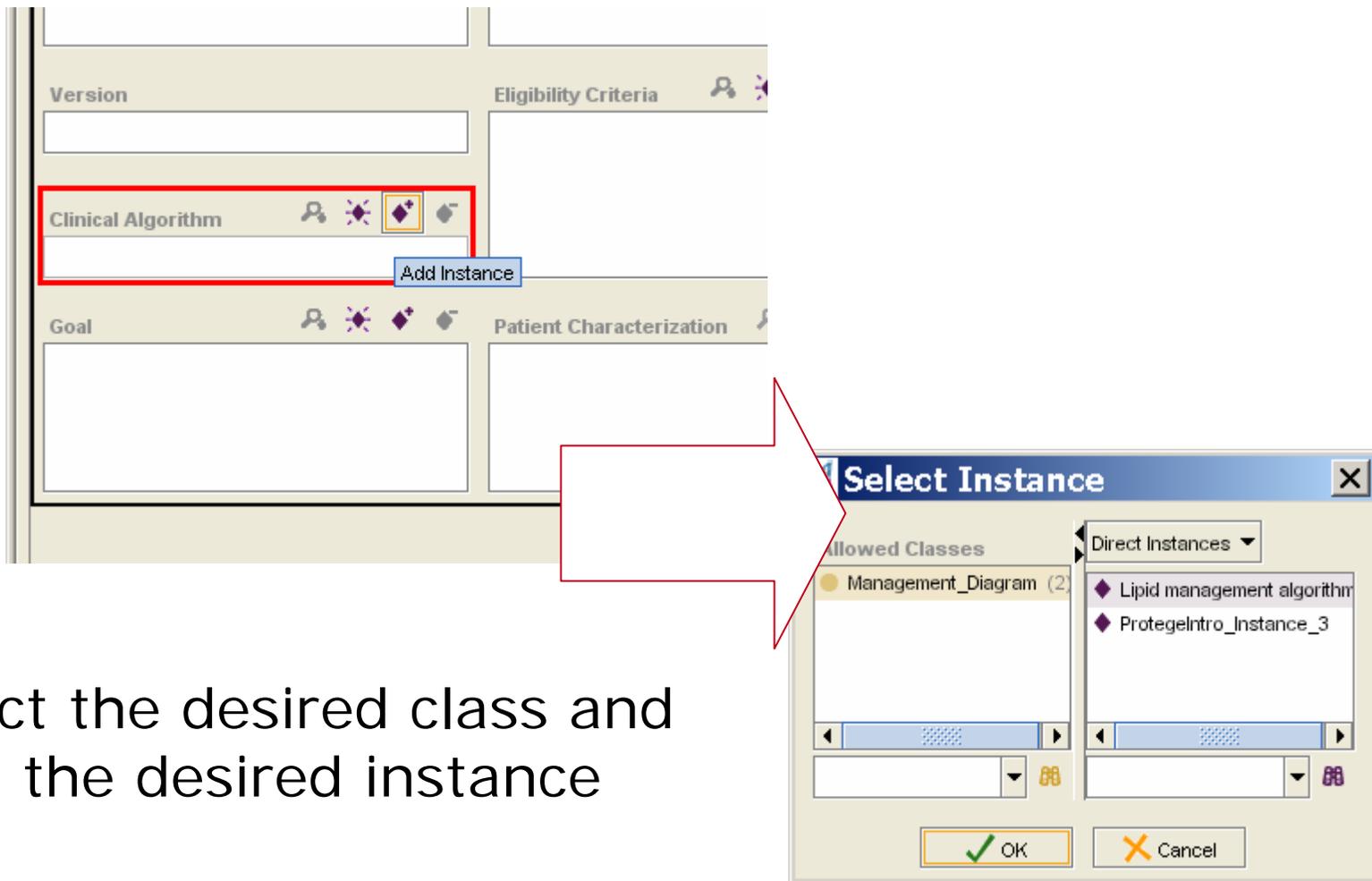


Removing an instance from a property

Use the '-' diamond

The screenshot shows a software interface with a central panel. At the top, a callout box contains the text "Use the '-' diamond". An arrow points from this box to a diamond icon with a minus sign in the "Clinical Algorithm" section. Below this section, a list contains the instance "ProtegeIntro_Instance_3". A blue box labeled "Remove Instance" is positioned over the diamond icon with a minus sign. The interface also shows other sections: "Eligibility Criteria" at the top right and "Goal" at the bottom left, both with search icons. The "Patient Characterization" section is at the bottom right.

