

Protégé

- Ontology editor (OWL)
- Knowledge Base Framework (OKBC πρωτόκολλο <http://www.ai.sri.com/~okbc/>)
- Δωρεάν, ανοιχτού κώδικα (<http://protege.stanford.edu/download/download.html>)
- Java
- Επεκτάσιμο
- Παραμετροποιήσιμο
- Έξοδος (RDF(S), OWL, XML schema)

Protégé-Frames

- Ολοκληρωμένο U.I. και Knowledge Server
- Γνωσιακό μοντέλο συμβατό με OKBC
 - Οντολογίες – σύνολο κλάσεων
 - Οργάνωση σε ιεραρχία τάξεων (subsumption)
 - Σύνολο χαρακτηριστικών κλάσεων
 - Σύνολο στιγμιοτύπων κλάσεων (υποδείγματα)

Protégé-Frames

- 4 tabs
 - Classes
 - Slots
 - Forms
 - Instances

Protégé-Frames [classes tab]

- Editor οντολογιών.
 - Ορίσμός κλάσεων,
 - Ιεραρχία κλάσεων,
 - Χαρακτηριστικά, χαρακτηριστικά περιορισμοί,
 - Σχέσεις μεταξύ κλάσεων,
 - Ιδιότητες των σχέσεων

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 φόρμα για εισαγωγή χαρακτηριστικών
- Διαφορετικός τύπος – διαφορετική φόρμα
- Δυνατότητα αλλαγής της θέσης, μεγέθους, ετικέτας κλπ των φορμών

Protégé-Frames [forms tab]

The screenshot shows the Protégé 3.1 beta interface with the 'Forms' tab selected. The main window is titled 'newspaper Protégé 3.1 beta'. The interface is divided into several panes:

- FORM BROWSER:** Shows a tree view of classes for the 'newspaper' project. The 'Editor' class is selected.
- FORM EDITOR:** Shows the form editor for the 'Editor' class. The 'Display Slot' is set to 'name' and the 'Widget' dropdown is open, showing a list of widget options: <none>, YellowStickyWidget, TextAreaWidget, TextFieldWidget, UniqueStringWidget, URLWidget, and ImageWidget. The 'Current Job Title' field is highlighted with a green border.
- FORMS:** Shows a list of forms for the 'Editor' class, including 'Name', 'Date Hired', 'Current Job Title', 'Phone Number', and 'Other Information'.

Protégé-Frames [instances tab]

- Εργαλείο προβολής/τροποποίησης στιγμιοτύπων των κλάσεων της οντολογίας

Protégé-Frames [instances tab]

The screenshot displays the Protégé 3.1 beta interface for a project named 'newspaper'. The 'Instances' tab is active, showing a class hierarchy on the left, a list of instances in the center, and a detailed editor for the selected instance on the right.

CLASS BROWSER
For Project: newspaper

Class Hierarchy

- :THING
 - :SYSTEM-CLASS
 - Author
 - News_Service (2)
 - Columnist (2)
 - Editor (4)
 - Reporter (3)
 - Content
 - Layout_info
 - Library (1)
 - Newspaper (6)
 - Organization (1)
 - Person

INSTANCE BROWSER
For Class: Editor

- Chief Honcho
- Mr. Science
- Ms Gardiner
- Sports Nut

INSTANCE EDITOR
For Instance: Chief Honcho (instance of Editor, inte...)

Name	Chief Honcho	Salary	150000.0
Date Hired		Responsible For	Sports Nut Ms Gardiner
Current Job Title		Sections	Magazine Local News Automotive Business World News
Phone Number			
Other Information			

Protégé-Frames [queries tab]

- Ερωτήματα στο project.
 - Επιλογή μίας ή περισσότερων κλάσεων
 - Ένα ή περισσότερα χαρακτηριστικά
 - Ένα ή περισσότερα κριτήρια
- Αποθήκευση των ερωτημάτων.

Protégé-Frames [queries tab]

The screenshot displays the Protégé 3.1.1 application window. The title bar reads "tutorial Protégé 3.1.1 (file:\C:\Program%20Files\Protege_3.1\examples\tutorial\tutorial.pprj, Protégé Files (.p...". The menu bar includes "File", "Edit", "Project", "Window", and "Help". The toolbar contains various icons for file operations and navigation. The main interface is divided into several sections:

- Navigation Tabs:** "Classes", "Slots", "Forms", "Instances", and "Queries" (selected).
- Query Editor:** A table with columns for "Class", "Slot", and "String". The "Class" column has a "+" icon, and the "Slot" column has a "-" icon. Below the table are buttons for "More", "Fewer", "Clear", and "Find". Radio buttons for "Match All" (selected) and "Match Any" are also present.
- Query Name:** A text input field with an "Add to Query Library" button.
- Query Library:** A list area for saved queries.
- Search Results:** A large empty area on the right side of the window.

Protégé-OWL

- Editor της OWL (web ontology language)
- Load – Σώζει οντολογίες γραμμένες σε RDF OWL
- Οπτικοποίηση κλάσεων, ιδιοτήτων και SWRL κανόνων (συνδιασμός OWL και RuleML)
- Ορισμός εκφράσεων OWL
- Διαθέτει ένα Java API για την ανάπτυξη U.I.

