VocBench

Caterina Caracciolo, FAO
Armando Stellato, Uni Tor Vergata

Food and Agriculture Organization
of the United Nations

VOCBENCH 2.0

A Collaborative Environment for Management of

SKOS/SKOS-XL Concept Schemes

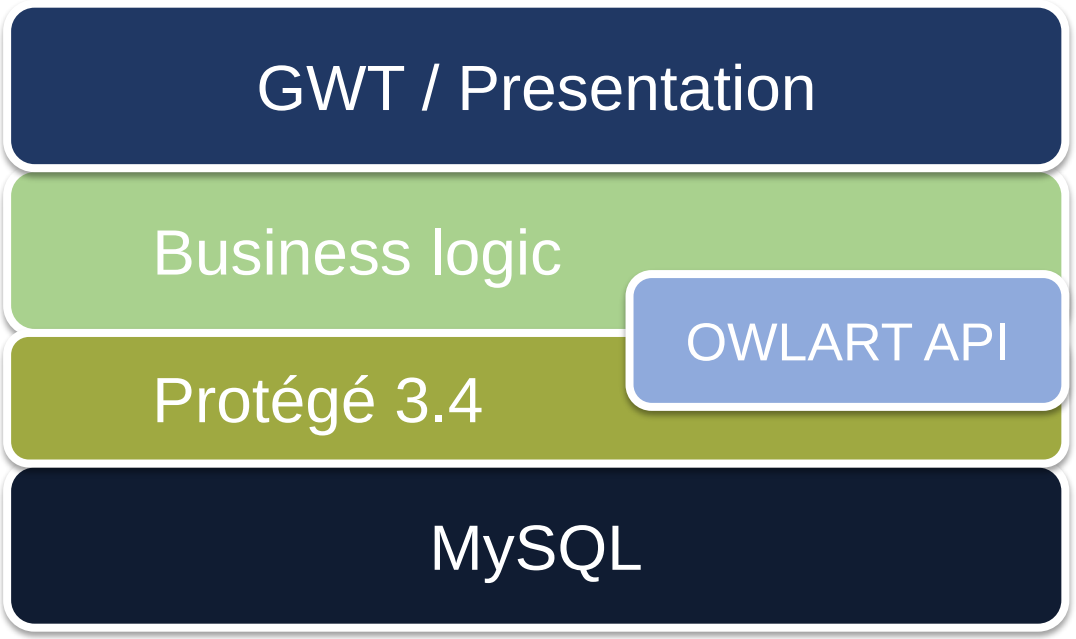
Why was it built?

AGROVOC (big agriculture vocabulary developed by FAO)

- In 2004: >32 000 concepts in up to 22 languages
- A global group of terminologists.
- No existing standard for thesauri
- No existing tool that met FAO's needs

V1 - 2010

- Google Web Toolkit
- Lucene
- Protégé API
- OWLART API
- MySQL
- Custom OWL model



V1 Problems

- Couldn’t support other triple stores (Glued to Protégé API)
- Custom OWL model
- No support for emerging standards, e.g. SKOS
- No import
- No support for alignments
  - AGROVOC aligned to a dozen other vocabularies
- Complicated export
- No SPARQL endpoint

Towards V2

- Market Analysis
  - Choose a commercial product or
  - Build a new version
- Collaboration
  - U. of Rome Tor Vergata
  - MIMOS Berhad (FAO Center of Excellence)
  - EC FP7 projects agInfra and SemaGrow
- The Decision
  - Build a new version leveraging new frameworks and an OSGi pluggable SOA design
  - Support SKOS (& SKOS-XL) and importing