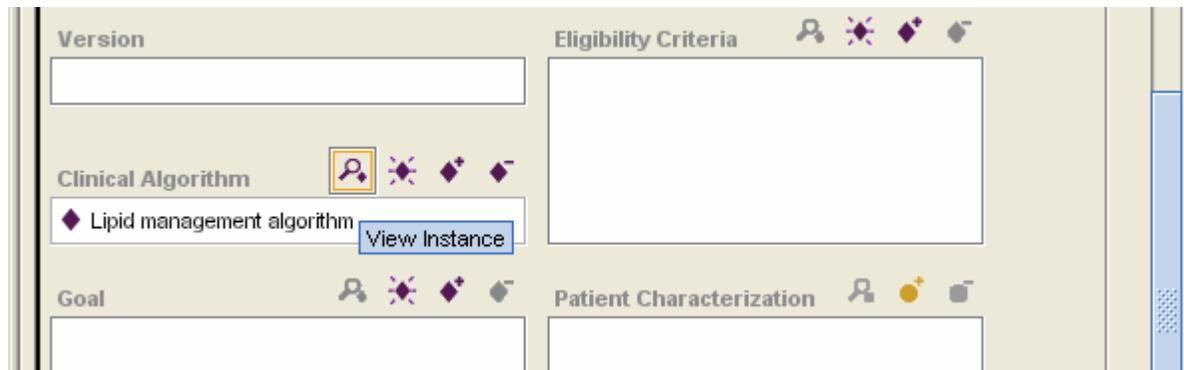
Adding a pre-existing instance as property value



