

Knowledge Engineering Tools and Semantic Translation

Andrzej Bialecki

WebGiro, Chief System Architect

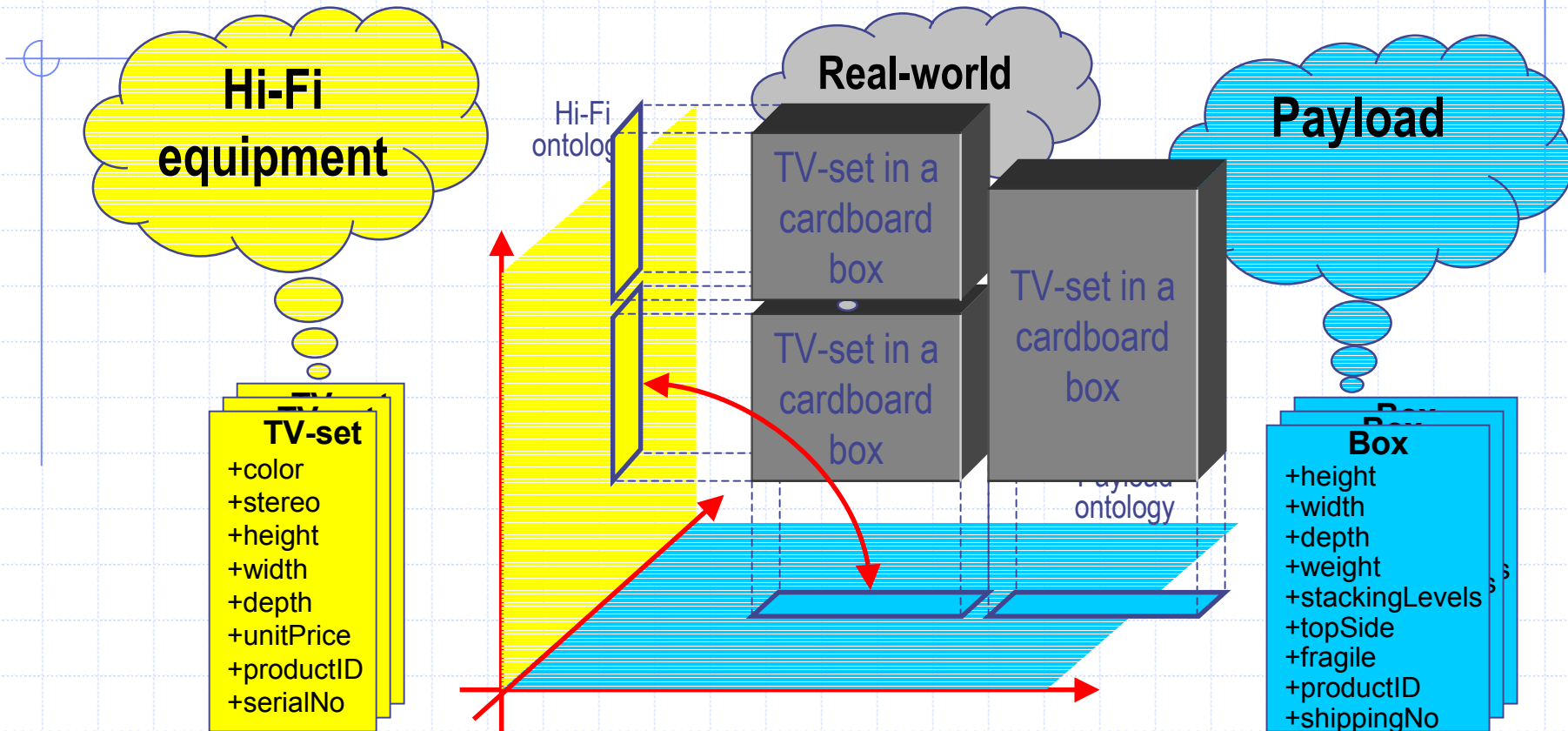
<abial@webgiro.com>

ECIMF Project Group Meeting, Brussels, 16.10.2001

Knowledge Engineering Tools

- ◆ **Ontology Engineering (OE)**
 - Ontology creation, editing, comparing ...
- ◆ **Knowledge Acquisition (KA)**
 - From human experts
 - From machine-readable sources
- ◆ **Automated Reasoning (AR)**
 - Inference
 - Verification
- ◆ **Often support multiple aspects**

Semantic Translation and ontologies



◆ Ontologies

- Shared conceptualization of a domain, "consensus view"
- Semantic content limited to the domain of interest

◆ Approximate re-classification

- ◆ Semantic enrichment (needed for disambiguation)
- ◆ Upper-level ontologies (shared vocabularies)

Why KE Tools for ST?

