

Protégé

- Ontology editor (OWL)
- Knowledge Base Framework (OKBC πρωτόκολλο <http://www.ai.sri.com/~okbc/>)
- Free, open source
(<http://protege.stanford.edu/download/download.html>)
- Java
- Expandable
- Parameterized
- Output (RDF(S), OWL, XML schema)

Protégé-Frames

- Integrated U.I. and Knowledge Server
- Knowledge model compatible with OKBC
 - Ontologies-set of classes
 - Organized in class hierarchies (subsumption)
 - Set of class properties
 - Set of class instances (patterns)

Protégé-Frames

- 4 tabs
 - Classes
 - Slots
 - Forms
 - Instances

Protégé-Frames [classes tab]

- **Ontology Editor**
 - Definitions of classes
 - Hierarchy of classes
 - Properties, properties-restrictions
 - Relations between classes
 - Properties of relations

Protégé-Frames [classes tab]

The screenshot displays the Protégé 3.1 beta interface with the 'Classes' tab selected. The window title is 'newspaper Protégé 3.1 beta'. The menu bar includes 'File', 'Edit', 'Project', 'Window', and 'Help'. The toolbar contains various icons for file operations and editing. The 'CLASS BROWSER' on the left shows the class hierarchy for the project 'newspaper'. The 'CLASS EDITOR' on the right shows the details for the 'Editor' class, including its name, role, and a table of template slots.

CLASS BROWSER
For Project: newspaper

Class Hierarchy

- :THING
 - :SYSTEM-CLASS
 - Author
 - News_Service
 - Columnist
 - Editor**
 - Reporter
 - Content
 - Layout_info
 - Library
 - Newspaper
 - Organization

Superclasses

- Author
- Employee

CLASS EDITOR
For Class: Editor (instance of :STANDARD-CLASS)

Name
Editor

Documentation
Editors are responsible for the content of sections.

Role
Concrete

Template Slots

Name	Cardinality	Type
current_job_title	single	String
date_hired	single	String
name	single	String
other_information	single	String
phone_number	single	String
responsible_for	multiple	Instance of Employee
salary	single	Float
sections	multiple	Instance of Section

Protégé-Frames [slots tab]

The screenshot displays the Protégé 3.1.1 interface with the 'Slots' tab selected. The window title is 'tutorial Protégé 3.1.1 (file:\C:\Program%20Files\Protege_3.1\examples\tutorial\tutorial.pprj, Protégé Files (.pont...'. The menu bar includes 'File', 'Edit', 'Project', 'Window', and 'Help'. The toolbar contains various icons for file operations and editing. The 'Classes' tab is active, showing a 'Slot Hierarchy' on the left and a 'SLOT EDITOR' on the right.

SLOT BROWSER
For Project: tutorial

Slot Hierarchy

- name
- :ANNOTATED-INSTANCE
- :ANNOTATION-TEXT
- :ASSOCIATED-FACET ↔ :ASSOCIATED-SLOT
- :ASSOCIATED-SLOT ↔ :ASSOCIATED-FACET
- :CREATION-TIMESTAMP
- :CREATOR
- :DIRECT-DOMAIN ↔ :DIRECT-TEMPLATE-SLOT
- :DIRECT-INSTANCES ↔ :DIRECT-TYPE
- :DIRECT-SUBCLASSES ↔ :DIRECT-SUPERCLASSES
- :DIRECT-SUBSLOTS ↔ :DIRECT-SUPERSLOTS
- :DIRECT-SUPERCLASSES ↔ :DIRECT-SUBCLASSES
- :DIRECT-SUPERSLOTS ↔ :DIRECT-SUBSLOTS
- :DIRECT-TEMPLATE-SLOTS ↔ :DIRECT-DOMAINS
- :DIRECT-TYPE ↔ :DIRECT-INSTANCES
- :DOCUMENTATION
- :FROM
- :MODIFICATION-TIMESTAMP
- :MODIFIER
- :NAME
- :PAL-DESCRIPTION
- :PAL-NAME
- :PAL-RANGE

SLOT EDITOR
For Slot: name (instance of :STANDARD-SLOT)

Name
name

Documentation

Value Type
String

Cardinality
 required at least
 multiple at most

Minimum **Maximum**

Inverse Slot

Template Values

Default Values

Domain

Superslots

Protégé-Frames [forms tab]

- Default form for creation of properties
- Different type – different form
- Capability for change of place, size, label etc of the forms

Protégé-Frames [forms tab]