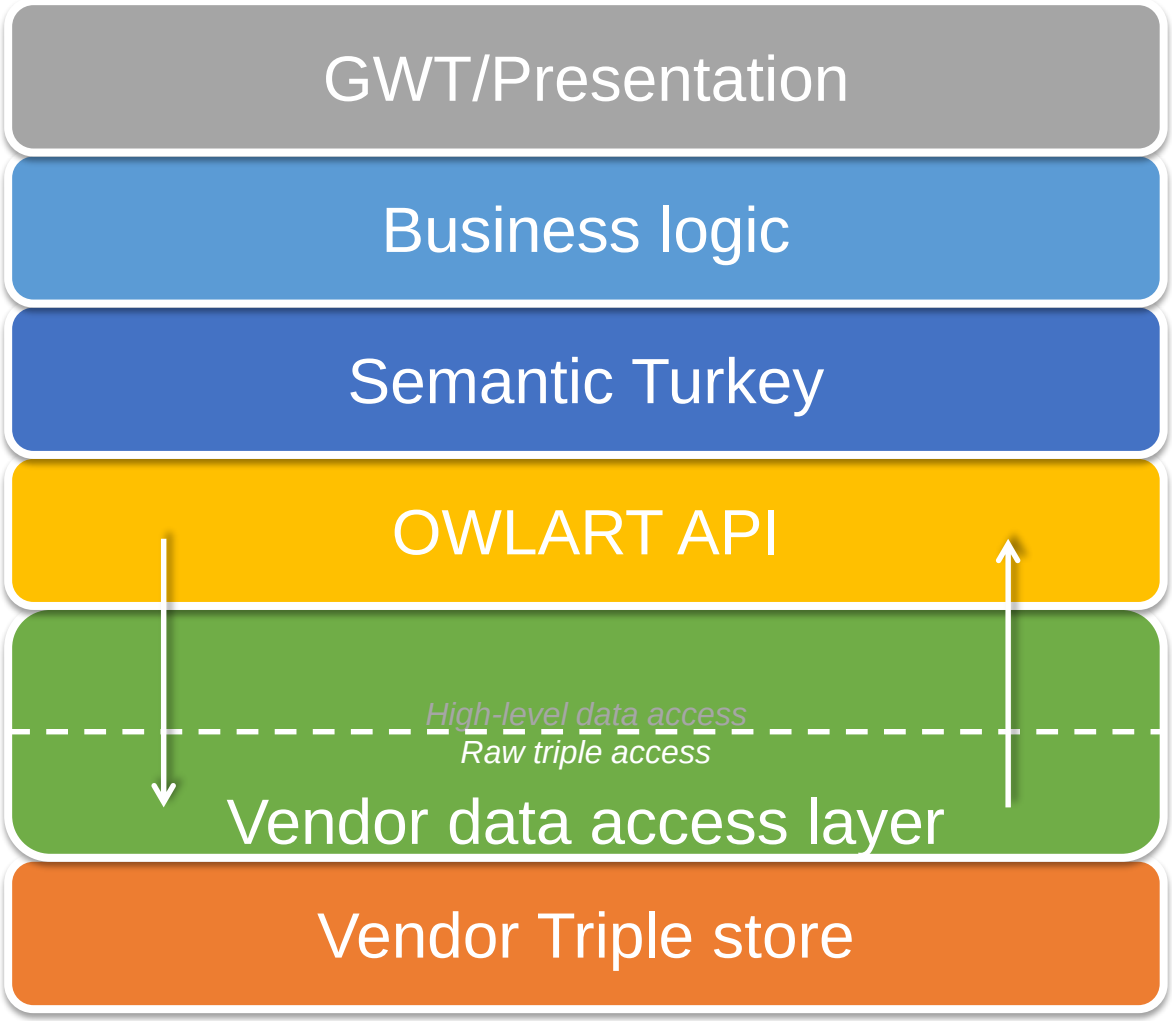
Objectives for VB2.0…

- A completely rebuilt backing framework for the service and data layers, based on an already existing open source project: Semantic Turkey[1]
  - Based on OSGi Open Services Gateway
  - Open Connectibility to most notable RDF middleware and triple storing technologies (Sesame2, Jena, Allegrograph…)
  - Native support for SKOS and SKOSXL over RDF (no more conversions from internal legacy models), other than OWL

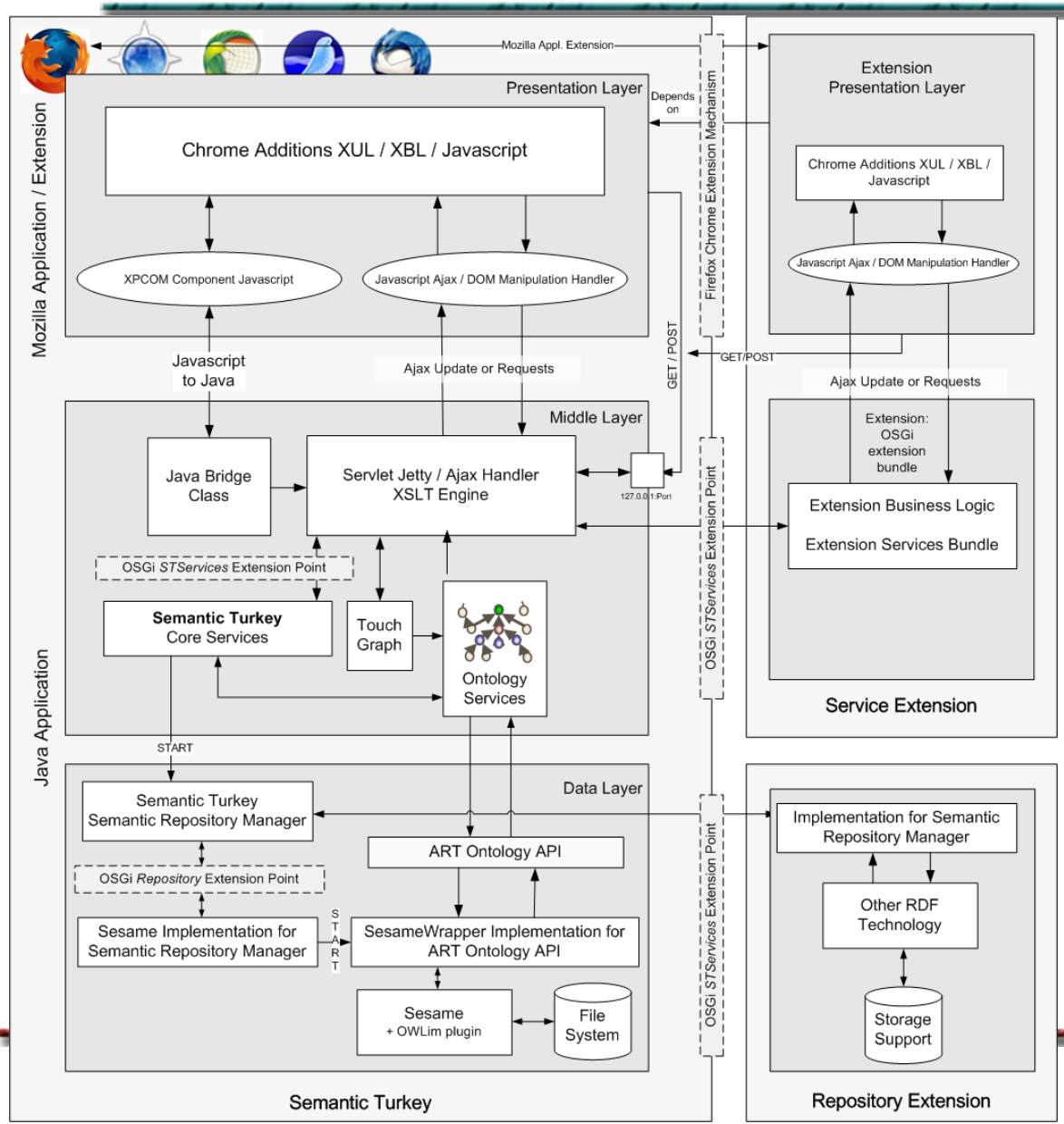
- VB1.0 User Interface remains mostly unchanged in the first release of VB2.0

1. http://semanticturkey.uniroma2.it/

V2 Design


GWT/Presentation
Business logic
Semantic Turkey
OWLART API
High-level data access
Raw triple access
Vendor data access layer
Vendor Triple store

Vocbench 2.0 (and ST) Architecture



Three layered extensible architecture

Presentation Layer

- GWT (Google Web Toolkit) Vocbench User Interface (*Mozilla apps in the original framework*)

Services Layer

- Enables communication between the client (Vocbench UI) and the ontology persistence layer.
- HTTP based Services accessed through the Ajax paradigm
- OSGi Extensible Servicing System

Persistence Layer

- Access to ontological knowledge.
- Based on dedicated ontology API, which can be implemented through use of different technologies.