- ◆ ST relies on the use of ontologies
 - Efficient ontology management needed
- ◆ Shared ontology design
 - Efficient ontology design support needed
- ◆ Validation and inference
 - Consistency checking and discovering non-obvious relations are crucial
- ◆ KE is a widely-researched topic
 - There are many supporting tools and models
 - BUT few of them support ontology mapping

Protégé 2000

- ◆ Features: OE, KA, AR
- ◆ Knowledge model: frame-based
 - Class, slot
 - Multiple inheritance
 - Template slots
- ◆ Modular and extensible
 - Plug-in system, with multitude of modules available (KA, inference, merging, WordNet...)
- ◆ Open Source (MPL)

Protégé Interface: Classes, slots

The screenshot shows the Protégé-2000 interface with the following components:

- Project:** newspaper (E:\Protege-2000\examples\newspaper\newspaper.pprj)
- Menu:** Project, Edit, Window, Help
- Toolbar:** File, Edit, Undo, Redo, Print, Delete
- Navigation:** Classes, Slots, Forms, Instances, Queries
- Relationship:** Superclas... (View, Create, Refresh, Delete)
- Class Hierarchy:**
 - .THING A
 - :SYSTEM-CLASS A
 - Author A
 - News_Service
 - Columnist M
 - Editor M
 - Reporter M
 - Content A
 - Layout_info A
 - Library
 - Newspaper
 - Organization
 - Person

- Superclasses:** Author A, Employee A
- Editor Class Details:**
- Name:** Editor
- Documentation:** Editors are responsible for the content of sections.
- Role:** Concrete
- Constraints:** (Empty)
- Template Slots:**

Name	Type	Cardinality	Other Facets
S byname	String	single	
S current_job_title	String	single	
S date_hired	String	single	
S name	String	single	
S other_information	String	single	
S phone_number	String	single	
S responsible_for	Instance	multiple	classes={Employee}
S salary	Float	single	
S sections	Instance	multiple	classes={Section}

Protégé Interface: OntoViz

newspaper Protégé-2000 (E:\Protege-2000\examples\newspaper\newspaper.pprj)

Project Edit Window Help Prompt

Classes Slots Forms Instances Queries Ontoviz

Config +C + - S Op C

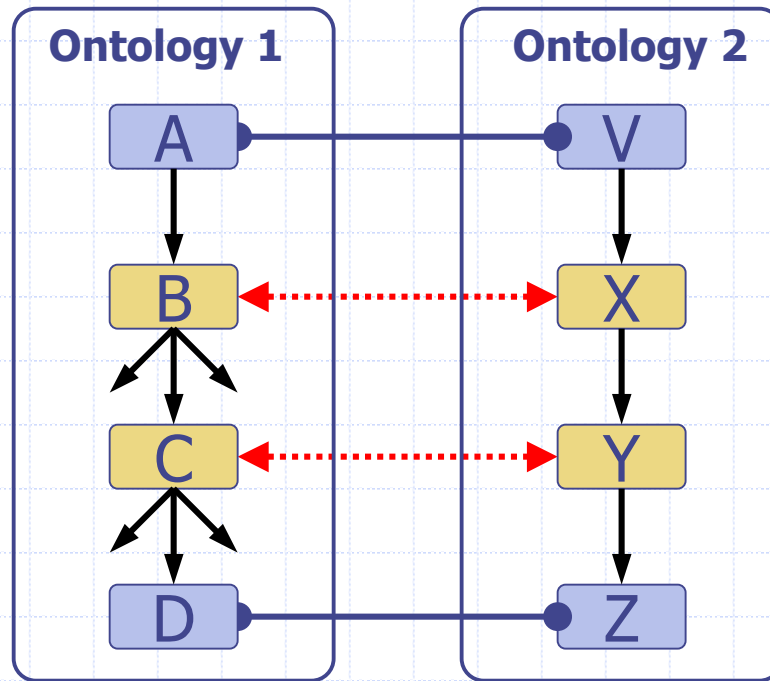
frame	sub	sup	slx	isx	sl
Author	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Person	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Newspaper	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Library	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Classes