The screenshot displays the Protégé 3.1 beta interface for a project named "newspaper". The main window title is "newspaper Protégé 3.1 beta (file:\C:\Temp\protege%203.1%20beta%20-%20164\examples\newspaper...)". The menu bar includes File, Edit, Project, Window, and Help. The toolbar contains icons for file operations and navigation. The interface is divided into several panels:

- Classes, Slots, Forms, Instances, Queries:** A row of tabs at the top of the workspace.
- FORM BROWSER:** Located on the left, it shows a tree view of classes for the "newspaper" project. The "Editor" class is selected and highlighted in green. Other classes include :THING, :SYSTEM-CLASS, Author (with sub-classes News_Service, Columnist, Reporter), Content, Layout_info, Library, Newspaper, Organization, and Person.
- FORM EDITOR:** The main workspace, currently showing the "Editor" class. It includes:
 - Display Slot:** A dropdown menu set to "name".
 - Widget:** A dropdown menu with a list of widget options: <none>, YellowStickyWidget, TextAreaWidget, **TextFieldWidget** (highlighted), UniqueStringWidget, URLWidget, and ImageWidget.
 - Form Fields:** Several input fields are visible, including "Name", "Date Hired", "Current Job Title" (highlighted with a green border), "Phone Number", and "Other Information".
 - Sections:** A panel on the right side of the form editor.

Protégé-Frames [instances tab]

Tool for view/modification of instances of classes of the ontology

Protégé-Frames [instances tab]

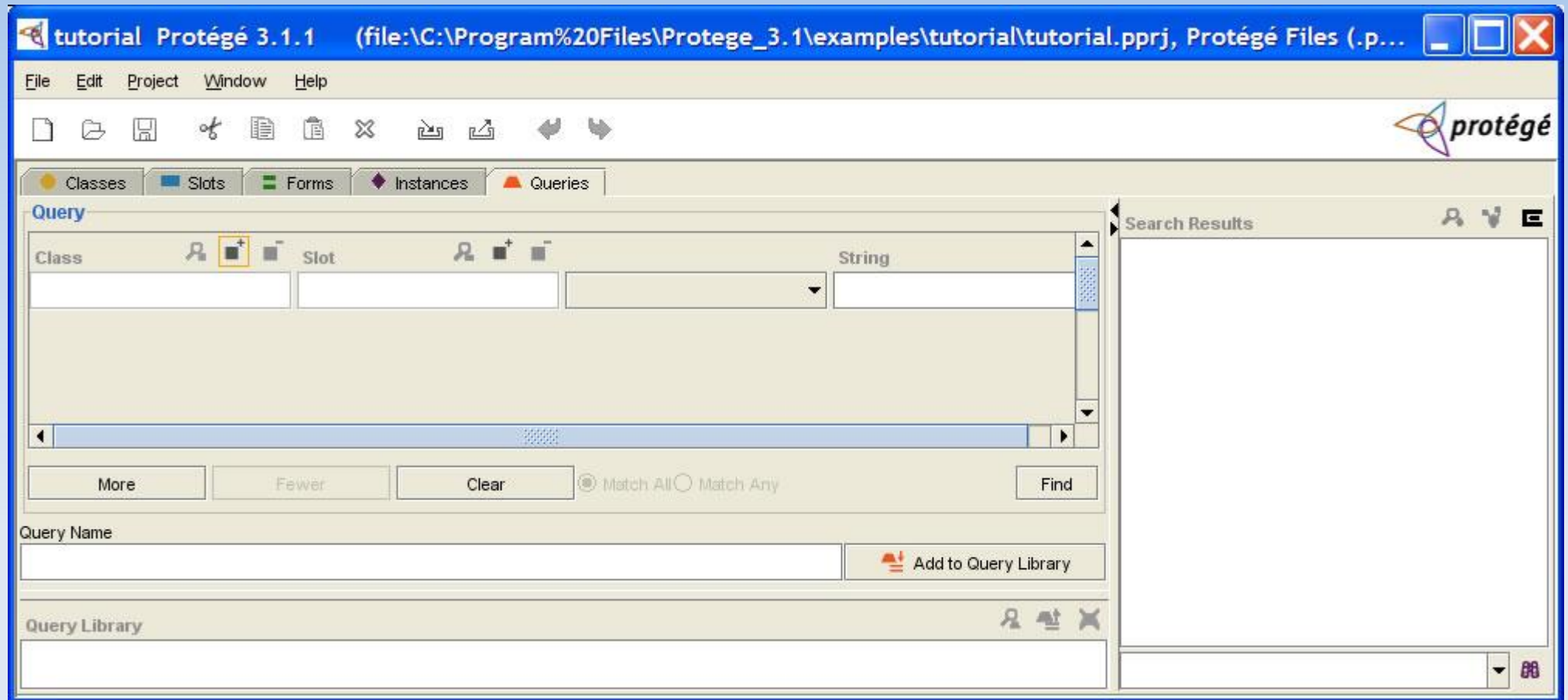
The screenshot displays the Protégé 3.1 beta software interface. The title bar reads "newspaper Protégé 3.1 beta (file:\C:\Temp\protege%203.1%20beta%20-%20164\examples\newspaper\newspa...". The menu bar includes File, Edit, Project, Window, and Help. The toolbar contains various icons for file operations and editing. The main interface is divided into several panes:

- Classes:** Shows a class hierarchy for the project "newspaper". The hierarchy includes: :THING, :SYSTEM-CLASS, Author (with sub-classes News_Service (2), Columnist (2), Editor (4), Reporter (3)), Content, Layout_info, Library (1), Newspaper (6), Organization (1), and Person.
- Instances:** Shows the instance browser for the class "Editor". The instances listed are: Chief Honcho, Mr. Science, Ms Gardiner, and Sports Nut.
- Instance Editor:** Shows the editor for the instance "Chief Honcho (instance of Editor, inte...". The editor contains several fields:
 - Name:** Chief Honcho
 - Salary:** 150000.0
 - Date Hired:** (empty field)
 - Current Job Title:** (empty field)
 - Phone Number:** (empty field)
 - Other Information:** (empty text area)
 - Responsible For:** Sports Nut, Ms Gardiner
 - Sections:** Magazine, Local News, Automotive, Business, World News

Protégé-Frames [queries tab]

- Queries in project
 - Choice of one or more classes
 - One or more properties
 - One or more criteria
- Storage for queries

Protégé-Frames [queries tab]



Protégé-OWL

- OWL (web ontology language) Editor
- Load – Save ontologies written in RDF OWL
- Visualization of classes, properties and SWRL rules (combination of OWL and RuleML)
- Definition of OWL expressions
- Offers a Java API for U.I. development

Protégé-OWL

- Choices are:
 - Owl Classes
 - Properties
 - Forms
 - Individuals

Protégé-OWL [OWLClasses]

- Viewing and modifying hierarchies
- Details of each class at the right side
- Adding comments, labels etc
- Restrictions
- Classification

Protégé-OWL [OWLClasses]

The screenshot displays the Protégé 3.1 interface for editing an OWL class. The main window title is "travel Protégé 3.1 (file:K:\protege-owl\owl\travel.pprj, OWL Files (.owl or .rdf))". The menu bar includes File, Edit, Project, OWL, Code, Window, Tools, and Help. The toolbar contains various icons for file operations and editing. The interface is divided into several panes:

- Subclass Relationship:** Shows the asserted hierarchy for the project "travel". The hierarchy is as follows:
 - owl:Thing
 - Accommodation
 - BedAndBreakfast
 - BudgetAccommodation
 - Campground
 - Hotel
 - AccommodationRating
 - Activity
 - Contact
 - Destination
 - BackpackersDestination
 - Beach
 - BudgetHotelDestination
 - FamilyDestination** (selected)
 - QuietDestination
 - RetireeDestination
 - RuralArea
 - UrbanArea
 - City
 - Town

- Class Editor:** For Class: FamilyDestination (instance of owl:Class). It has tabs for Name, SameAs, and DifferentFrom. The Name tab is active, showing the class name "FamilyDestination". Below it, the `rdfs:comment` is "A destination with at least one accommodation and at least 2 activities." To the right, the Annotations table shows:

Property	Value	Lang
rdfs:comment	A destination with at le...	
- Asserted Conditions:** Shows conditions for the class. Under the "NECESSARY & SUFFICIENT" section, there are three conditions:
- Destination
- hasAccommodation ≥ 1
- hasActivity ≥ 2Under the "NECESSARY" section, there are no conditions.
- Properties:** Shows the class's properties:
- hasAccommodation (multiple Accommodation) with cardinality ≥ 1
- hasActivity (multiple Activity) with cardinality ≥ 2
- hasPart (multiple Destination)
- Disjoints:** Shows the class is disjoint with RetireeDestination.