Why should I buy it?

Collaborative Management

- Validation&Publication Workflow (propose, validate, publish, revise, deprecate…)
- Fine grained user management
  - both users and functionalities may be associated in groups
  - Functionalities (or groups of) may be assigned to different users (or groups of)
- Full editing history (not only concepts, but most of the actions can be subject to validation too)
- RSS Feeds
- Fine-grained metadata and editorial notes: SKOS-XL and reified definitions allow for timestamped status and rich editorial notes

Multilinguality

- Strong support for multi-lingual thesauri management
- Application itself is also multilingual (currently support for english, dutch, spanish, more languages coming)

Native RDF support

- Support for different triple stores
- Possibilty to SPARQL query/update through a dedicated interface with syntax completion/highlight
- SKOS-XL management
  - If preferred, SKOS-core export through available conversion tools

Large scale thesauri management

- Scalability limited only by the underlying triple store

Extensibility

- OSGi connectable services

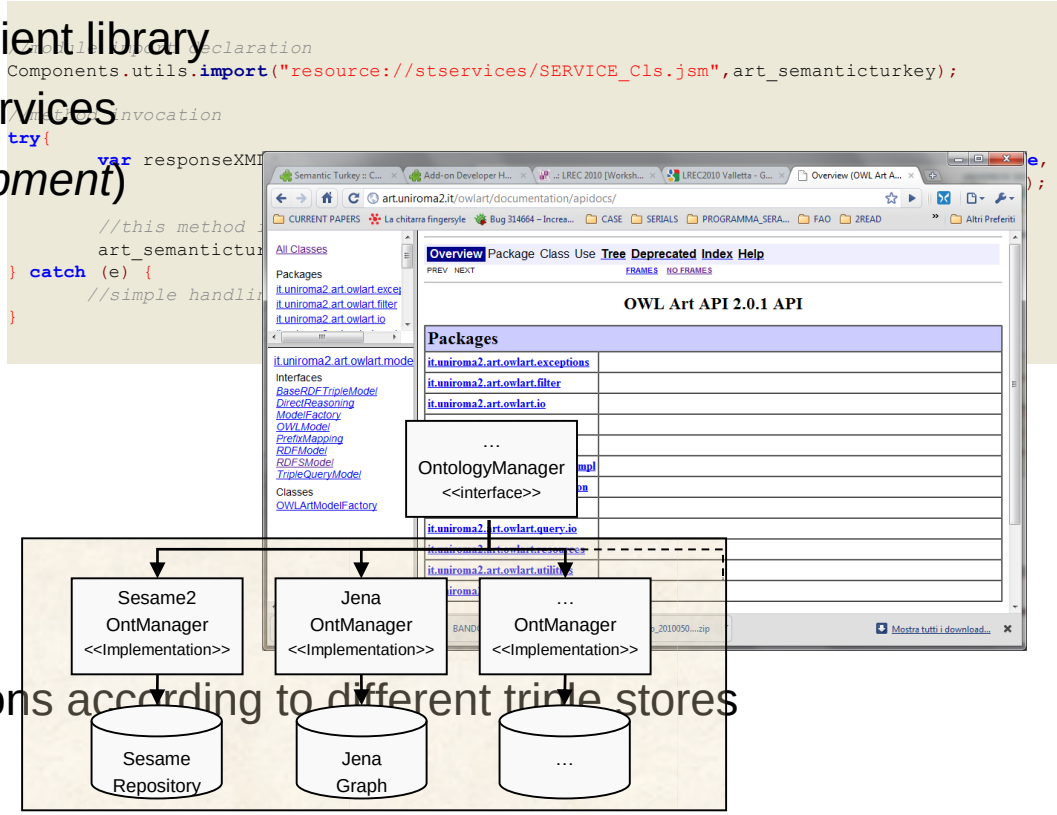
Advanced skos:ConceptScheme Management

- SKOS allows for non-trivial management of multiple conceptual schemes, which is fully supported by VB

And, last but not the least: **Free and Open Source**! (http://vocbench.uniroma2.it)

Extensibility

- Client: GWT[1] / Mozilla[2] technologies
  - GWT Java / Mozilla JavaScript client library
  - Java/JavaScript Client API for services
  - Ontology Widgets (*under development*)
- Services: OSGi
  - Service Extensions
  - Plugins
- Data and Models: OSGi
  - Ontology Manager implementations according to different triple stores



1. Google Web Toolkit Java Client on the VOCBENCH Collaborative Suite
2. Mozilla Javascript Client on Semantic Turkey based desktop tools

Recent Achievements: Service Deployment through Spring + custom controller generation

Current service implementation

```
if (request.equals(Req.isTopConceptRequest)) {
     String skosConceptName = setHttpPar(Par.concept);
     String schemeName = setHttpPar(Par.scheme);
     checkRequestParametersAllNotNull(Par.concept, Par.scheme);
     response = isTopConcept(skosConceptName, schemeName);


public Response isTopConcept(String skosConceptName, String schemeName) {
       SKOSModel skosModel = getSKOSModel();

       try {
           ARTResource[] graphs = getUserNamedGraphs();
           ARTURIResource skosConcept = retrieveExistingResource(skosModel, skosConceptName, graphs);
           ARTURIResource skosScheme  = retrieveExistingResource(skosModel, schemeName, graphs);
           return createBooleanResponse(skosModel.isTopConcept(skosConcept, skosScheme, graphs));
       } catch (NonExistingRDFResourceException e) {
           return logAndSendException(e);
       } catch (ModelAccessException e) {
           return logAndSendException(e);
       }

    }
```