Protégé-OWL

- Οι επιλογές είναι οι:
 - Owl Classes
 - Properties
 - Forms
 - Individuals

Protégé-OWL [OWLClasses]

- Προβολή και τροποποίηση ιεραρχιών
- Λεπτομέρειες της κάθε κλάσης στα δεξιά
- Προσθήκη σχολίων, ετικετών κλπ
- Περιορισμοί
- 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:VC:\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 starts with `owl:Thing` and includes classes like `Accommodation`, `Activity`, and `Destination`. The `FamilyDestination` class is highlighted under the `Destination` class.
- Class Editor:** Shows the editor for the `FamilyDestination` class. It includes:
 - Name:** `FamilyDestination`
 - rdfs:comment:** "A destination with at least one accommodation and at least 2 activities."
 - Annotations:** A table with one entry: `rdfs:comment` with the value "A destination with at le...".
 - Asserted Conditions:** Shows the class is **NECESSARY & SUFFICIENT** with conditions: `Destination`, `hasAccommodation ≥ 1`, and `hasActivity ≥ 2`.
 - Properties:** Lists properties: `hasAccommodation` (multiple Accommodation, ≥ 1), `hasActivity` (multiple Activity, ≥ 2), and `hasPart` (multiple Destination).
 - Disjoints:** Shows `RetireeDestination` as a disjoint class.

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

classification

- έλεγχος συνέπειας της οντολογίας
- έλεγχος ανεπιθύμητων σχέσεων ανάμεσα σε κλάσεις
- αυτόματη ταξινόμηση στιγμιοτύπων σε κλάσεις

Protégé-OWL [OWLClasses]

The screenshot displays the Protégé 3.1 interface for editing an OWL ontology. The main window shows the 'SUBCLASS RELATIONSHIP' view for the project 'travel'. The 'Asserted Hierarchy' and 'Inferred Hierarchy' panes show a tree structure of classes. The 'CLASS EDITOR' pane is open for the 'Capital' class, showing its name, comment, and asserted conditions. The 'Classification Results' pane at the bottom shows the current state of the classes and their superclasses.

Asserted Hierarchy

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

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: Capital

rdfs:comment:

Asserted Conditions:

- City (NECESSARY & SUFFICIENT)
- City (NECESSARY)

Classification Results

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

Protégé-OWL [Properties]

- Προβολή/Τροποποίηση χαρακτηριστικών των ιδιοτήτων

The screenshot displays the Protégé 3.1 software interface. The title bar indicates the project file is 'travel.pprj'. 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 'travel' project, including hasAccommodation, hasActivity, hasCity, hasContact, hasEMail, hasPart, hasRating, hasStreet, hasZipCode, and isOfferedAt.
- PROPERTY EDITOR:** Currently displays the configuration for the 'hasAccommodation' property. It includes tabs for Name, Equivalents, SameAs, and DifferentFrom. The Name tab is active, showing the property name 'hasAccommodation' and an 'rdf:type' field set to 'owl:ObjectProperty'. There is also an 'Annotations' table with columns for Property, Value, and Lang.
- Domain and Range:** The Domain is set to 'Destination' and the Range is set to 'Accommodation'. Both are shown in separate text areas.
- Property Characteristics:** A set of checkboxes for Functional, InverseFunctional, Symmetric, and Transitive. All are currently unchecked.
- Inverse:** A checkbox for the Inverse property, which is also unchecked.

Protégé-OWL [individuals]

- Προβολή/τροποποίηση στιγμιοτύπων
- Αυτόματη δημιουργία φορμών ανάλογα με τον ορισμό της κλάσης και των ιδιοτήτων της
- `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 under `UrbanArea`.
- INSTANCE BROWSER:** Shows instances for the selected class `Capital`. The "Asserted" tab is active, showing instances `Canberra` and `Sydney`. The `Sydney` instance is selected.
- INDIVIDUAL EDITOR:** Shows the editor for the selected instance `Sydney` (instance of `Capital`). It includes fields for Name, SameAs, DifferentFrom, and Annotations. The Name field contains "Sydney". Below the editor, there are property lists for `hasAccommodation` (containing `FourSeasons`), `hasPart` (containing `BondiBeach` and `CurrawongBeach`), and `hasActivity`.

The bottom status bar shows icons for saving, undo, and redo.

Protégé-OWL – RDF Schema

προβολή / τροποποίηση RDFS.