At the bottom right, there are radio buttons for "Logic View" (selected) and "Properties View".

classification

- Ontology consistency checking
- Checking undesirable relations between classes
- Automatic classification of instances into classes

Protégé-OWL [OWLClasses]

travel Protégé 3.1 (file:C:\protege-owl\owl\travel.pprj, OWL Files (.owl or .rdf))

File Edit Project OWL Code Window Tools Help

OWLClasses Properties Forms Individuals Metadata

SUBCLASS RELATIONSHIP For Project: ● travel

Asserted Hierarchy

- Contact
- ▼ ● Destination
 - BackpackersDestination
 - Beach
 - BudgetHotelDestination
 - FamilyDestination
 - QuietDestination
 - RetireeDestination
- ▶ ● RuralArea
- ▼ ● UrbanArea
 - ▼ ● City
 - Capital
 - Town

SUBCLASS RELATIONSHIP For Project: ● travel

Inferred Hierarchy

- Destination
 - ▶ ● BackpackersDestination
 - Beach
 - BudgetHotelDestination
 - FamilyDestination
 - ▼ ● QuietDestination
 - RetireeDestination
 - Capital
 - ▶ ● RuralArea
 - ▼ ● UrbanArea
 - ▼ ● City
 - Capital
 - Town

CLASS EDITOR For Class: ● Capital (instance of owl:Class)

Name SameAs DifferentFrom

Capital

rdfs:comment

Asserted Inferred

Asserted Conditions

- NECESSARY & SUFFICIENT
- NECESSARY
- City

Logic View Properties View

Class	Changed superclasses
● Campground	Moved from Accommodation to BudgetAccommodation
● Capital	Added RetireeDestination
● NationalPark	Added BackpackersDestination
● RetireeDestination	Moved from Destination to QuietDestination
● Safari	Inconsistent

Classification Results

Protégé-OWL [Properties]

The screenshot displays the Protégé 3.1 interface for editing an OWL property. The window title is "travel Protégé 3.1 (file:IC:\protege-owl\owl\travel.pprj, OWL Files (.owl or .rdf))". The menu bar includes File, Edit, Project, OWL, Code, Window, Tools, and Help. The toolbar contains various icons for file operations and editing. The main interface is divided into several panes:

- PROPERTY BROWSER:** Shows a list of properties for the project "travel". The selected property is "hasAccommodation". Other properties include "hasActivity ↔ isOfferedAt", "hasCity", "hasContact", "hasEMail", "hasPart", "hasRating", "hasStreet", "hasZipCode", and "isOfferedAt ↔ hasActivity".
- PROPERTY EDITOR:** Shows the configuration for the "hasAccommodation" property (instance of owl:ObjectProperty). It includes tabs for Name, Equivalents, SameAs, and DifferentFrom. The Name field contains "hasAccommodation". There is an "rdfs:comment" field with a search icon.
- Annotations:** A table with columns for Property, Value, and Lang.
- Domain and Range:** The Domain is set to "Destination" and the Range is set to "Accommodation".
- Property Characteristics:** Checkboxes for Functional, InverseFunctional, Symmetric, and Transitive. The "Inverse" checkbox is checked.

Protégé-OWL [individuals]

- View/modification of instances
- Automatic creation of forms depending on the definition of a class and its properties
- `xds:string` – text field

Protégé-OWL [individuals]

The screenshot displays the Protégé 3.1 interface for editing an OWL ontology. The window title is "travel Protégé 3.1 (file:C:\protege-owl\owl\travel.pprj, OWL Files (.owl or .rdf))". The menu bar includes File, Edit, Project, OWL, Code, Window, Tools, and Help. The toolbar contains various icons for file operations and editing. The main interface is divided into several panes:

- CLASS BROWSER:** Shows the class hierarchy for the project "travel". The hierarchy starts with `owl:Thing` and includes classes like `Accommodation`, `Activity`, `Destination`, `RuralArea`, and `UrbanArea`. The `Capital` class is highlighted.
- INSTANCE BROWSER:** Shows instances for the selected class `Capital`. The instances listed are `Canberra` and `Sydney`.
- INDIVIDUAL EDITOR:** Shows the editor for the individual `Sydney` (instance of `Capital`). It includes fields for Name, SameAs, DifferentFrom, and Annotations. The `hasAccommodation` property is set to `FourSeasons`, and the `hasPart` property is set to `BondiBeach` and `CurrawongBeach`.

The interface also features a status bar at the bottom with icons for saving, undo, and redo.