Annotations may contain explicit validation checks with respect to the application's semantics

Arguments and returned values are now explicitly managed with their native types.
Marshalling/Unmarshalling to the serialization formats adopted by the service is demanded to dedicated components

Separation of service method-controller / automatic generation of controller

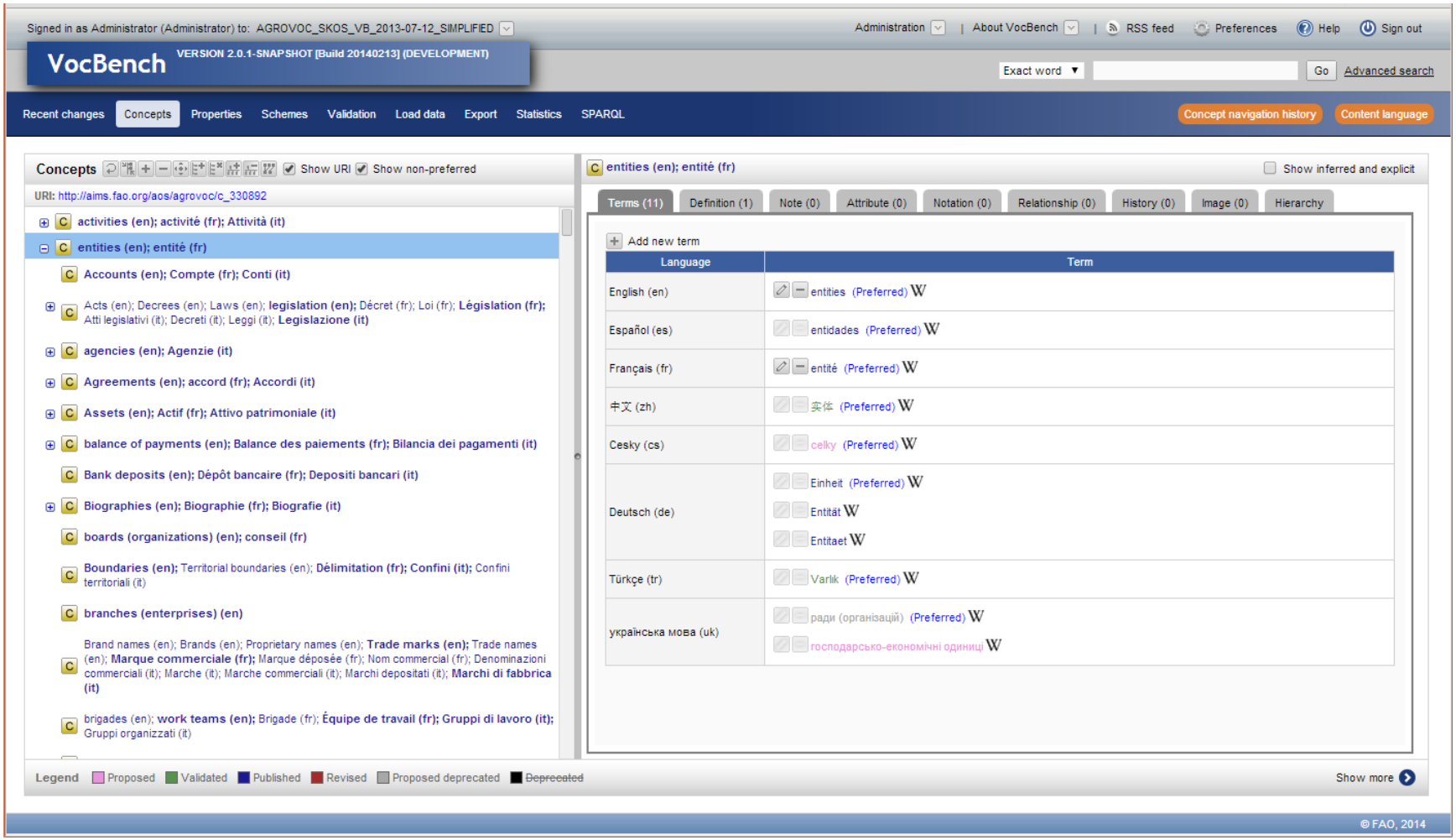
```
public boolean isTopConcept(@present ARTURIResource skosConcept, @Optional(defaultValue = ":mainScheme") ARTURIResource skosScheme)
                                                                                    throws NonExistingRDFResourceException, ModelAccessException {
       SKOSModel skosModel = getSKOSModel();
       ARTResource[] graphs = getUserNamedGraphs();
       return skosModel.isTopConcept(skosConcept, skosScheme, graphs);
}
```

The method signature then drives the generation of the controller, which is the direct frontend for the service. Exceptions are serialized in the response (the content of which codes both data and application-level error codes) and data validation annotations are managed by Spring data validation methods

