NeOn-project.org

NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 – “Semantic-based knowledge and content systems”

D8.5.2 Second Prototype of the Semantic Nomenclature

Deliverable Co-ordinator: **Germán Herrero**

Deliverable Co-ordinating Institution: **Atos Origin (ATOS)**

Other Authors: **Tomas Pariente Lobo (ATOS)**
Emmanuel Jamin (ATOS)

This document describes the implementation of the second prototype of the Semantic Nomenclature. It explains the architecture of the prototype, a brief description of the networked ontologies used, and provides an insight into the design and deployment of the software. This second version is a web based application making use of ontological knowledge to assist end-users (pharmacists, doctors) to access different data sources about pharmaceutical product and related information.

Document Identifier:	NEON/2009/D8.5.2/v1.0	Date due:	October 31st, 2009
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	October 31st, 2009
Project start date:	March 1, 2006	Version:	v1.0
Project duration:	4 years	State:	Final
		Distribution:	Public

NeOn Consortium

This document is a part of the NeOn research project funded by the IST Programme of the Commission of the European Communities by the grant number IST-2005-027595. The following partners are involved in the project:

Open University (OU) – Coordinator Knowledge Media Institute – KMi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Enrico Motta E-mail address: e.motta@open.ac.uk	Universität Karlsruhe – TH (UKARL) Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB Englerstrasse 11 D-76128 Karlsruhe, Germany Contact person: Andreas Harth E-mail address: aha@aifb.uni-karlsruhe.de
Universidad Politécnica de Madrid (UPM) Campus de Montegancedo 28660 Boadilla del Monte Spain Contact person: Asunción Gómez Pérez E-mail address: asun@fi.upm.es	Software AG (SAG) Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com
Intelligent Software Components S.A. (ISOCO) Calle de Pedro de Valdivia 10 28006 Madrid Spain Contact person: Jesús Contreras E-mail address: jcontreras@isoco.com	Institut 'Jožef Stefan' (JSI) Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si
Institut National de Recherche en Informatique et en Automatique (INRIA) ZIRST – 655 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person: Jérôme Euzenat E-mail address: Jerome.euzenat@inrialpes.fr	University of Sheffield (USFD) Dept. of Computer Science Regent Court 211 Portobello street S14DP Sheffield United Kingdom Contact person: Hamish Cunningham E-mail address: hamish@dcs.shef.ac.uk
Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: staab@uni-koblenz.de	Consiglio Nazionale delle Ricerche (CNR) Institute of cognitive sciences and technologies Via S. Martino della Battaglia, 44 - 00185 Roma-Lazio, Italy Contact person: Aldo Gangemi E-mail address: aldo.gangemi@istc.cnr.it
Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de	Food and Agriculture Organization of the United Nations (FAO) Viale delle Terme di Caracalla 1 00100 Rome Italy Contact person: Caterina Caracciolo E-mail address: Caterina.Caracciolo@fao.org
Atos Origin S.A. (ATOS) Calle de Albarracín, 25 28037 Madrid, Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientelobo@atosorigin.com	Laboratorios KIN, S.A. (KIN) C/Ciudad de Granada, 123 08018 Barcelona, Spain Contact person: Antonio López E-mail address: alopez@kin.es

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed writing parts of this document:

ATOS

Change Log

Version	Date	Amended by	Changes
0.1	2009-10-31	Germán Herrero	Initial version of deliverable document
0.2	2009-10-31	Germán Herrero	Section 1 added
0.3	2009-11-01	Germán Herrero	Sections 2, 2.3, 2.4 added
0.4	2009-11-01	Germán Herrero	Sections 2.1, 2.2, 3.2 added
0.5	2009-11-29	Germán Herrero	Section 3.3 added, Section 3.2 updated
0.6	2009-11-30	Germán Herrero	Sections 4, 5 added
0.7	2009-11-30	Germán Herrero	Section 6 added, Index and table of figures and tables updated
0.8	2009-11-30	Emmanuel Jamin	Section 3.1, 3.4, 3.5 added, Index and table of figures and tables updated
0.9	2009-11-30	Emmanuel Jamin	Section 3.6 added, Index and table of figures and tables updated
0,95	2009-11-30	German Herrero	First draft sent to QA
1.0	2009-12-09	Tomás Pariente	Internal QA and final draft sent to QA
1.1	2009-12-22	Tomás Pariente	Final draft. Revision after QA

Executive Summary

This document describes the design and implementation of the second prototype of the Semantic Nomenclature.

The evolution of the Semantic Nomenclature prototype from the previous version is significant from two perspectives. The first prototype was delivered as a J2EE Web application using Watson as the backbone, while the current prototype is Ajax-based (using the Google Web Toolkit framework) and uses more NeOn services in the business and back-end infrastructure. The second main difference is that the knowledge base delivered on the first prototype was still using a set of non-connected ontologies, while the current version uses a more enriched and real network of pharmaceutical ontologies, covering a range of functionalities more focused on the case study requirements.

The document explains the architecture of the Semantic Nomenclature, the networked ontologies used, and provides an insight into the design and deployment of the software developed.

It is foreseen that during the evaluation of the prototype some enhancements will be delivered. This document goes together with the software developed for this second prototype as a part of task 8.5.

Table of Contents

List of Figures

List of Tables

1 Introduction

This document describes the design and implementation of the second prototype of the Semantic Nomenclature system. The document explains the architecture of Semantic Nomenclature, the networked ontologies used, and provides an insight into the design and deployment of the software developed.

The Semantic Nomenclature prototype is supported at the infrastructure level with a knowledge base (KB) which provides relevant information about pharmaceutical products and associated knowledge about other types of entities like active ingredients, diseases, laboratories, etc. This prototype provides a KB consisting of several networked ontologies on the pharmaceutical products domain. Some of the ontologies have been designed using the NeOn Methodology while others have been simply added to the network from existing ontologies. The knowledge base is populated with original data available from the main pharmaceutical product database (Digitalis) from the Spanish Government and from the public information provided by the GSCOP in its database (BOTPlus). The conceptual structure of the KB is based on the ontologies reengineered from the RTMS databases and the main application ontology is developed following the requirements and recommendations detailed by domain experts and other domain approaches (Snomed). For more information about the ontology network check [1].

1.1 Differences to the first Semantic Nomenclature Prototype

The evolution of the Semantic Nomenclature prototype from the previous version [1] is significant, both from the composition of the knowledge base and the infrastructural points of view. The major changes are the following:

- The first prototype was delivered as J2EE Struts-based Web based application,

  The current version of the prototype is also a Web-based application, but in this case is Ajax-based, using the Google Web Toolkit (GWT)[1]. This architectural decision gives more development flexibility and improves the user experience, and it is more in line with the recommendations for a NeOn runtime architecture given in deliverable [2] At the server-side we update the back-end from Watson to Cupboard for repository and query services.

- The ontologies used in the first prototype were still not connected (networked).