- Select the desired class and then the desired instance

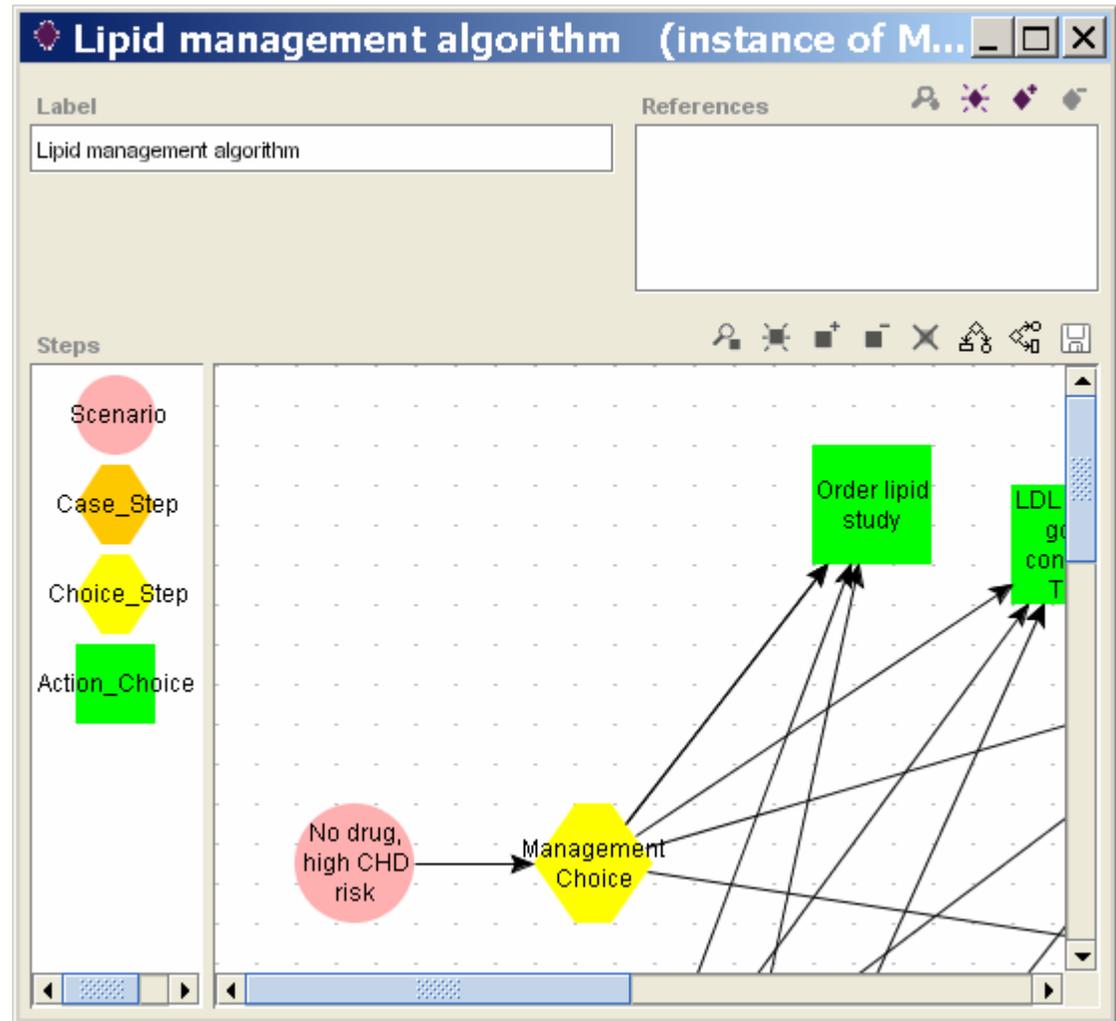
Viewing and editing an instance

- Click on magnifying glass icon

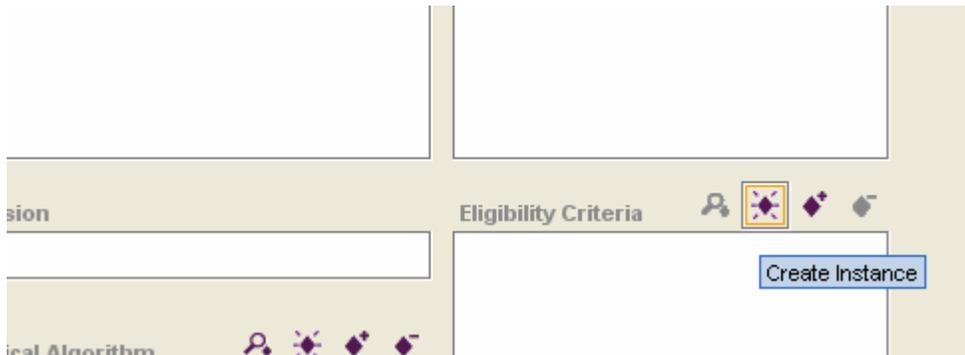


Working with Protégé graph

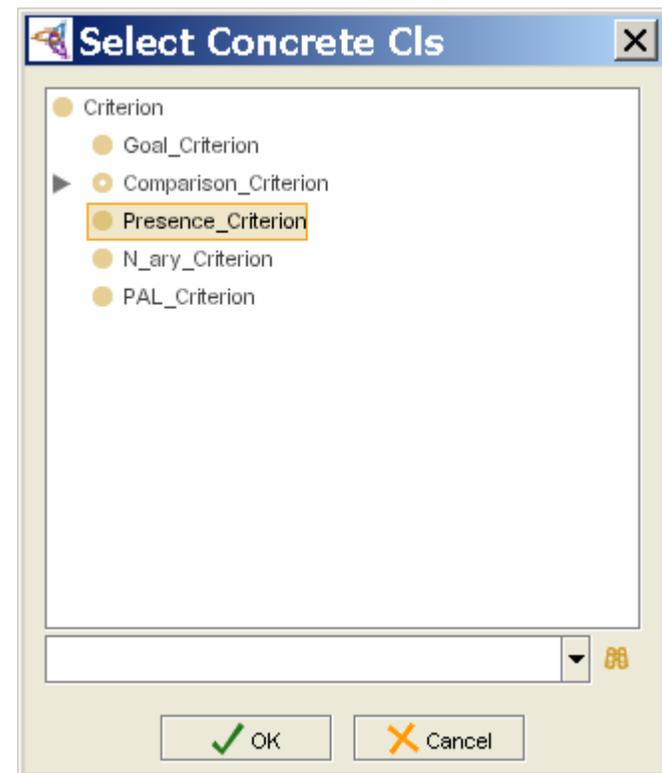
- Drag and drop an icon from palette to create an instance of the class represented by the icon
- Drag and drop arrows from one icon to another to create links