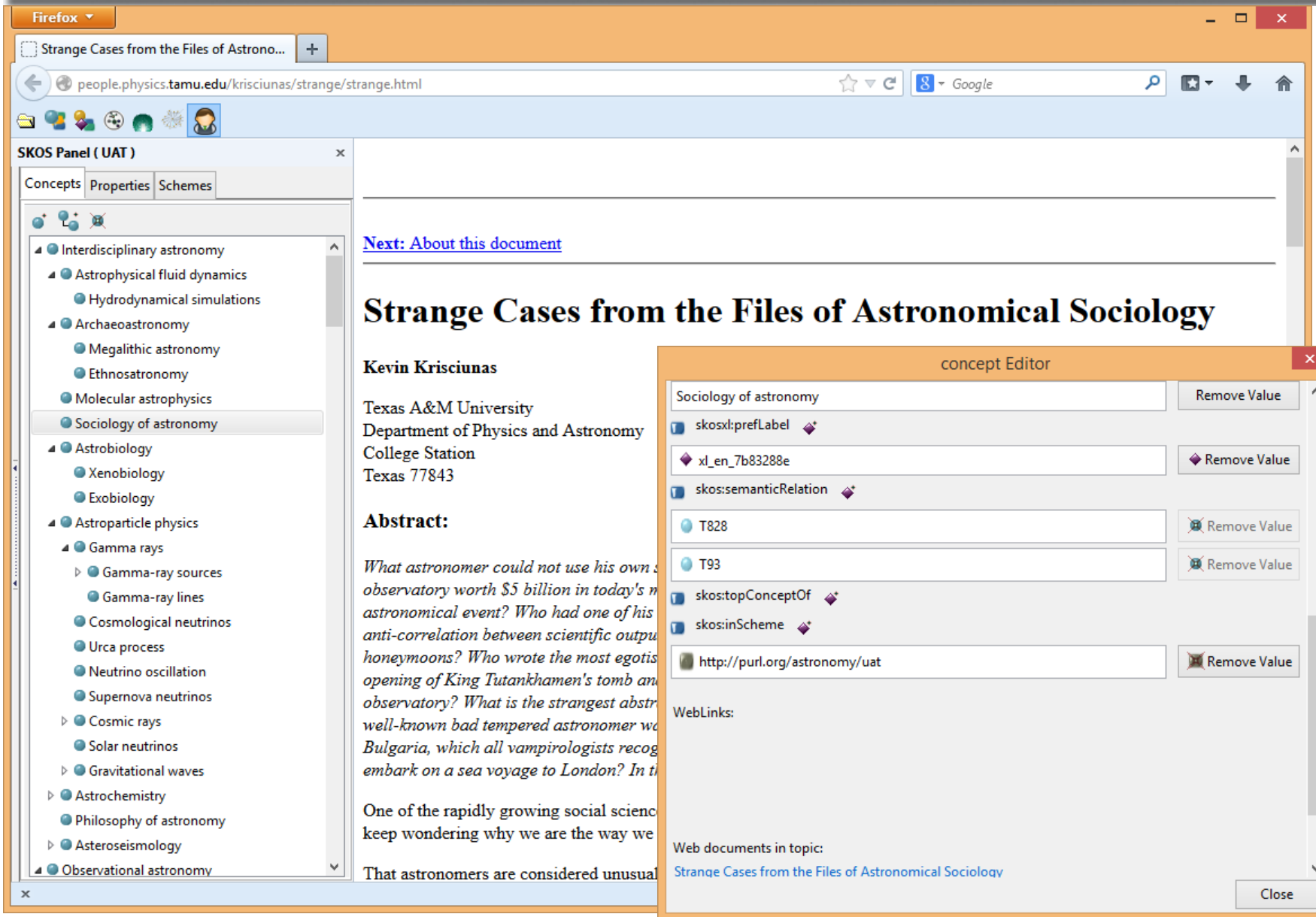
V2: Triple Store Support

- What triple stores do we currently support?
  - Sesame2 (standard internal triple stores, both in-memory and native) and
  - OWLIM (through sesame2 remote connection), plus
  - ANY repository which can be accessed through a plain sesame2 remote connection.
- For OWLIM, we make use of the specific triple indexing it offers
- Non-updated interfaces for Allegrograph and Jena
- Support for Virtuoso coming from a VocBench partner

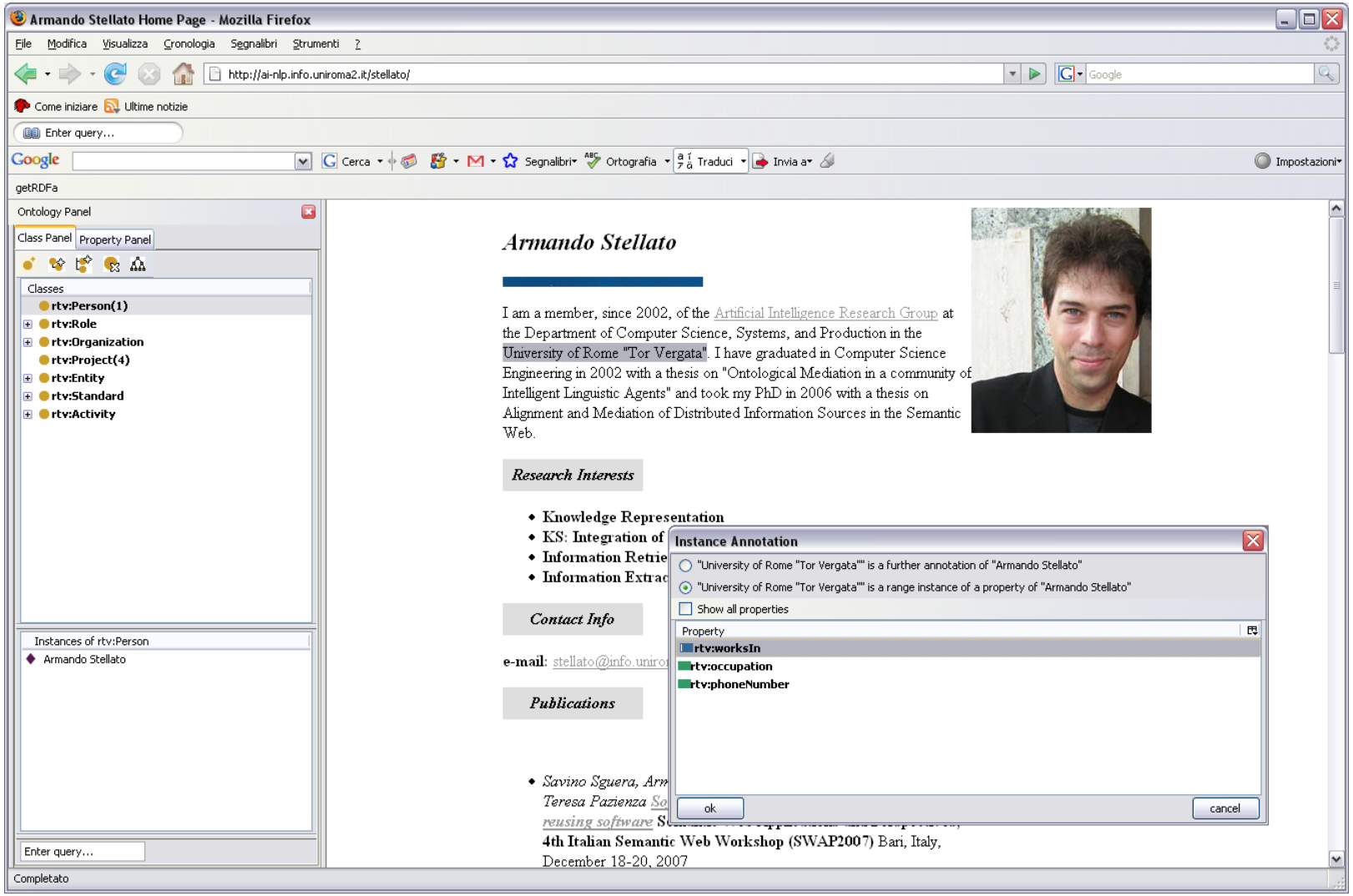
…and here we are!!