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:

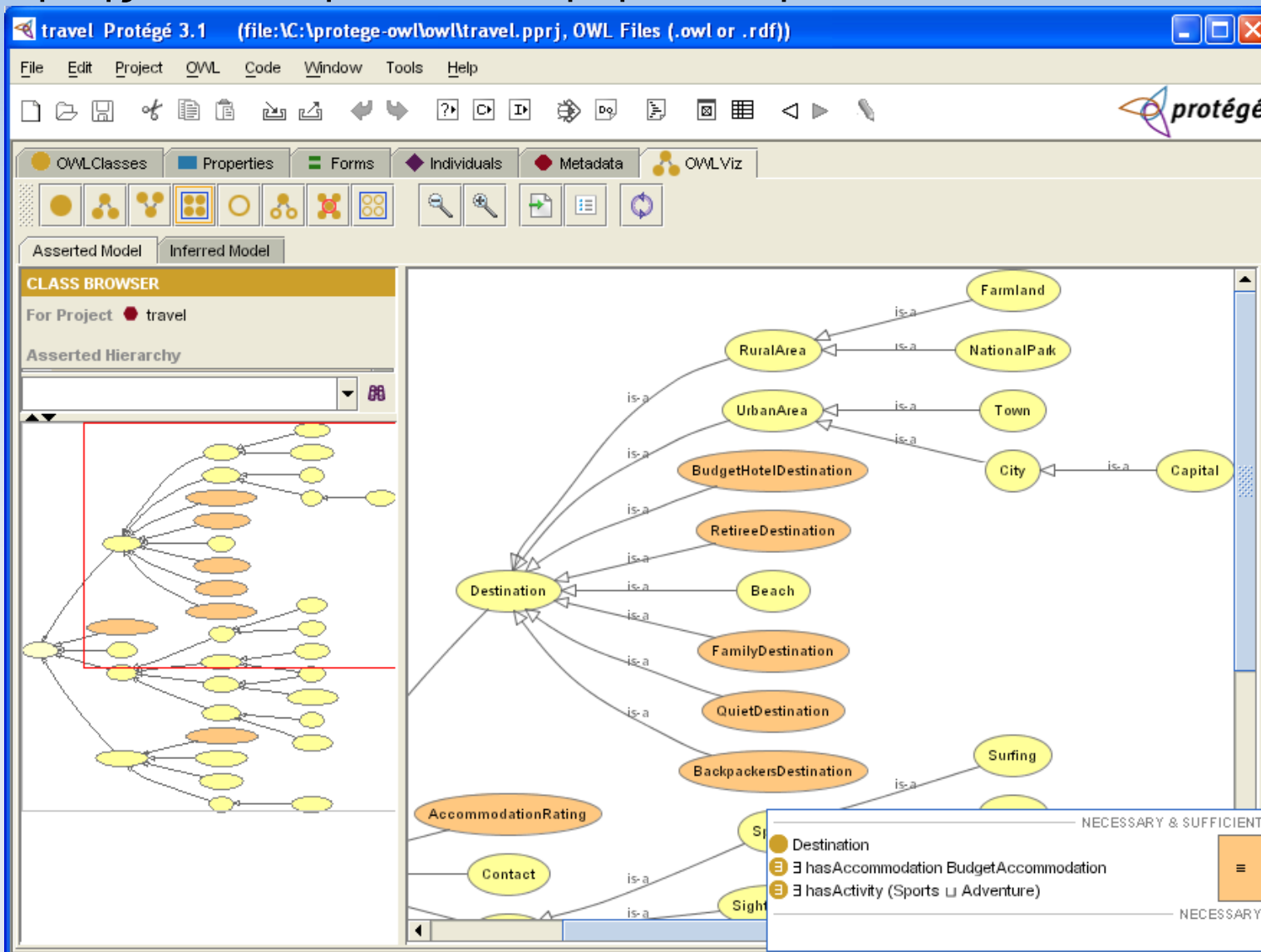
- Subclass Relationship:** Shows the asserted hierarchy for the project "travel-rdf". The hierarchy is as follows:
 - owl:Thing
 - Accommodation
 - AccommodationRating
 - Activity
 - Contact
 - Destination
 - BackpackersDestination (selected)
 - Beach
 - BudgetHotelDestination
 - FamilyDestination
 -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

- Class Editor:** Shows the editor for 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 columns Property, Value, and Lang. It contains one entry: rdfs:comment with the value "A destination that prov..." and an empty Lang field.
- rdfs:subClassOf:** A list showing "Destination" as a subclass.
- Properties:** A table with columns Property, Cardinality, and Type. It lists three properties:

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

Protégé-OWL OWLViz

- Δημοφιλής επέκταση, οπτικοποίηση οντολογιών



Protégé-OWL SWRL Rules

SWRL editor

The screenshot shows the Protégé 3.1 SWRL editor interface. The title bar indicates the file is 'family.swrl' and the project is 'Protégé 3.1'. 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 window is divided into tabs: 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 toolbar with icons for OWL classes, properties, individuals, and SWRL rules, along with logical symbols like \wedge , \rightarrow , $()$, $[]$, and \leftarrow .

ΕΠΕΚΤΑΣΕΙΣ

- 6 τύποι plug-in:
 - Tab widget: u.i. tab, : Queries Tab
 - Slot widget: form widget: προβολή/προσθήκη τιμής : TextFieldWidget
 - Back-end: μηχανισμός αποθήκευσης: RDF back-end
 - Createproject: μετατροπή διαφορετικά γραμμένων knowledge-base σε protege
 - Export: αντίστροφο του back-end
 - Project: χρήση ενός protege u.i.