  The second prototype is using several ontologies that form a real network and provide different kinds of knowledge to the overall knowledge base. The second prototype is using a combination of OWL ontologies and A-Box populated into RDF triples

- More use of NeOn Runtime Services,

  In this second prototype we added more NeOn runtime Services to the application. As the project has been evolving in the last months, more NeOn runtime services are provided and were added to the prototype those which are adequate to solve some functionalities or requirements expected from the prototype

- The first prototype covered a limited set of the most important requirements.

[1] http://code.google.com/webtoolkit/

For the second prototype we are focussing on provide a new set of functionalities to the end-users and a enriched functionality of search information.

It is foreseen that during the evaluation of the prototype some more enhancements will be delivered and reported back along with the evaluation report.

1.2 Structure of the document

In this document we first present in Section 2 the main goals, requirements and knowledge base of the Semantic Nomenclature. The design and architecture of the current release of the Semantic Nomenclature prototype is explained in Section 3. After some conclusions and the list of references (Sections 4 and 5), we include as appendix a brief user manual of the graphical user interface of the application.

2 Goals, Requirements and Knowledge Base of Semantic Nomenclature

2.1 Prototype Scope

The main goal of the second prototype of the Semantic Nomenclature case study is to provide a NeOn-based Nomenclature via Web, in order to allow a search of products over the ontology network and provide collaborative and social functionalities to the user to improve the ontology network. This goal may be divided in several sub-objectives:

- Fine-tune the Semantic Nomenclature ontology network.
- Develop a Web application on top of the Semantic Nomenclature ontology network that allows users to navigate and improve the network
- Show the current status of the NeOn technology to develop web-based applications

The original requirements document [3] and the original design document [4] for Semantic Nomenclature use case define a number of use cases and requirements to inform the design and development activities. Some of them have been implemented in the first prototype, which has been evaluated. This evaluation mainly points to weaknesses in the presentation of the functionality, the general look and feel and also the weakness of use of networked ontologies. In [3] several scenarios of usage of the Semantic Nomenclature were proposed. The majority of the proposed scenarios could not be achieved in the first version of the prototype, because they relied on the availability of mappings between ontologies, which was not available by the time the first prototype was delivered. The current iteration in the implementation of the Semantic Nomenclature improves the feature-set of the first prototype of Semantic Nomenclature [1] in a number of dimensions:

- More accurate query answering mechanism
- Access to more data (new A-Box)
- More collaborative and social functionalities
- Usage of multiple, networked ontologies

The Semantic Nomenclature prototype is an eye-catching for the pharmaceutical community as a new nomenclature (compendium) based on semantic web technologies. The prototype is targeting mainly pharma knowledge experts. The main result is focused on the runtime aspects of NeOn, but it relies on the work done at the design time on the ontologies using the NeOn Toolkit, different NeOn plugins and the NeOn methodology. Apart from being a testbed of the NeOn runtime services, the goal is to offer a view over a set of networked ontologies, allowing functionalities such as querying, adding new ontologies to the network, rating of ontology elements, or adding new ontology mappings.

2.2 Requirements

The main requirements of the Semantic Nomenclature case study are described in [3] and [4]. Although there is a lack of international standard to describe drugs, during the lasts years we have been witnesses of the increased attention of the health domain towards what is called Semantic Interoperability in eHealth. One of the pillars to achieve semantic interoperability in this domain is precisely having a common or interoperable description of drugs. There are efforts such as the widely usage in certain countries of Snomed CT[2], HL7[3].

[2] http://www.ihtsdo.org/snomed-ct/

More important for us, is the widely spread idea that ontologies are a very useful way to describe drug models. Initiatives such as BioPortal[4], where access to the most widely used ontologies in the biological community can be found, the OBO Foundry[5], which states a language and a set of principles for creating biomedical ontologies, or the recommendations from roadmaps such as the one delivered by the SemanticHEALTH[6] project are clear examples that ontologies are becoming an undeniably way to describe the entities of the Health domain, and particularly the description of the drugs. However, it is not foreseen that a new standard for drugs description will appear in the future. Nevertheless it is also clear that isolated formal descriptions are not useful when talking about semantic interoperability. Mappings between different descriptions should be made in order to achieve interoperability. In is here where NeOn, and particularly the approach developed within this case study could be helpful.

There is also a different angle to consider, which is related with the Open Linked Data initiative. Linked Data[7] is a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF. Among the current Linked Data datasets up to today there are several health-related open resources such as Drugbank[8], Dailymed[9] or Diseasome[10], and also several other generic-purpose resources containing information about drugs, such as DBPedia[11]. Although using Linked Data is not a requirement for this case study, we also included in the current prototype a possible link to make use of Linked Data from the Semantic Nomenclature application and a small attempt to allow creating mappings between our ontologies and some of the Linked Data datasets. This is nevertheless a work in progress just to show a possible path for future enhancements.

For practical terms, it is a requirement in NeOn to base our approach in using or reengineering OWL ontologies.

2.3 Knowledge Base – Ontologies

This second iteration of the Nomenclature Ontology Network is organized in four levels, as the first iteration: the Representation Ontology (OWL), General Ontologies, Domain Ontologies and the Application Ontologies. However, in the first iteration we did not include mappings between the ontologies, whish is no longer the case.

At the domain level are located ontologies that define the main notion and concepts of the pharmaceutical domain. In this level ontologies from the main standard terminologies or vocabularies in the eHealth domain might be added to the network, such as Snomed CT, Mesh or NCI. At this level we also include ontologies providing a classification of pharmaceutical terms, such as the ATC[12] classification (WHO recommendation mapped to

[3] http://www.hl7.org/

[4] http://bioportal.bioontology.org/

[5] http://www.obofoundry.org/

[6] http://www.semantichealth.org/

[7] http://linkeddata.org/

[8] http://www.drugbank.ca/

[9] http://dailymed.nlm.nih.gov

[10] http://diseasome.eu/

[11] http://dbpedia.org/

[12] http://www.whocc.no/atcddd/

many terminologies) or the SPC model [13](Summary of Product Characteristics), modelled as an ontology in NeOn from a physician's information document issued by the European Union.

At application domain are classified the ontologies which represents the knowledge of the real-world resources after being re-engineered as ontologies. In the Semantic Nomenclature case study we have included as a matter of example two resources widely used in Spain with information about drugs: The Digitalis ontology, reengineered from the governmental public database of marketed pharmaceutical products, which is the de-facto Spanish standard for pharmaceutical products available in the Spanish market; and the BOTPlus ontology, modeled from the set of public information provided for this nomenclature of Spanish pharma products.

The Semantic Nomenclature Application Ontology plays a central role of being the main façade of the ontology network, mapping different ontologies at a conceptual level. In Figure 1 the current Semantic Nomenclature Ontology Network used by the prototype is depicted. The domain ontologies allow us to connect our application resources to international health terminologies. The Semantic Nomenclature Application ontology is enriched with the other application ontologies connected via mappings (Digitalis or BOTPlus) and with the most important classification system domain ontologies involved in the Semantic Nomenclature scenario as the ATC. Moreover, as was described in [5], in this ontology network are added more resources as SPC, RxNorm or DrugOnto that enrich the ontology network.