Exercise: Create an instance of eligibility criteria



- Click on “new-instance” button above Eligibility Criteria
- Select Presence_Criterion



Presence Criterion

- Example of statement
 - *Presence of Diabetes Mellitus in Note Entry*

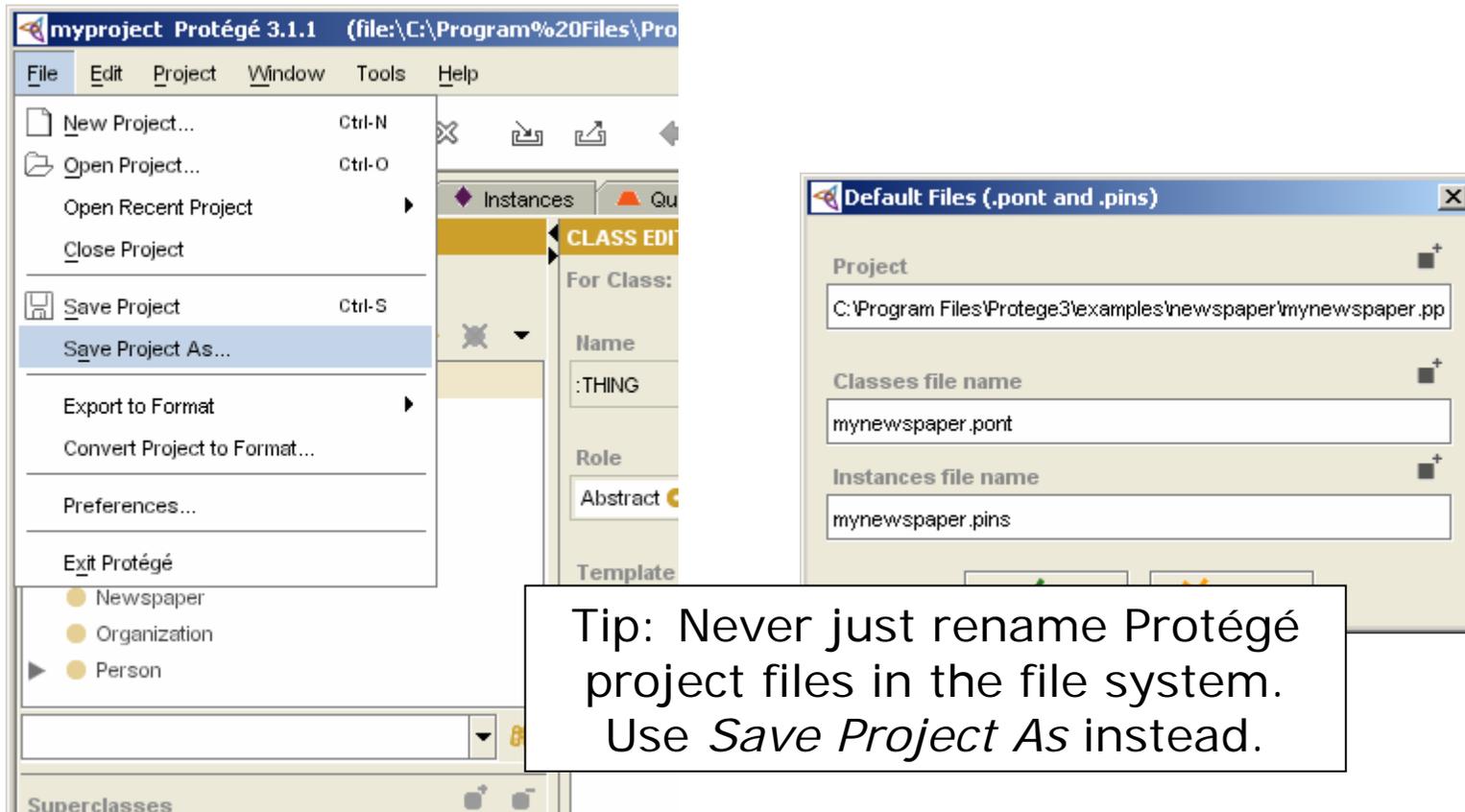
The screenshot shows a window titled "Presence of Diabetes Mellitus in Note E...". The interface is divided into several sections:

- Label:** A text box containing "Presence of Diabetes Mellitus in Note Entry".
- Entry Type:** A dropdown menu with "Note_Entry" selected. To the right are icons for search, add, and remove.
- Presence:** A checkbox labeled "Presence" which is checked.
- Mood:** A text box that is currently empty. To the right are icons for search, add, and remove.
- Domain Term:** A dropdown menu with "Diabetes Mellitus" selected. To the right are icons for search, add, and remove.
- Period:** A text box that is currently empty. To the right are icons for search, add, and remove.

Questions?

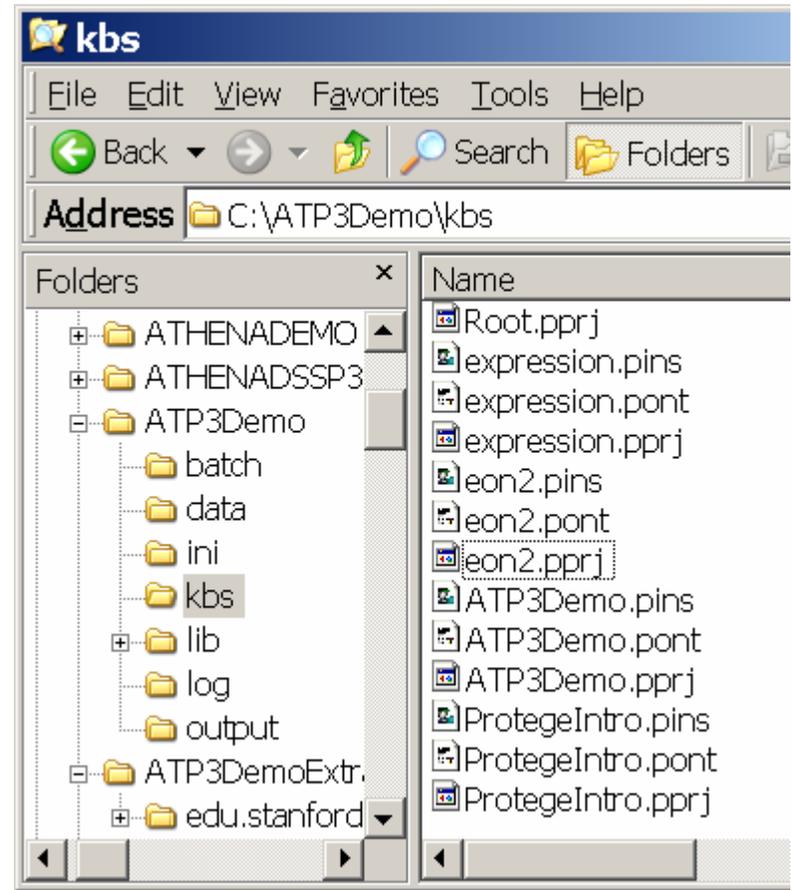
Working with projects

- *Save* and *Save project as*

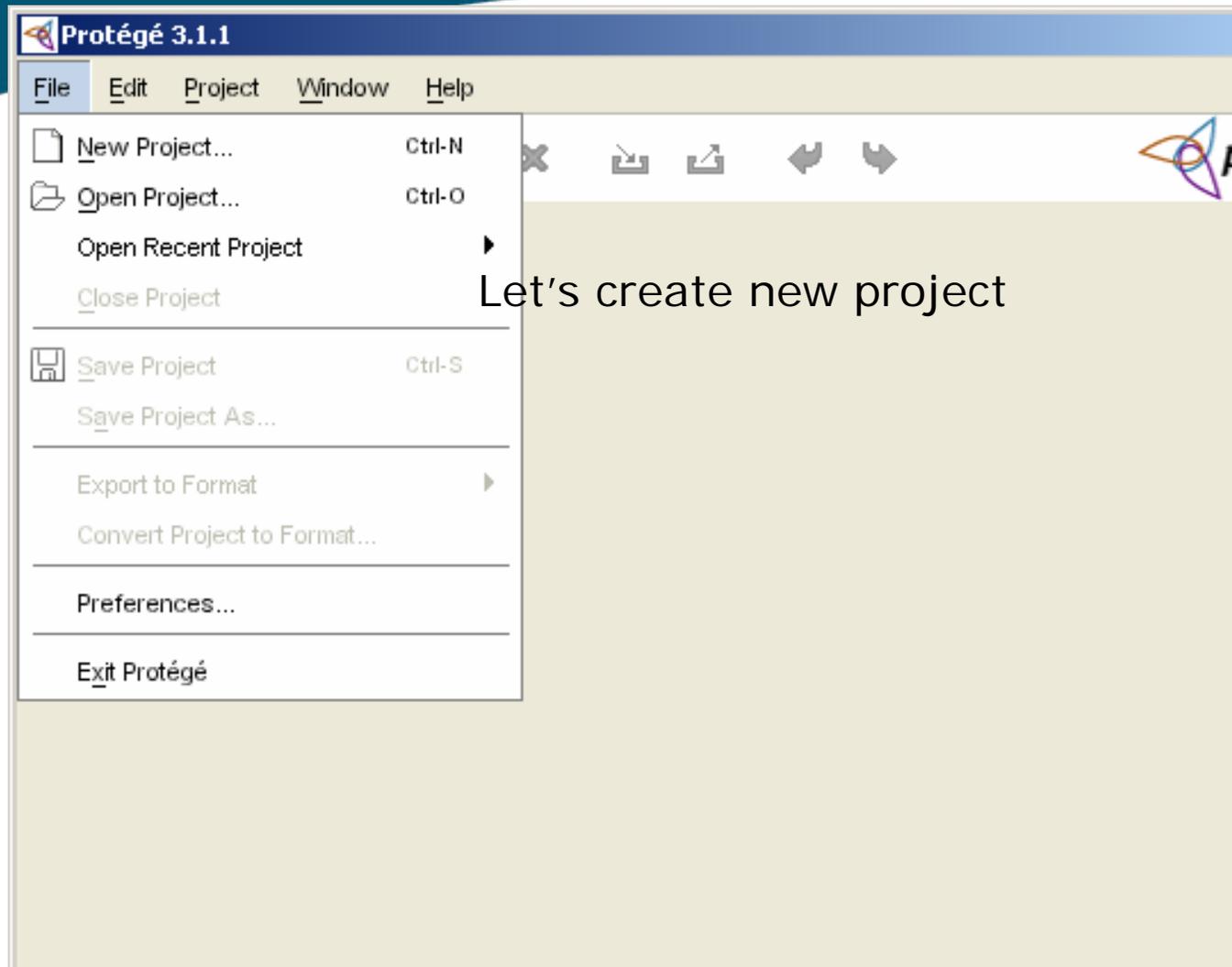


The idea of a “Protégé project”

- A Protégé knowledge bases is organized as a set of related projects
 - class definitions stored in .pont file
 - instances definitions stored in .pins file
 - Project file (.pprj) stores project description and GUI customization information

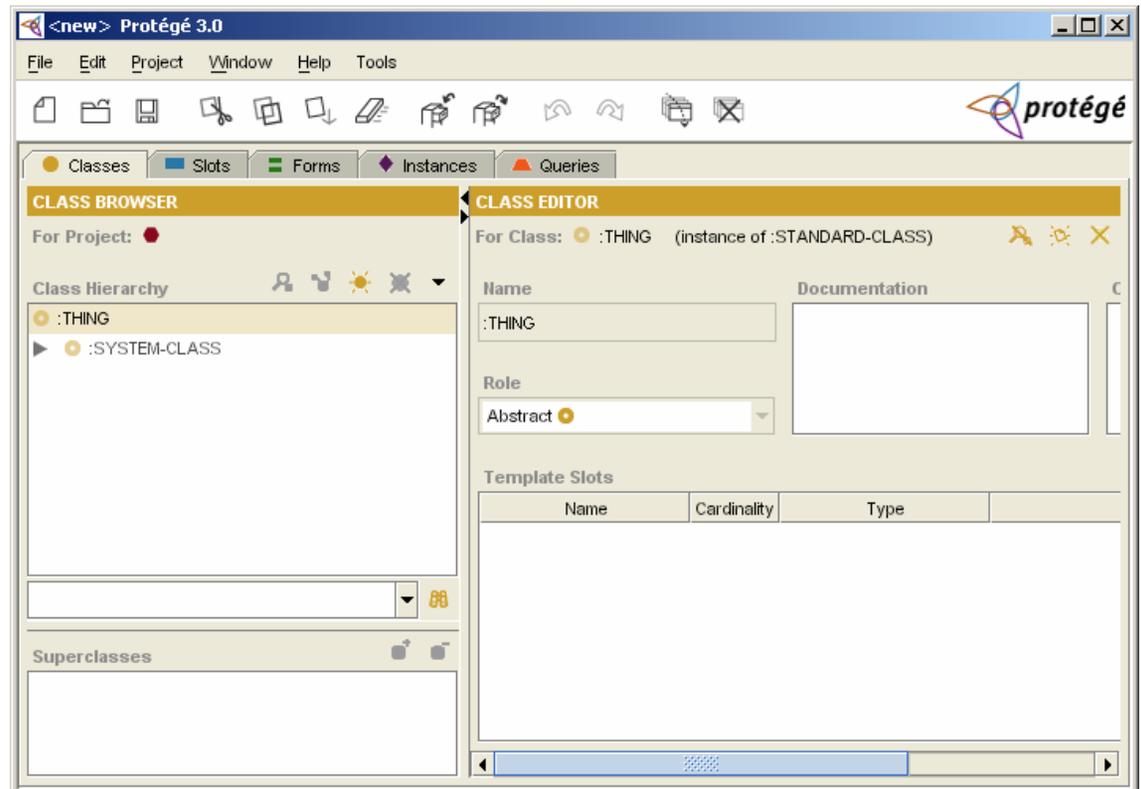
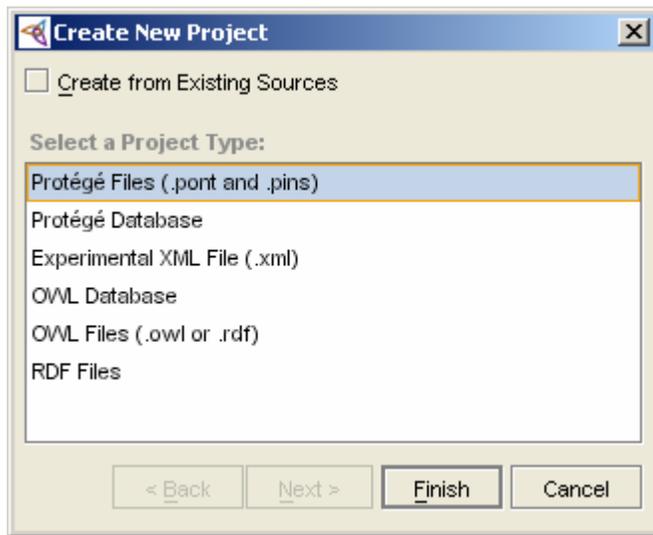