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

```
graph BT; Library -- isa --> THING; Newspaper -- isa --> THING; Person -- isa --> THING; Author -- isa --> Person; Employee -- isa --> Person; Salesperson -- isa --> Employee; Reporter -- isa --> Employee; Columnist -- isa --> Employee; Editor -- isa --> Author; News_Service -- isa --> Author; SanJoseMercuryNews[San Jose Mercury News] -- io --> Library; Editor --- EditorBox[Editor];
```

Anchor-PROMPT: semantic matching



- ◆ Requires providing pairs of anchors
 - These can also be suggested based on heuristics (e.g. lexicographical similarity)
- ◆ Suggests matching terms by analyzing non-local context
 - Tests have shown ca. 60% correct guesses

Anchor-PROMPT: e-commerce scenario

◆ Two ontologies

- O1: some well-known standard (e.g. ebXML)
- O2: simple general ledger database tables
 - ◆ Tables -> classes
 - ◆ Fields -> slots

◆ User with limited ontology knowledge (typical for SME)

- User can provide the most obvious matches
 - ◆ E.g. O1(Person) -> O2(Person)
- Anchor-PROMPT suggests less obvious matches
 - ◆ E.g. O1(Address) -> O2(Location)

◆ Current implementation:

- Only earlier version available (plain PROMPT)
- Only ontology merging or sub-selecting, not mapping

Protégé Interface: PROMPT

MotorVehicle Protégé-2000 (E:\Protege-2000\examples\rdf\MotorVehicle.ppr)

Project Edit Window Help Prompt

Classes Slots Forms Instances Queries Prompt

Suggestions Conflicts Creating operation

To Do list

Name	Arg1	Arg2	Params
merge	Person MotorVehicle	Person newspaper	
copy	MotorVehicle MotorVehicle		params = {subs}
copy	Author newspaper		params = {subs}
copy	Content newspaper		params = {subs}
copy	Employee newspaper		params = {subs}
copy	Layout_info newspaper		
copy	Library newspaper		
copy	Newspaper newspaper		
copy	Organization newspaper		