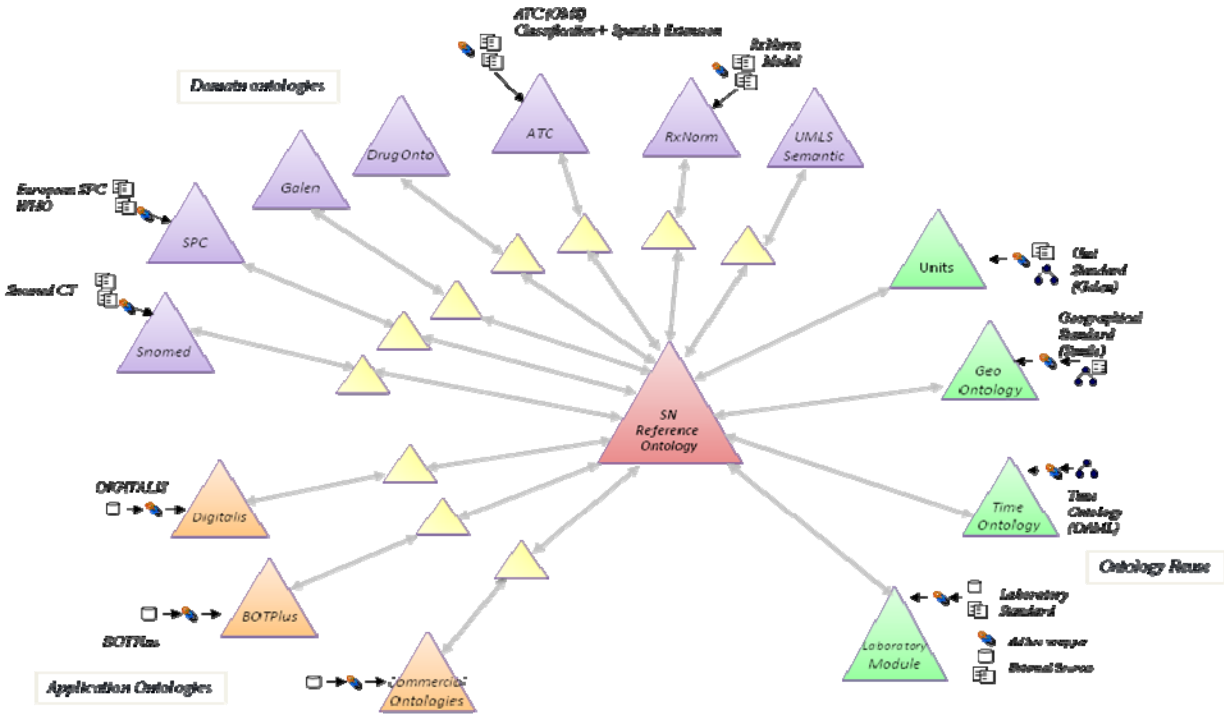


Figure 1: Nomenclature Ontology Network

One of the objectives of this network is to allow an easy mapping to new ontological resources. This can be done at conceptual level using the Semantic Nomenclature Application ontology as gateway, or at instance level, by creating alignments between instances of the ontology network. This new version of the Semantic Nomenclature

[13] http://ec.europa.eu/enterprise/pharmaceuticals/eudralex/vol-2/c/spcguidrev1-oct2005.pdf

Ontology Network should evolve in more iterations of the lifecycle model, where new resources or ontologies that could appear related with medical vocabularies used in the world are added and integrated. These ontologies may come from the current stakeholders (as ontologies of laboratory products, hospital ontologies) or external ones (ontologies from other countries or similar domains) or new standards or recommendations in the Health domain. This could be the case of the Translational Medicine Ontology (TMO)[14], which is the ontology developed by W3G Health Group to allow the integration of data throughout the drug development process.

With this new version of the ontology network we maintain the expectations of the first version, so is aligned with the goals extracted from the case study scenarios described in [3]. Also, in this case, with the review of the main goal and knowledge described in the Semantic Nomenclature Application Ontology, we achieve the new requirements proposed by the domain experts for cover the ambiguity between clinical and branded drugs.

This ontology network makes possible the easy interoperability and integration of the distributed resources for the description of pharmaceutical products. Moreover, the ontology network facilitates the aggregation of drug-related information in a semantic way because the Semantic Nomenclature Application is connected via mappings, stored as ontologies which are part of the knowledge, with different pharmaceutical ontologies at different levels. This solution makes possible the collection of information for concrete products and maintains the legacy databases updated, because the pharmaceutical product information gathered in the networked ontologies give an added value to the actors. Based on the ontology network, the actors can improve their commercial database and reduce their effort in complementing typical pharmaceutical compendium characteristics by giving flexible, extensible and reliable information about drugs to the users of the Pharmaceutical domain.

The following table is a summary of the ontologies involved in the Semantic Nomenclature prototype and its purpose:

Ontology	Type of Ontology	Motivation / Relevance
Semantic Nomenclature App Ontology	Application Ont.	The Semantic Nomenclature Application Ontology is the core of the ontology network used in the case study. This ontology has three main goals: act as a bridge between the different application ontologies and domain ontologies; the second goal is implement one the new requirements as the disambiguate between the clinical drug/branded drug; the third functionality is act as the application ontology for the Semantic Nomenclature prototype, because from this ontology, using the mappings and relations can be accessed the rest of the ontology network. The Ontology is based on the main recommendations provided by the pharmaceutical product, and also using the semantic model of Snomed as background knowledge, mainly from the Pharmaceutical/Biological product term used in the terminology.
Digitalis Ontology	Application Ont.	The knowledge model represented in the schema of the database Digitalis is re-engineered in an ontology. The main concept is Pharmaceutical_Product that could be the point of link with the Nomenclature ontology. This link is possible via a mapping between Digitalis ontology and the Nomenclature Ontology. Other classes represent the main concepts extracted from the tables of the DigitalisDB and the relations represented in their schema model are used to describe with more detail the information around the marketed product and its use. This Ontology provides one of the main A-Box information to the application