VB “desktop version”: Semantic Turkey



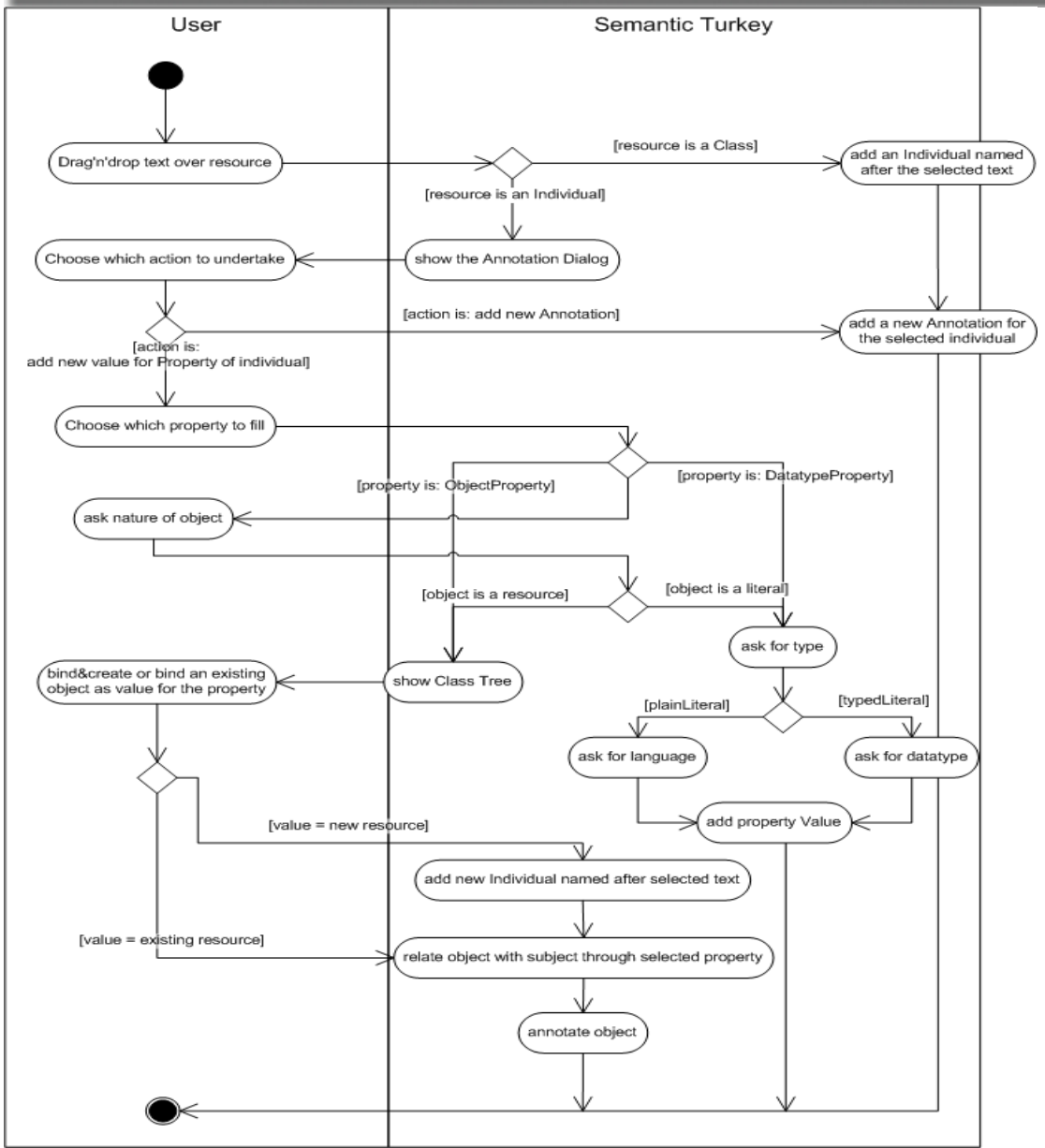
Semantic Turkey in Action: Semantic Annotation



Semantic Turkey in Action: Semantic Annotation

- No automatic ontology building from text but…
- with just one intuitive drag'n'drop operation (and few HC interactions), the system:
  - Creates a new *Domain Object* instance
    - (and/or builds a new lexicalization for the already existing instance on the annotate page)
  - Creates a new *SemanticAnnotation* instance
  - Creates a new *WebPage* instance
  - Relate all of them through dedicated properties
  - …(depending on the specific operation)