Result classes Result slots Result instances

merged

- :THING
- :SYSTEM-CLASS
- MotorVehicle
- Person

Edit Operation

merge classes

MotorVehicle newspaper

Choose class + - Choose class + -

Person Person

parents (up to the root) subclasses instances

Do It ToDo list Cancel

Reason for selected suggestion

frames have identical names

Do It

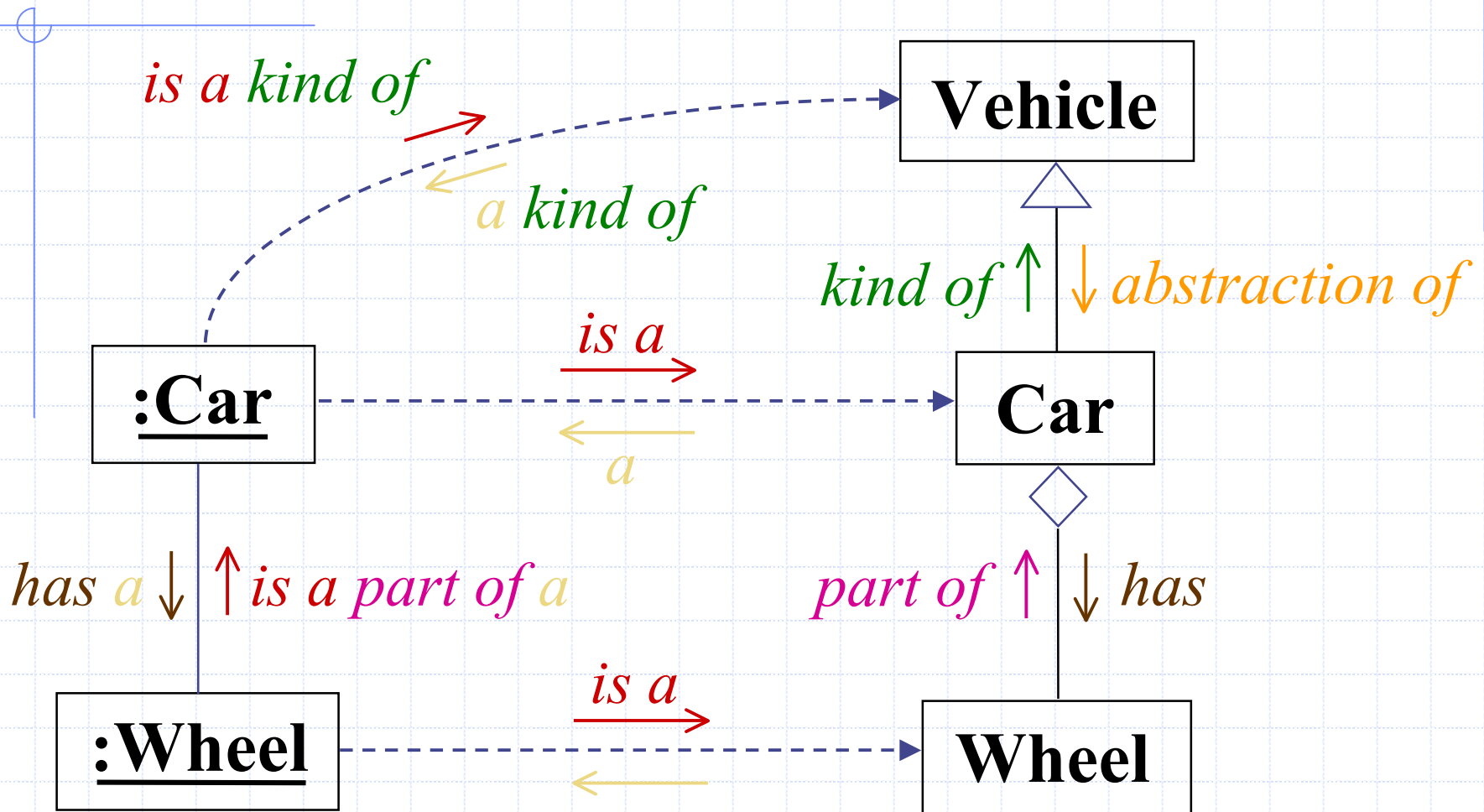
Protégé Summary

- ◆ Very flexible and extensible
- ◆ Cross-platform, open source (MPL)
- ◆ Rich functionality
- ◆ Extensive API for model management
- ◆ Explicit support for semantic matching
- ◆ BUT: the knowledge model is difficult for non-experts
 - This could be hidden behind a special-purpose user interface module