[14] http://esw.w3.org/topic/HCLSIG/PharmaOntology

Ontology	Type of Ontology	Motivation / Relevance
BOTPlus Ontology	Application Ont.	The BOTPlus ontology gathers the knowledge represented in the schema of the BOTPlus database. As in the Digitalis ontology, the BOTPlus ontology captures more data than the marketed product information, as information about interactions, pathology, active ingredients. BOTPlus provides the most bigger A_Box to the system
ATC Ontology	Domain Ont.	the concept ATC_Classified_Product represents all the pharmaceutical products classified through the ATC code. This conceptualization of the hierarchy allows inference over the ontology model and obtains the therapeutical, anatomical, pharmacological or chemical group of one determinate pharmaceutical product from its ATC code.
RxNorm Ontology	Domain/App Ont	RxNorm is a standardized nomenclature for clinical drugs, produced by the National Library of Medicine (NLM). RxNorm contains the names of prescription and many non-prescription formulations that exist in the United States.
SPC Ontology	Domain/App Ont.	The SmPC is the basis of information for healthcare professionals on how to use the medicinal product safely and effectively. This guideline will be included in The Rules Governing Medicinal Products in the European Union Volume 2C Notice to Applicants. We re-engineered the model as a semantic model
UMLS Ontology	Domain Ont.	UMLSKS provides access to multiple knowledge sources in the medical domain (SNOMED included). The purpose of the UMLS Semantic Network is to provide a consistent categorization of all concepts represented in the UMLS Metathesaurus and to provide a set of useful relationships between these concepts. All information about specific concepts is found in the Metathesaurus. The Network provides information about the set of basic semantic types, or categories, which may be assigned to these concepts, and it defines the set of relationships that may hold between the semantic types. This ontology will allow us connect our resources to the UMLSKS system
Drug-Ont Ontology	Domain Ont.	The LSDIS lab's15 collaborative research project on Active Semantic Electronic Patient Record has provided different populated ontologies in the healthcare domain, and the DrugOnt schema is related with drugs, including concepts such as indications, interactions, formulary, etc.
MeSH Ontology	Domain Ont.	MeSH is the National Library of Medicine's (USA) controlled vocabulary thesaurus. It consists of sets of terms naming descriptors in a hierarchical structure that permits searching at various levels of specificity.
Translational Medicine Ontology	Domain Ont.	
Galen Ontology	Domain Ont.	The GALEN ontology is a result from the OpenGALEN Foundation16 (a non profit organisation). The main goal of the ontology is provide terminology and classifications related with the anatomy, surgical deeds, diseases, and their modifiers used in the definitions of surgical procedures. Despite Galen is old-fashioned, it was used in different healthcare systems.

[15] http://lsdis.cs.uga.edu

[16] http://www.opengalen.org/

Ontology	Type of Ontology	Motivation / Relevance
NCI Ontology	Domain Ont.	NCI thesaurus is a terminology and biomedical ontology used in a growing number of NCI and other systems. It covers vocabulary for clinical care, translational and basic research, and public information and administrative activities.
Disease Ontology	Domain Ont.	Disease Ontology is a controlled medical vocabulary developed at the Bioinformatics Core Facility in collaboration with the NuGene Project17 at the Center for Genetic Medicine18. This ontology is not key in the case study because is not about pharmaceutical product description, but we think that can improve the ontology network, due we can include and describe relations between pharmaceutical products, active ingredients or therapeutic use with the diseases described in the ontology

Table 1: Relevant resources involved in the prototype

2.4 Target User Groups

The users of the Semantic Nomenclature Web prototype will be mainly pharma knowledge experts with a limited knowledge of ontologies. It is not intended for users with no knowledge about ontologies at all, but they do not need to be ontology experts to get benefits from using the prototype. But the prototype is not closed to these domain actors, it will be open to any other kind of users, as people who want to retrieve a semantic enriched information about pharmaceutical products. Moreover, the prototype provides functionalities to biomedicine, pharmacy or healthcare researchers to assess about the models of the ontologies used by the prototype or add new models or discuss with other colleagues.

[17] http://www.nugene.org/

[18] http://www.cgm.northwestern.edu/

3 Prototype Architecture and Design

3.1 Overall Architecture

The implementation objective consists on integrating the ontology network presented before in a user-friendly Web application. We clearly separated the implementation between a presentation layer at the client side and business processing at server side. On the one hand, the server is essentially dedicated to data processing and management of the functional process, and its architecture is generic to interact with several software components provided by NeOn or other 3rd parties. On the other hand, the client side is dedicated to the information presentation and the data interaction. The prototype is using GWT as an Ajax-based framework. The software integration was facilitated by the communication mechanism provided by the GWT technology. For example, the RPC mechanism permits to interoperate with different kind of software components as a semantic repository, or the Neon plugins through Web Services. Moreover, the JSON communication between the server and client sides allows the exchange of structured data.

This approach was much appropriated to differentiate the generic implementation (1, and 3 part of the Figure 2) and the specific application implementation (2, and 4 parts). It will also be very a very efficient approach to extend the Web application with more Web 2.0 functionalities (as, social networking, geo-localization, etc.)

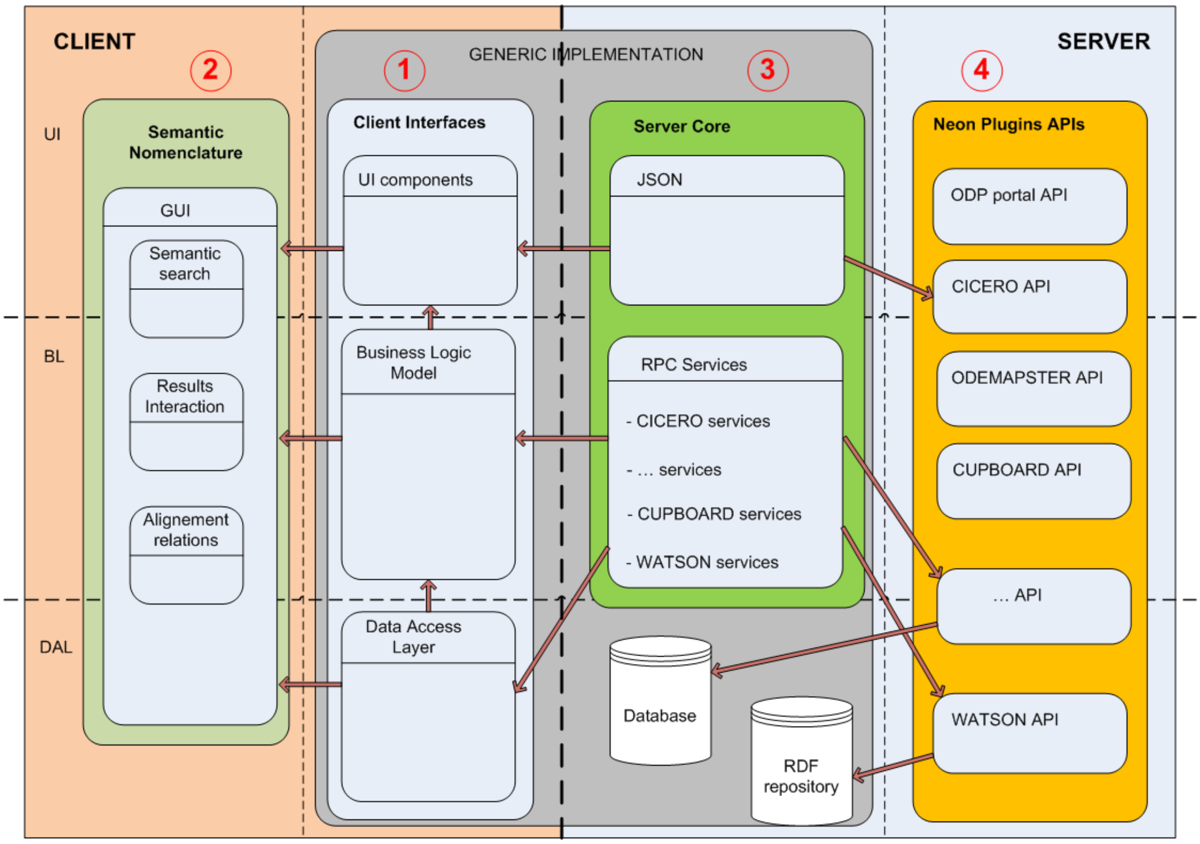


Figure 2: Semantic Nomenclature Prototype v2 Architecture

The server side is composed of the Server Core and of the integrated modules (essentially the Neon plugins, but also a Sesame[19] semantic repository in order to show the extensibility

[19] http://www.openrdf.org/

of the solution). The Server Core contains all the modules interfaces that have to be integrated. For example, the cupboard interface is integrated as a jar module in the application project, and permits to interoperate with the available Web Service (start a SPARQL query).