Semantic Annotation Functionalities



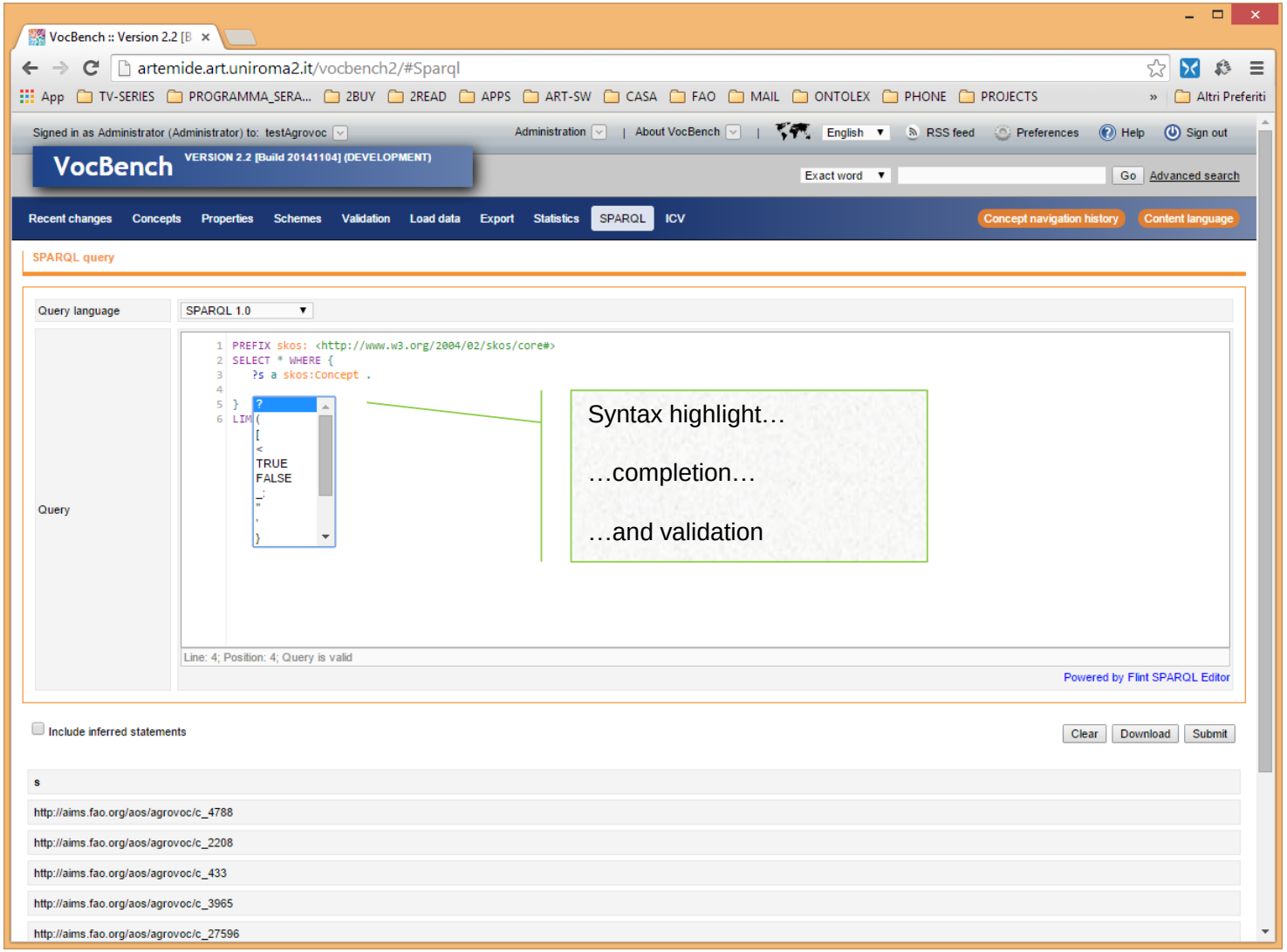
Specialized operations based on context

- Full OWL Editing
  - Individual creation
  - Web reference annotation
  - Multilingual annotations
  - Property valuing, depending on property type and range restrictions

but…

- Knowledge Acquisition must be kept simple for the user!
- Result: context-sensitive drag'n'drop based operations

Embedded SPARQL Editor



VocBench Evolution

This is a non-exhaustive list of features added along the various versions. Only the major news are reported here

VB2.2

- important change in ST: moved to the karaf OSGi container. Just run it through the bin/st_server_run batch/bash file. The new ST is totally compatible with the same data folder (SemanticTurkeyData) of the previous Semantic Turkey , so no need to update/change the existing data.
- ST Instance Manager: no more need to pass an ST connection URL (which was rather annoying to be copied and pasted from the installation instructions). The static part of the connection URL is hardcoded in the system and is always paired (and updated) with the ST version being used by the current VB, while the dynamic part can be defined only in terms of port and host. Also, through the ST instance manger, users can define multiple ST installations in terms of the sole baseURL and port and then just invoke them when defining a new project. A default ST configuration is already stored in the system, and corresponds to the default settings of ST (localhost, 1979).
- Running more projects on the same ST instance: from ST 0.10 (paired with VB2.2), it is now possible to manage more projects in parallel with the same ST instance (technical details on the ST site: http://semanticturkey.uniroma2.it). Through the ST instance manager, it is possible to associate and host different projects on ST instances, though this is no more necessary.
- User experience: New features/improvements:
  - A much requested feature: user-customizable entity id-generator (e.g. the way the name of concepts, xlabels etc.. is generated can be customized)
  - Validation for move/link/unlink actions
  - downloadable SPARQL results
  - scheme management: now possible to edit lexicalizations for schemes as well
  - Interface for checking and repairing dangling concepts
- bug fix:
  - Fixed locking of visitor only mode. In visitor mode, user can login as Adminstrator to get out of visitor mode.
  - In ST: no more projects deleted due to a lost connection to a remote repository
  - RDF Export not working
  - Removed the user/grojup management capability from the Publisher. Note this is not removed from existing installation
- New experimental resource view. To test it: login as administrator, you will see new button on top of concept tree. Select any concept and click "<s-p-o>" button near the visualization button. It will popup the resource view.
- *Developers only*: Context injectable services: now services address objects injected through an extensible "context": the context currently includes the: read and write graph(s) and the project to be considered (following the multi-project management enabled in this version)