Suppose you want to create a new guideline knowledge base



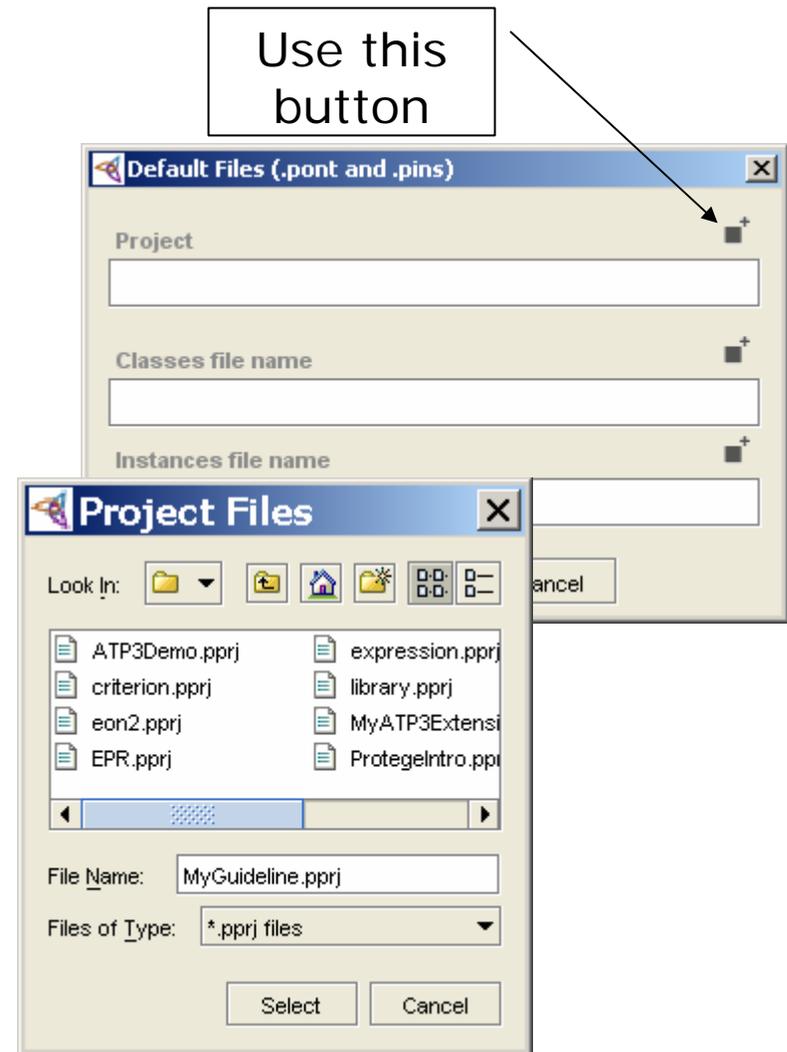
We have a new project

- Select *Protégé Files (.pont and .pins)*



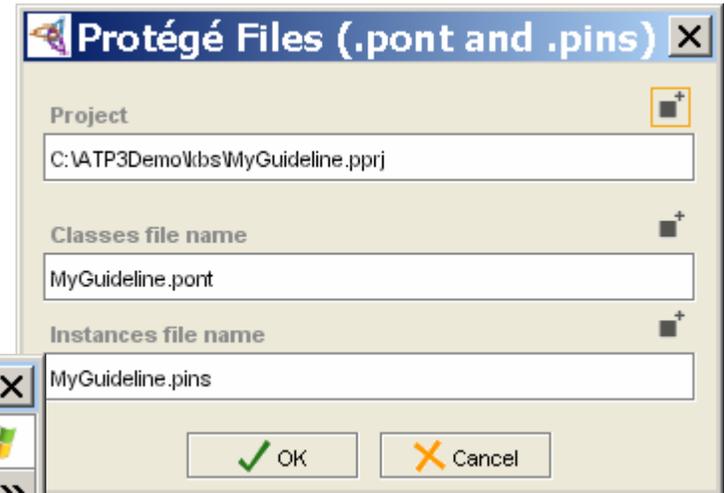
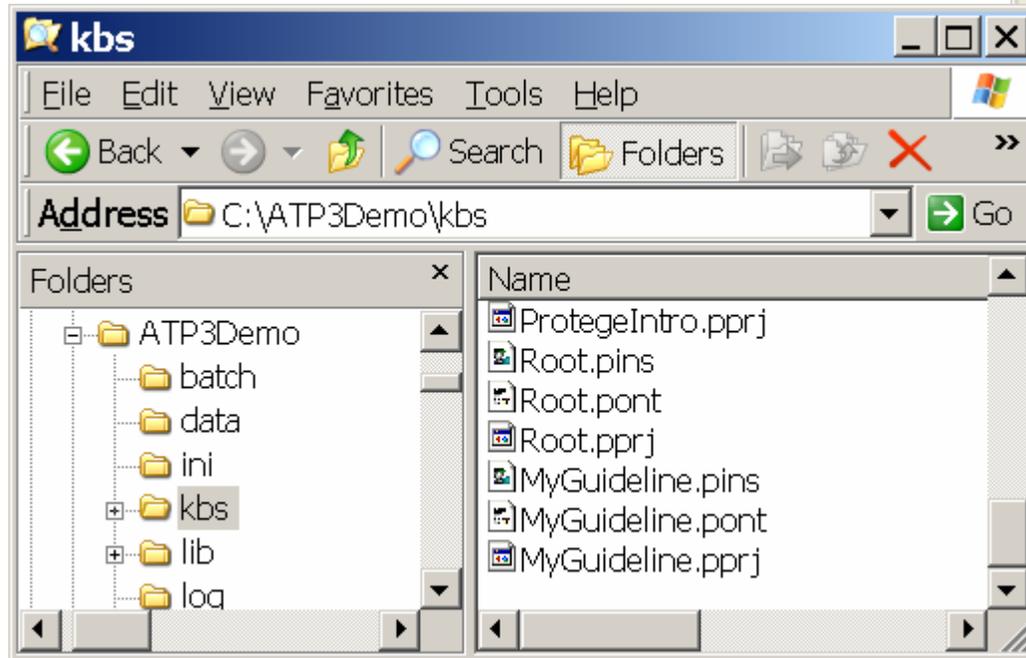
Name the new project myGuideline.pprj and save in the C:\ATP3Demo\kbs directory

- *Save Project* brings up a dialog box
- Navigate to the C:\ATP3Demo\kbs directory



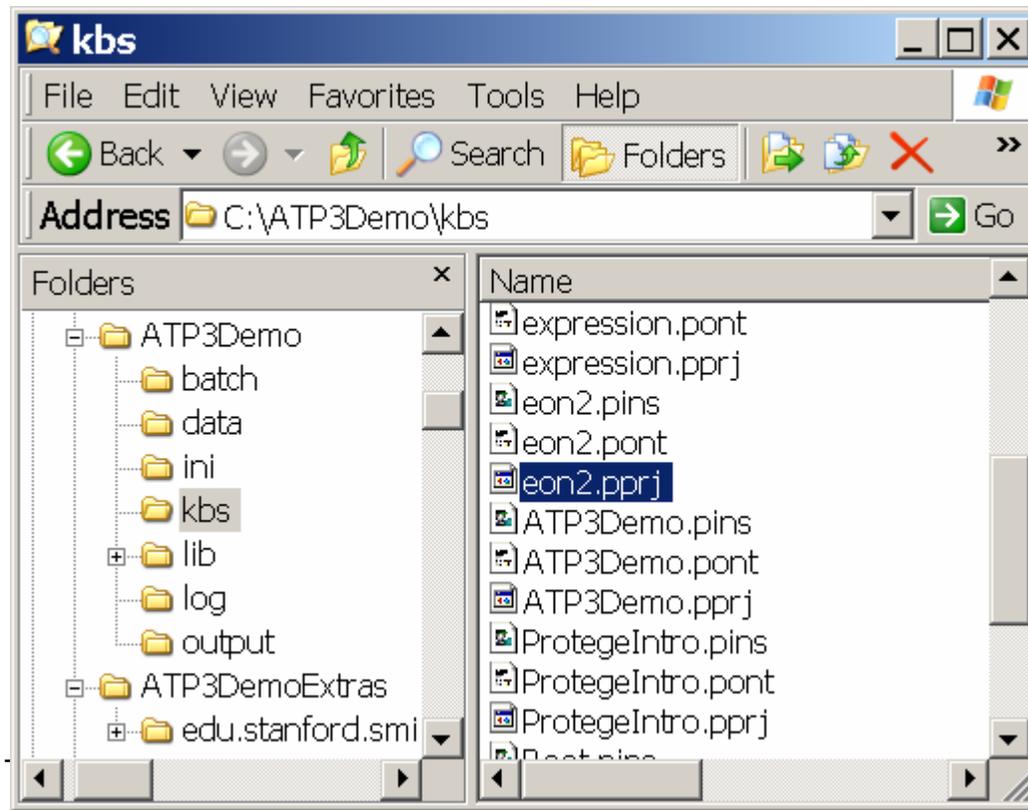
Name the new project myGuidelinen.pprj and save in the C:\ATP3Demo\kbs directory