The client side is composed of all the graphic components, the related data model of each one, and the interaction procedure. The main part of the generic development is the model of the handled data (for example, a list of triples). The data models are much closer to the interface results, and will be used by the graphic components. Finally, the rest of the client side is related to the widget management (query, or result panels for example).

3.2 Population of the Knowledge Base

The process of populating the knowledge base with the databases content was done manually. In a first approach we used the ODEMapster plugin, but we found some problems with the language used (UTF-8 against ISO-8859-1) and also that some parts of the ontology were difficult to map using the R2O links. The problem of the codification of the file generated with the A-Box is due to the special characters used in Spanish like "ñ" and also, the most typical characters as "&" which are not supported by UTF-8 used in the ODEMapster engine. These characters are typical in several labels and characteristics in different instances of pharmaceutical products, laboratories, descriptions, ingredients, etc.

Based on this, we developed two ad-hoc wrappers for populate information from Digitalis and BOTPlus respectively. Since not all the information must be extracted, the process of extracting the information and populate them as semantic information into RDF triples was develop ad-hoc wrappers that access to the databases and generates instances based on the ontology models. In both cases, Digitalis and BOTplus, the wrappers update each A-Box with the last information of the databases.

In the population process is key for the prototype that the generated instances in both A-Box must have significant and meaningful labels, because at the UI of the prototype are these labels displayed in the different widgets.

The generated A-Box is around 250MB of instances with information of pharmaceutical products and related entities as active ingredients, ingredients, laboratories, symptoms... This generated A-Box are stored in separated files from the model and loaded in the Semantic Nomenclature Ontology Space in Cupboard, as part of the data back-end.

3.3 Server Layer

The server layer implements the business logic (logic tier) of the web application. For technical details abut the implementation of this layer, see appendix 6.1.

The back-end of the Semantic Nomenclature prototype consists on a business layer that uses Cupboard for managing the ontologies described before (ontology registry and repository) and a RDF repository that manages the different alignments between the entities involved in the Nomenclature.

Cupboard is an online system for sharing and reusing ontologies linked together with alignments, and that are attached to rich metadata and reviews. It is therefore used in the current prototype as a core infrastructure component to explore, query and exploit ontologies. Cupboard does not provide one single space where ontologies are exposed. On the contrary, each user can create its own ontology space, containing and relating the selected ontologies. Cupboard provides a number of advanced functionalities on top of these ontology spaces to facilitate the management, sharing and reuse not only of stand-alone ontologies, but also of networks of ontologies connected through alignments. [4]

In our scenario, we use Cupboard as the underlying core infrastructure to explore ontologies, query them, exchange them and exploit them. A new Ontology Space in Cupboard called 'SemNomenclature' has been created, where the main ontologies of our case study (T-Box, A-Box) have been registered.

This back-end component of Semantic Nomenclature provides to the upper tiers of the architecture to:

- Manage the Semantic Nomenclature Knowledge Base in the 'SemNomencalture' ontology space
- Query the ontology space with SPARQL
- Management and access of the OWL ontologies via the Cupboard Web Service.

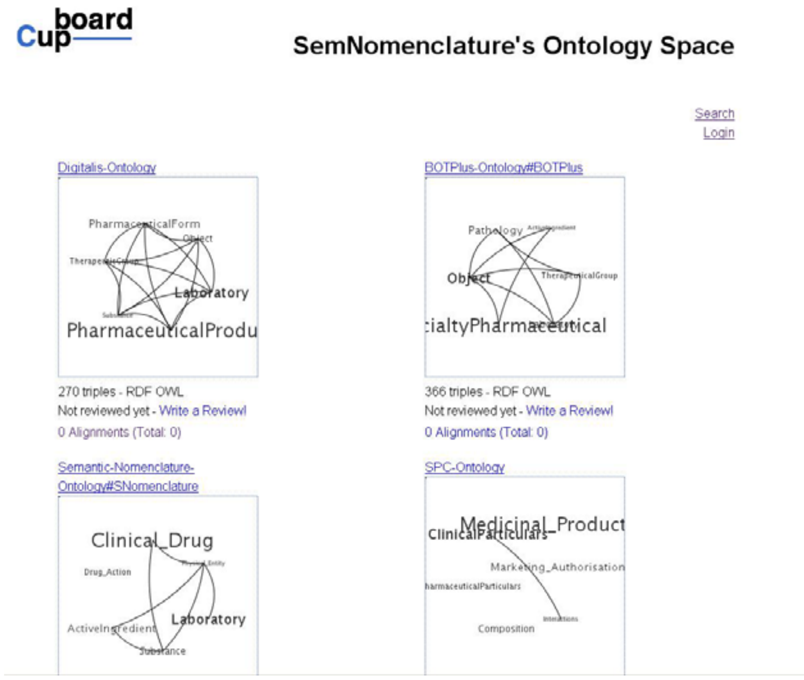


Figure 3: SemNomenclature Ontology Space in Cupboard[20]

The implementation aspects and all configuration details are hidden behind the Cupboard WS-Client that encapsulates the access to Cupboard (deployed in a KMI server) and abstracts away from the Cupboard implementation details. The WS-client used in the prototype is the version 1.0. The main operations used in the Nomenclature prototype are showed in the next table:

Operation	Description
getUrnSemanticContentSearch	Service for searching and inspecting semantic documents
executeSPARQLQuery(ontology, query)	Executes the SPARQL query against the ontology stored in the ontology space
OntologySpaceService(OntologySpace ID, Password);	Access to one ontology space in Cupboard using the password
listOntologies()	Returns a list of ontology uri's of the ontology space

Table 2: Cupboard WS operation used

A detailed overview of the Cupboard WS description for managing the ontology space is presented in [4].

[20] http://cupboard.open.ac.uk:8081/cupboard/SemNomenclature

Our main goal in the design process of the prototype architecture is the use of the online space where we can upload the network of ontologies, describe metadata about them and add alignments between them. Moreover, the prototype takes advantage of the sophisticated searching mechanisms to find ontologies in our/other ontology spaces and review mechanisms to assess the ontologies according to 5 different dimensions (reusability, coverage, complexity, modeling and correctness). Cupboard is used in the prototype as a virtual ontology infrastructure for each ontology space.