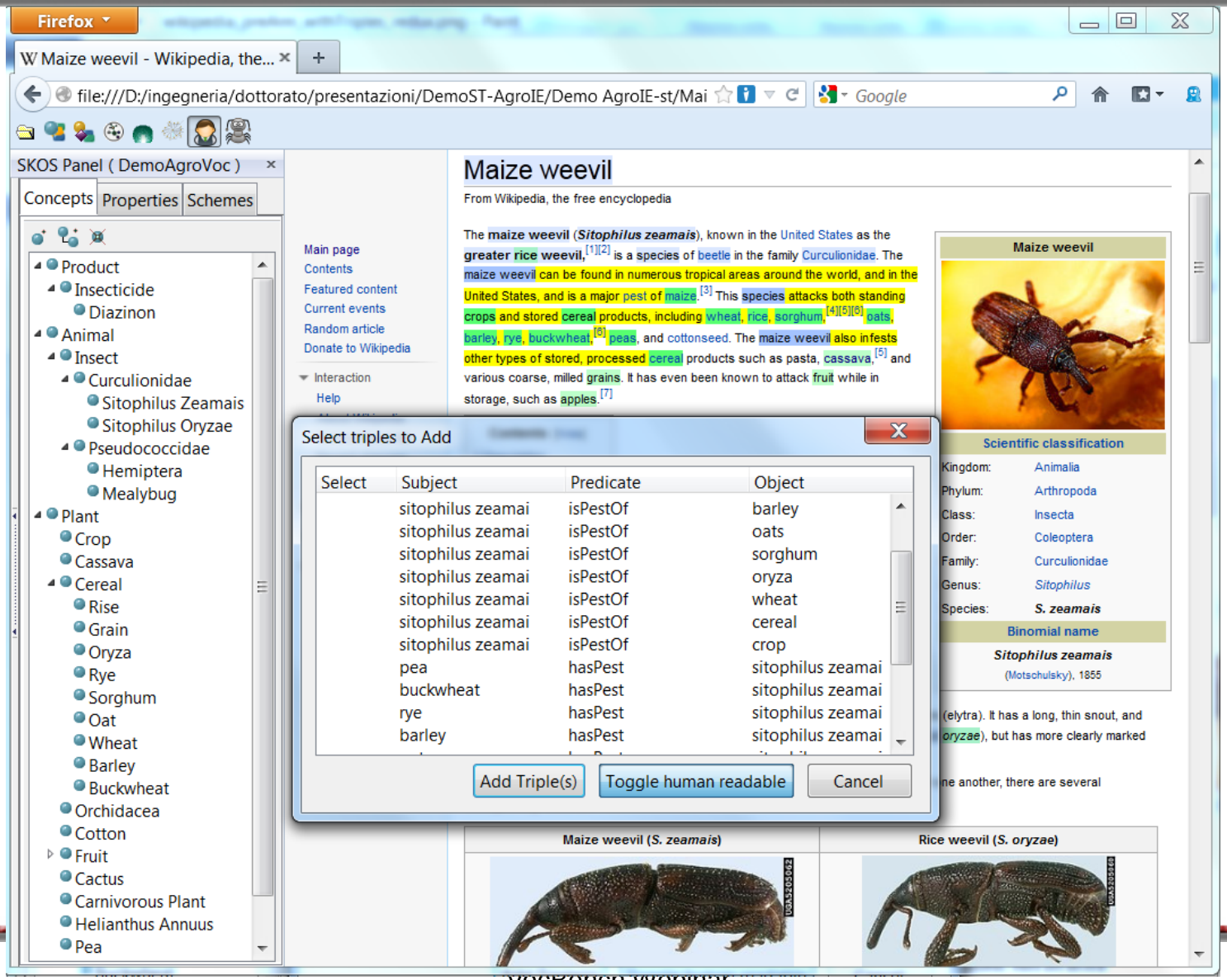
VB2.1

- A completely rebuilt installation mechanism for an headache-free installation experience!
  - Self-installing DB, with auto-updating scripts
  - Wizard-driven system configuration, with import/export of configuration profiles
- SPARQL module: query/update content directly through the SPARQL query language for RDF; syntax completion & highlight
- Multi scheme management: now concepts can be shared among different schemes
- RSS feeds for all editing actions

VB2.0

- A Completely re-engineered RDF backend, based on RDF Management platform Semantic Turkey
  - Support for different triple stores
  - Extension mechanism based on OSGi
- Multi scheme management. Several skos:ConceptSchemes can be developed for the same dataset, providing different views on the data
- Statistics module: a module providing resuming information about the loaded data.
- Export module: for exporting all or part of the content of a project according to several existing RDF serialization standards
- Load data module: for loading bulk data serialized in some RDF serialization standard
- Ontology Import Management (Administration-->Ontologies): to owl:import ontologies to be used as property vocabularies for the modeled thesauri
- New tabs under the concept view for covering extensively the SKOSXL standard (note, notations)

…and a look further beyond!



Contacts

Caterina Caracciolo caterina.caracciolo@fao.org

Armando Stellato stellato@info.uniroma2.it

VocBench site: http://vocbench.uniroma2.it/

VocBench pages@FAO: http://aims.fao.org/tools/vocbench-2/

VocBench SandBox installation: http://202.73.13.50:55481/vocbench/

You can also follow VB by registering to:

- AIMS Community Site: http://aims.fao.org/ (you can selected the topics you are interested into)
- VocBench Mailing Lists:
  - User: http://groups.google.com/group/vocbench-user
  - Developer: http://groups.google.com/group/vocbench-developer
- Semantic Turkey Mailing Lists:
  - User: http://groups.google.com/group/semanticturkey-user
  - Developer: http://groups.google.com/group/semanticturkey-developer

VocBench – features overview

- Full support to multilinguality
- Supports collaborative editing
- Formalized workflow
- Scheme and data management

- Open source

VocBench - Pointers

- VocBench Sandbox v2.2 (latest stable): http://202.73.13.50:55481/vocbench/
- VB project
- VB users mailing list
- AIMS community, for news: http://aims.fao.org/user/register
- ..