Protégé-OWL – RDF Schema

The screenshot displays the Protégé 3.1 interface for editing an RDF Schema. The main window is titled "travel-rdf Protégé 3.1 (file:\C:\protege-owl\owl\travel-rdf.pprj, OWL Files (.owl or .rdf))". The interface is divided into several panes:

- Subclass Relationship:** Shows the project "travel-rdf" and the "Asserted Hierarchy" tree. The hierarchy is as follows:
 - owl:Thing
 - Accommodation
 - AccommodationRating
 - Activity
 - Contact
 - Destination
 - BackpackersDestination
 - Beach
 - BudgetHotelDestination
 - FamilyDestination
 - QuietDestination
 - RetireeDestination
 - RuralArea
 - Farmland
 - NationalPark
 - UrbanArea
 - City
 - Capital
 - Town

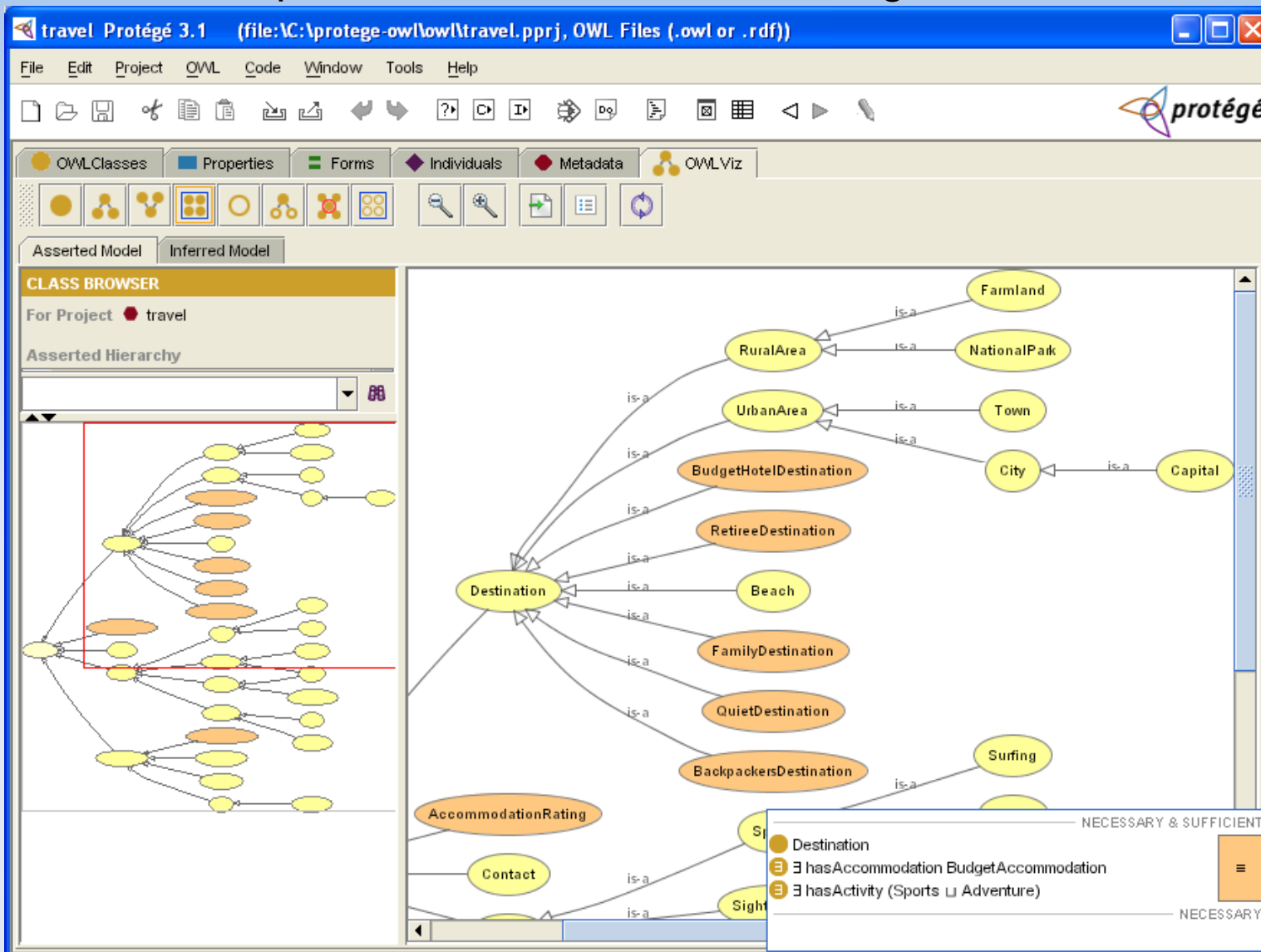
- Class Editor:** Shows the class "BackpackersDestination" (instance of rdfs:Class). It includes:
- Name:** BackpackersDestination
- rdfs:comment:** A destination that provides budget accommodation and offers sport or adventure activities.
- Annotations:** A table with one entry:

Property	Value	Lang
rdfs:comment	A destination that prov...	
- rdfs:subClassOf:** A list containing "Destination".
- Properties:** A table showing the class's properties:

Property	Cardinality	Type
hasAccommodation	Multiple	Accommodation
hasActivity	Multiple	Activity
hasPart	Multiple	Destination

Protégé-OWL OWLViz

- Expansion, visualization of ontologies



Protégé-OWL SWRL Rules

SWRL editor

The screenshot shows the Protégé 3.1 SWRL editor window. The title bar reads "family.swrl Protégé 3.1 (file:VC:\protege-owl\owl\family.swrl.pprj, OWL Files (.owl or .rdf))". The menu bar includes File, Edit, Project, OWL, Code, Window, Tools, and Help. The toolbar contains various icons for file operations and editing. The main interface has tabs for OWLClasses, Properties, Forms, Individuals, Metadata, and SWRL Rules. The SWRL Rules tab is active, displaying a table of rules.

Name	Expression
Def-hasAunt	$\rightarrow \text{hasParent}(?x, ?y) \wedge \text{hasSister}(?y, ?z) \rightarrow \text{hasAunt}(?x, ?z)$
Def-hasBrother	$\rightarrow \text{hasSibling}(?x, ?y) \wedge \text{Man}(?y) \rightarrow \text{hasBrother}(?x, ?y)$
Def-hasDaughter	$\rightarrow \text{hasChild}(?x, ?y) \wedge \text{Woman}(?x) \rightarrow \text{hasDaughter}(?x, ?y)$
Def-hasFather	$\rightarrow \text{hasParent}(?x, ?y) \wedge \text{Man}(?y) \rightarrow \text{hasFather}(?x, ?y)$
Def-hasMother	$\rightarrow \text{hasParent}(?x, ?y) \wedge \text{Woman}(?y) \rightarrow \text{hasMother}(?x, ?y)$
Def-hasNephew	$\rightarrow \text{hasSibling}(?x, ?y) \wedge \text{hasSon}(?y, ?z) \rightarrow \text{hasNephew}(?x, ?z)$
Def-hasNiece	$\rightarrow \text{hasSibling}(?x, ?y) \wedge \text{hasDaughter}(?y, ?z) \rightarrow \text{hasNiece}(?x, ?z)$
Def-hasParent	$\rightarrow \text{hasConsort}(?y, ?z) \wedge \text{hasParent}(?x, ?y) \rightarrow \text{hasParent}(?x, ?z)$
Def-hasSibling	$\rightarrow \text{hasChild}(?x, ?y) \wedge \text{hasChild}(?z, ?y) \wedge \text{differentFrom}(?x, ?z) \rightarrow \text{hasSibling}(?x, ?z)$
Def-hasSister	$\rightarrow \text{hasSibling}(?x, ?y) \wedge \text{Woman}(?y) \rightarrow \text{hasSister}(?x, ?y)$
Def-hasSon	$\rightarrow \text{hasChild}(?x, ?y) \wedge \text{Man}(?x) \rightarrow \text{hasSon}(?x, ?y)$
Def-hasUncle	$\rightarrow \text{hasParent}(?x, ?v) \wedge \text{hasBrother}(?v, ?z) \rightarrow \text{hasUncle}(?x, ?z)$

Below the table is a keyboard shortcut palette with the following symbols: a yellow circle, a blue square, a purple diamond, a red question mark, a green plus sign, a blue 'B', a red '≠', a green '=', a green checkmark, and a red 'X'. Below these are symbols for a table, logical AND (\wedge), logical implication (\rightarrow), parentheses ($()$), brackets ($[]$), and a left arrow (\leftarrow).

Expansions

- 6 types of plug-in:
 - Tab widget: u.i. tab, : Queries Tab
 - Slot widget: form widget: view/addition of value: TextFieldWidget
 - Back-end: storage mechanisms: RDF back-end
 - Createproject: conversion of different knowledge-bases in protege
 - Export: inverse of back-end
 - Project: use of a protege u.i.