The Semantic Nomenclature includes a second repository that manages the different mappings (at conceptual and individual level) of the case study ontology network. The current implementation of the repository of alignments is based in Sesame. The use of Sesame is only done as an implementation decision to allow storage of alignments in a dedicated RDF repository. In this sense, it could be easily replaced by other repositories, such as the Alignment Server, or even Watson In this repository the alignments coming from the Alignment Server and also the candidate mappings done manually by the end-users are uploaded. The application accesses using SPARQL both to Cupboard and the alignment repository to present the results to the user.

In the next figure are depicted the different services in the Semantic Nomenclature prototype and the associated interfaces and classes.

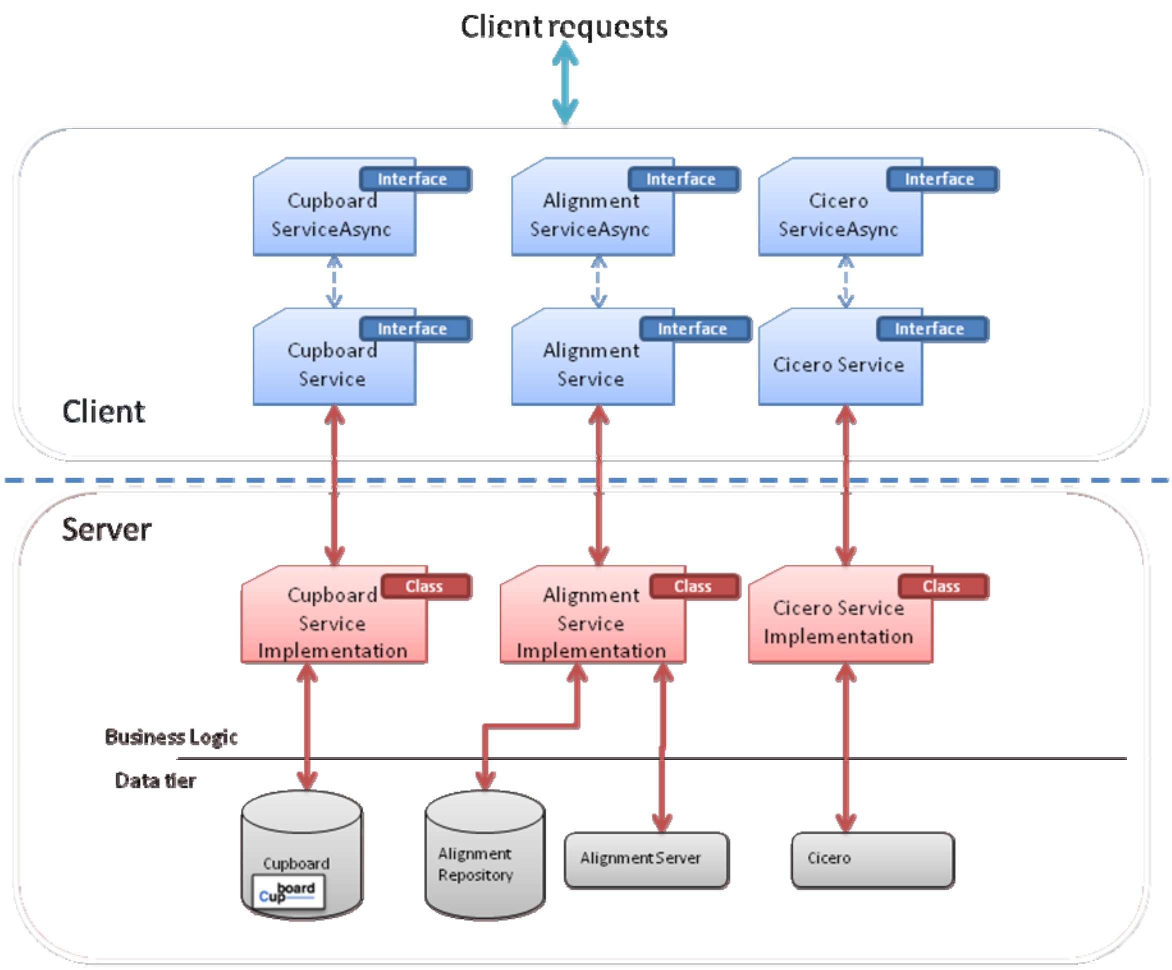


Figure 4: RPC Semantic Nomenclature

The current version of the prototype allows searching information related with the Nomenclature instances in Linked Data, and more specifically with Linking Open Drug Data

(LODD). Among the current LODD datasets up to today we have selected Drugbank[21] as a proof of concept. Using LODD is not a requirement for this case study, but we included in the current prototype a link to make use of LODD resources from the Semantic Nomenclature application and a side functionality to create mappings between our ontologies and Drugbank resources. The triples generated are stored in the local repository and showed to the end-user in the aligned results as candidate alignments. This is nevertheless a work in progress just to show a possible path for future enhancements. Therefore aspects such as how to maintain consistency between (external) Linked Data resources and (internal) local ontologies and who (what user profile) should be in charge of keeping such consistency are not addressed in the prototype. For more information about the use of LODD in the prototype check appendix 6.2.

3.4 Presentation Layer

3.4.1 Overview of the GUI structure

The Presentation Layer is a Web application that allows the users to interact with the Semantic Nomenclature ontology network on the Web.

The core of the presentation layer is focused on querying the medical ontologies of the Semantic Nomenclature. However, some generic functionalities of any web application have been also being included. The user can log in the application using a personal account (OpenID, or Cupboard account), and the system will keep track of their activity. The personal identification allows managing the user session, and saves relevant resources.

The entire GUI is AJAX-based technology. It means that the graphical design is developed on JS widget provided by GWT, which can be enriched by HTML code, and formatted by the CSS code.

- The HEAD of the application defines the application with the Neon logo, the application title and a widget to allow the user identification in the system. The login widget authorizes the interaction of the user with the application within a personal session.
- The main part of the application is the knowledge interaction panel. Located at the central area, it is mainly composed of the query widget (ontology, and concept selection) and of the result panel. Those panels are mainly composed of grids designed to select or interact with triples data. For example, the selected concepts of a specific ontology can be chosen to define a SPARQL query.
- The left area is dedicated to the personal data. To select and save the most relevant resources, the user dispose of a personal bookmark, Here other tools such as feeds browser, or Neon tools are available to help the user to follow the ontology evolution and to organise his own data.
- The right area is a browsing area through resources from the web that are related to the retrieved ones in the semantic nomenclature. The tools can help the user to enrich the ontology network. It is in two different widgets. The first one is a browser that displays the possible concepts mapping detected by the Alignment server. And the second is a browser that looks for similar resources on external resources web sites (as the DBpedia, or Linked of Drug Data).
- Finally, the south area serves to log the user activity and display his activity history. It is very helpful to retrieve interesting steps of his work session.