- Finish saving



We want to re-use the EON guideline model used in ATHENA systems

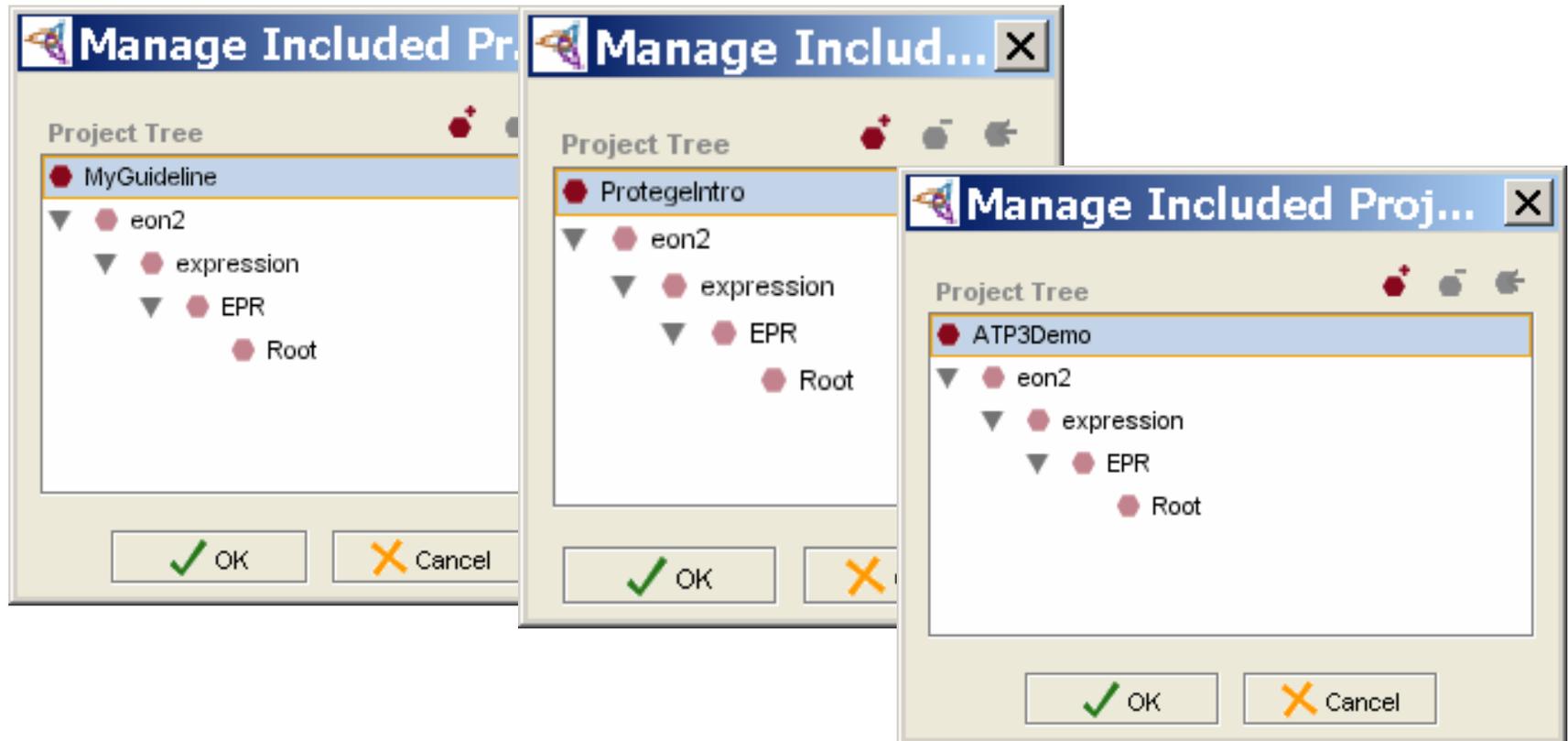
- The generic guideline model is a collection of Protégé projects included in the eon2.pprj project



Project inclusion

- Once eon2 project is included in myGuideline project
 - All classes and instances in the eon2 project are available for use
 - But you can't change them
- Allow reuse of one project in many different projects

MyGuideline, ProtegeIntro, ATP3Demo all share the same underlying eon2 projects



Questions on working with Protégé projects?

Summary

- Protégé provides a graphical user-interface environment for creating knowledge bases
- Information in a Protégé knowledge base is organized around the idea of
 - Classes
 - Instances
 - Properties
- EON provides a set of Protégé projects for creating guideline knowledge bases