Conzilla Concept Browser

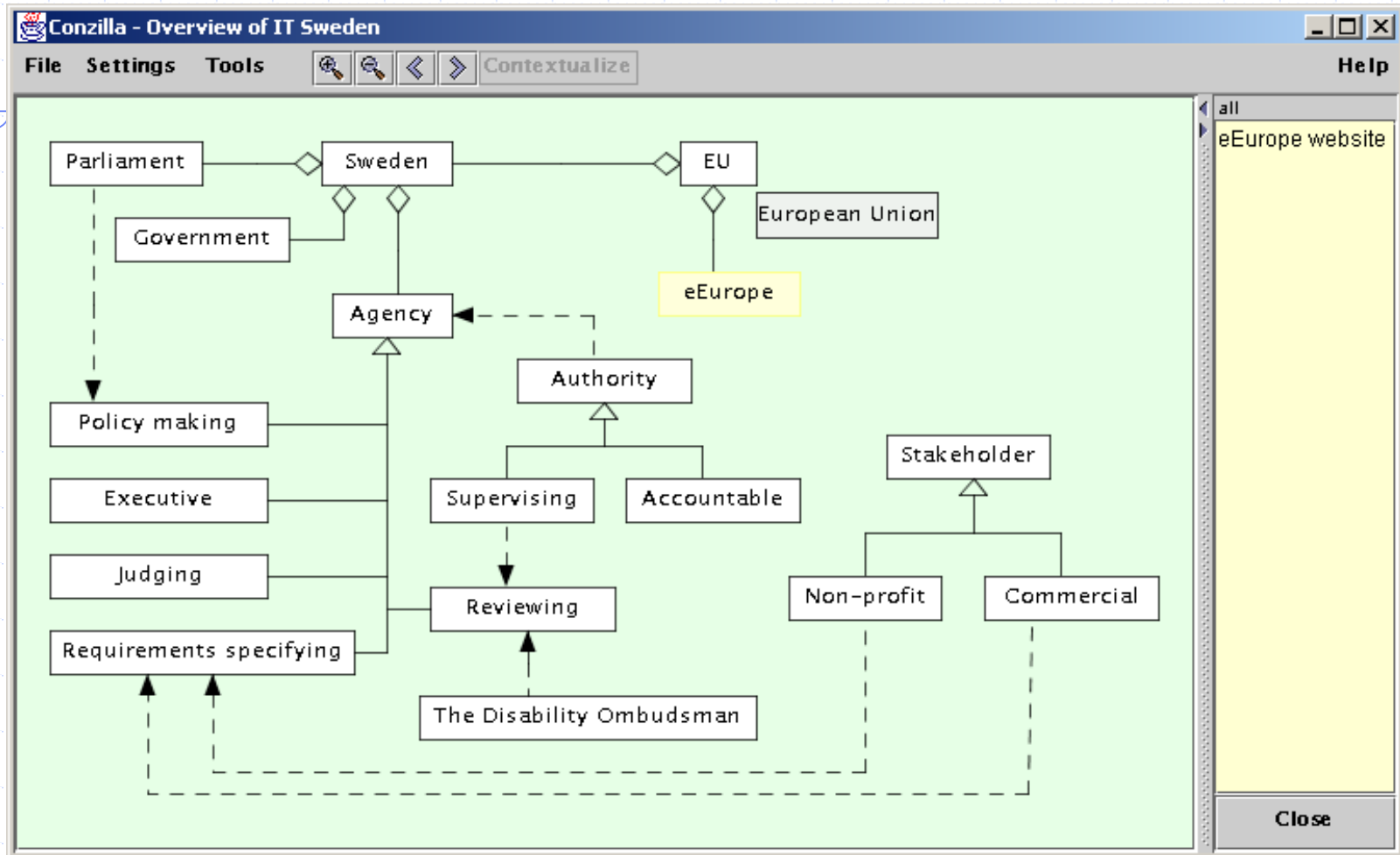
- ◆ Features: OE, KA
- ◆ Cross platform, open source (GPL/MPL)
- ◆ Knowledge model: "neurons"
 - Associated with multilingual meta-data
 - Neuron types:
 - ◆ Class
 - ◆ Specialization (subclass-of [= *isa* in KE])
 - ◆ Association (related-to)
 - ◆ Aggregation (part-of)
 - ◆ Instantiation (is-a [\neq *isa* in KE!])

Conzilla Unified Language Modeling



- ◆ A UML profile, readable in natural language
- ◆ More precise and concise than traditional KE notation (isa, instanceof)

Conzilla Interface



- ◆ Easily converted to/from natural language
- ◆ Easily accessible meta-data (multi-lingual)
- ◆ Easily accessible content
- ◆ Understandable for both business and technical people

Conzilla Summary

- ◆ Unique context navigation
 - Helps to keep track of the concept relationships
- ◆ UML profile for diagrams
 - UML is well-understood for a wider audience
 - ULM - diagrams easily converted to/from natural language
- ◆ BUT:
 - Currently uses its own XML format (RDF in works)
 - Difficult content editing

Conclusions: Protégé/Conzilla/PROMPT

- ◆ Conzilla as a Protégé plug-in:
 - ◆ Adds a non-expert interface to the KB
 - ◆ Easier to understand graph. visualization of the ontology
 - ◆ Ontology can be exported to many formats
 - ◆ All other Protégé plug-ins become available for additional processing
 - ◆ Advanced Protégé interface available for experts
- ◆ Anchor-PROMPT can help less experienced users
- ◆ All code is under open source license (MPL)

Further information

◆ Protégé website

- <http://protege.stanford.edu>

◆ Anchor-PROMPT

- *Anchor-PROMPT: Using Non-Local Context for Semantic Matching*, N.F. Noy, M.A. Musen, Aug 2001

◆ Conzilla

- <http://www.conzilla.org>

◆ ECIMF Project Information Center

- <http://www.ecimf.org>