[21] http://www.drugbank.ca/

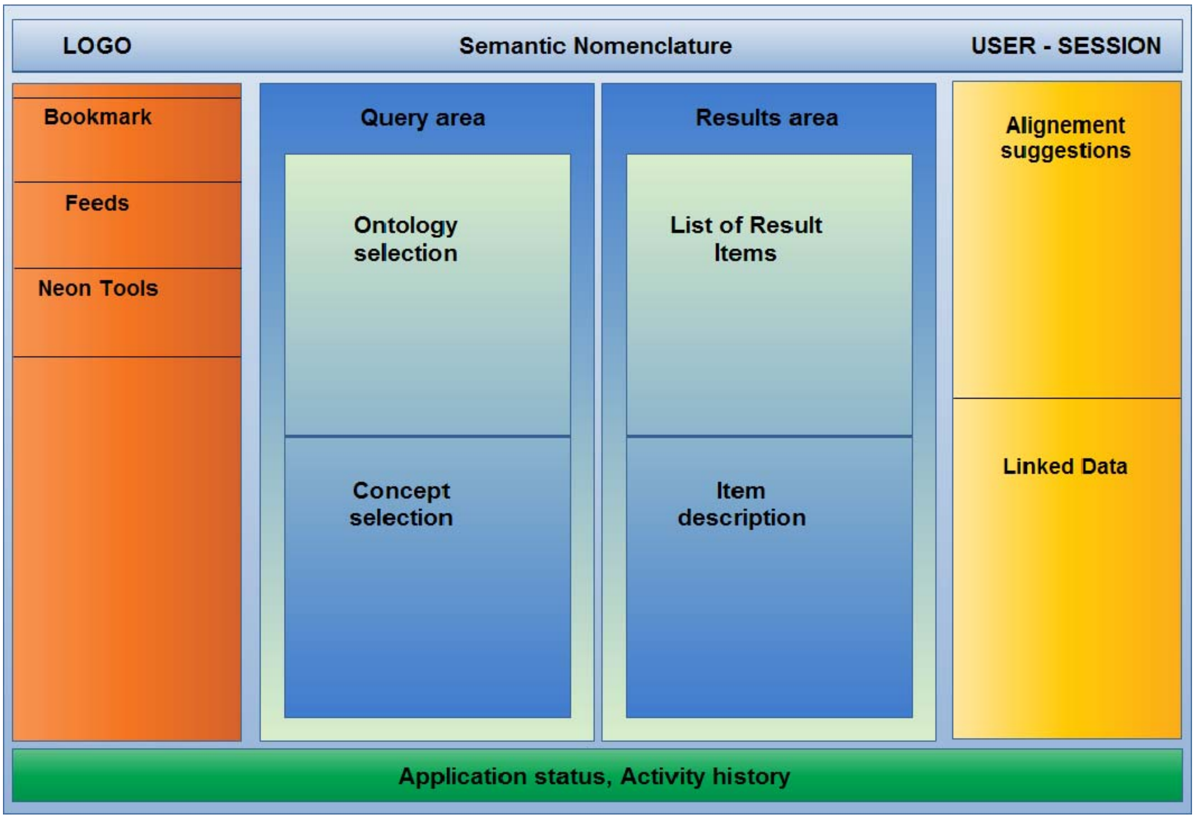


Figure 5: Semantic Nomenclature web application GUI sections

3.4.2 GUI components of the Semantic nomenclature

In the GUI architecture, specific graphical components have been implemented. They are based on the components provided by the GWT-Ext[22] library. The most important are the following:

- The OntologyComboBox widget (in the OntololySelectionPanel): is a ComboBox which contains the list of the available ontologies in the semantic network. It permits to select the main ontology that is concerned by the user search.
- The SemanticItemGrid widget (in the ConceptSelectionPanel) is composed of the list of all the concept of the primary ontology. The interesting concepts can be selected and will be considered to formulate the query.
- The SearchButton (in the QueryPanel) get the values of the ontology selection and the concept selection to generate a SPARQL query that will be sent to the cupboard engine.
- The SemanticItemGrid widget (in the ResultItemsGrid) is the list of the query results. This component displays the different triples that have been retrieved. For each result item, two buttons are associated. One permit two display the alignement suggestions, and the other permits to browse the related resources available on the web.
- And the DescriptionItemGrid (in the ItemDescriptionPanel) is a grid that displays the complete description of a selected item in the result list. It lists the diffentent properties of the resource and the related value.

[22] GWT-Ext library: http://gwt-ext.com/

3.5 NeOn Runtime Services

As explained before (section 3.1), the Neon plugins are integrated in the Server Core thanks to their interface specification. In the case of Web Service (Cicero, Cupboard), the interface is defined as a WSDL file that describes the different methods available by the module. And in the case of a jar component, it is composed of an API that provides public methods of the component. The next table is the list of the different Neon plugins that have been integrated in the semantic nomenclature prototype.

Neon plugins	Description	Integration
Cicero	Wiki, collaborative tool for knowledge management	WSDL
Cupboard	Semantic repository	WDSL
Alignment	Conceptual correlation, suggestions in the retrieval process	REST interface

Table 3: NeOn Runtime Services - S. Nomenclature prototype

Whatever the integration technology, the Neon plugins are all defined in the Server Core as services (RPC) to interoperate with the client side. Now a description of the services is described:

The Services provided by the Alignment Server are the following methods:

- `String getAlignments(String ont1, String ont2);`
- `List<EntityItem> getAlignments(String resource);` retrieves the list of the possible mappings from a specified concept.
- `List<EntityItem> getEntityAlignDetails(String resource);` retrieves the details of the alignment items.

The Cicero Services are:

- `String createIssue(String ontElement, String issue);` permits to add an issue on a specific resource.
- `String getIssue(String ontElement);` retrieves the issue related to a specific resource.

The Cupboard is used as a semantic repository. Their services are mainly related to semantic search functionalities, and especially to SPARQL functionalities.

- `String getOntologySpace (String uname, String password, String ontspace);` permits to connect the user on the specified ontology space.
- `String Sparql (String ont1, String sparql);` is a generic method the send a SPARQL query against a specified ontology.
- `public List<EntityItem> getConceptProperty(String cpId, String url);` is a query service that retrieves all the concepts defined in a specific ontology.
- `public String singleSparql(String ont1, String concept, String prop, String value);` is a SPARQL service that retrieve the resource that match with the specific subject, property and value in a specific ontology.
- `public List<EntityItem> getEntityDetails(String resource);` is a SPARQL service that retrieve the full description of a specified resource.

- `public List<EntityItem> startSingleSparql (String ont1, String concept, String prop, String value);`
- `public List<EntityItem> simpleSparql (List<EntityItem> userinput, String url, int i);`

Using the same integration mechanism, other Neon plugins might be easily integrated in the such as the visualization and SearchPoint plugins.

For a more detailed explanation please refer to the Appendix 6.3 where a description of the semantic nomenclature application is provided.

3.6 Current user functionalities of the Web prototype

The web application provides to the end-user different functionalities to interact with the semantic nomenclature. As the ontology network is complex, users have first of all to select an ontology as their primary or base ontology. This primary ontology will be considered as a reference by the user, and from its relations with other ontologies it will allow to express SPARQL queries to other ontologies in the network. In fact, other functionalities to get conceptual similarities and browse through related knowledge are useful to get a better view of the ontology network.

Below there is a summary of the main functionalities offered by the Web application:

- Select a primary ontology: From the Ontology ComboBox, the user can select his primary ontology. This ontology will be the reference to do SPARQL query.
- Semantic search in the semantic nomenclature: Using the selected primary ontology as reference, the user can select some relevant concepts he is interested in. Also, he can select secondary ontologies that have an interest in his search objective. Those selection values are used to generate the SPARQL query and retrieve results.
- Browsing in the query results: After the retrieval process, the system provide a list of results. Those results are presented as a list of triples that defines the relevant resources. The user has several interaction possibilities with each result resource. He can get the full description of the selected resource, or get the existing alignments. Finally, he can get the some related resources on the web (implemented as an example using Drugbank).
- Get the description of a result resource: The full description can be retrieved through a SPARQL query, and displayed in a specific grid. This grid presents to the user all the resource properties and the associated values.
- Map the retrieved resources with the alignment suggestions: From a result resource the user can get all the mapping suggestions provided by the Alignment Server. Each mapping suggestion can be confirmed by the user, and update in the ontology network.
- Add an issue in the Cicero wiki about the ontology evolution: When a mapping is created by the user, he can describe his action by adding an issue in the Cicero wiki. He has also the possibility to get the different issues related to a resource.

According the evaluation to be done by the users (pharma domain experts), improvements will be done in the next weeks until the very end of the project.

4 Conclusion

This document describes the Semantic Nomenclature second prototype developed within WP8. This second prototype tries to solve some of the lacks detected in the first version of the prototype and covers the main problems detected in the pharmaceutical scenario of the case study. For this purpose, we refreshed the goals and requirements of the case study and the current reviewed work done in the knowledge base with the last version of the Semantic Nomenclature ontology network. The document is accompanied with the software developed for the prototype within WP8.

From the knowledge base perspective, the current prototype provides several networked ontologies on the pharmaceutical domain. The knowledge base delivered on the first prototype was still using a set of non-connected ontologies, while the current version uses a more enriched and real network of pharmaceutical ontologies, covering a range of functionalities more focused on the case study requirements. Some of the ontologies have been designed using the NeOn Methodology while others have been added to the network from existing resources. The knowledge base is populated with original data available from the main pharmaceutical product database (Digitalis) from the Spanish Government and from the public information provided by the GSCOP in its database (BOTPlus). The conceptual structure of the KB is based on the ontologies reengineered from the RTMS databases and the main application ontology is developed following the requirements and recommendations detailed by domain experts and other domain approaches.

In terms of architecture, the evolution of the Semantic Nomenclature prototype from the previous version is significant. The first prototype was delivered as a J2EE Web application using Watson as the backbone while the current prototype is Ajax-based (using the GWT framework) and uses more NeOn services in the business and back-end infrastructure. Moreover, the new architecture is aligned with the WP6 recommendation for web applications and allows us to interact with different NeOn services in order to achieve the goals of the prototype and an easy maintenance of the application.

From the technological perspective, the deliverable describes the GUI, the usage of the GWT technology and how different RPC services have been implemented, one per each NeOn component used in the prototype. We described the role and usage of NeOn components: i) Cupboard to explore ontologies, query them, exchange them and exploit them through the Nomenclature ontology space; ii) the Alignment Server to retrieve the alignments and mappings between the Nomenclature ontologies; and iii) Cicero to create new issues when a new mapping is suggested. We also stressed the value of combining the Nomenclature ontology entities with Linked of Drug Data, that allow pharma end-users retrieve more information from Linked Data and create new candidate mappings that enrich the ontology network. Also, and overview of the client GUI taking into account the set of widgets to satisfy the user requirements is provided.

The main benefit from this approach is to show how the NeOn take about networking ontologies can fit in the pharma domain as a vehicle to link and aggregate content from previously disconnected resources. The prototype is leveraged on top of a a series of networked ontologies that (re)engineered using the NeOn Toolkit and some of its plugings using the NeOn technology.

As a case study, the Semantic Nomenclature depends on the availability of the different NeOn components and methodology. The late availability of some of the services used in our architecture impacted in the development of the Web application. In the last months of the project, the Semantic Nomenclature prototype will be evaluated in order to provide a last assessment of the software and ontology network. For these two reasons, the Semantic

Nomenclature prototype will be constantly evolving to a final release at the end of the project.

5 References

[1] Pariente T, Herrero G. Prototype v1 of the NeOn-enabled Semantic Nomenclature system using networked technologies. NeOn Deliverable D8.5.1, 2008

[2] Waterfeld W, Erdmann M, Schweitzer T, Haase P. Specification of NeOn architecture and API v2. NeOn Deliverable D6.9.1, 2008

[3] Gómez-Pérez J.M, Pariente T, Buil-Aranda C, Herrero G. Software architecture for the NeOn pharmaceutical case studies. NeOn Deliverable D8.2.1, 2007

[4] Gómez-Pérez J.M, Pariente T, Daviaud C, Herrero G. Analysis of the pharma domain and requirements. NeOn Deliverable D8.1.1, 2006

[5] Herrero G., Pariente T. Revision of Ontologies for Semantic Nomenclature: pharmaceutical networked ontologies. NeOn Deliverable D8.3.2, 2008

[6] d'Aquin M. Euzenat J., Le Duc C. Lewen H. K-CAP 09 Sharing and Reusing Aligned Ontologies with Cupboard , K-CAP09 Conference, 2009

[7] Pariente T. Herrero G. Prototype v1 of the Neon-enabled Semantic Nomenclature system using networked ontologies. NeOn Deliverable D8.5.1, 2007

[8] GWT documentation http://code.google.com/webtoolkit/tutorials/1.6/RPC.html , November 2009

[9] LinkedLifeData, data integration platform for biomedical domain. http://www.linkedlifedata.com/ , November 2009

6 Appendixes

6.1 Technical implementation of the server layer

The Semantic Nomenclature first prototype was as a J2EE client/server web application that followed the typical MVC pattern for the interaction between the UI and the business logic. Apache Struts was chosen as a free open-source framework for helping in the creation and development of Java web applications based on the MVC pattern. In this first prototype, were developed different servlets and business classes in the front-end server side to provide the business and interaction with NeOn technologies as Watson which were the backend of the web application.

As was introduced in the previous sections, for this second prototype we move from the Struts 2 web application to a Rich Internet Application based on the GWT technology. Google Web Toolkit (GWT) allows create AJAX web application coded in Java language that after will be cross-compiled by GWT engine in optimized Javascript code that works in major browsers. During development, we can iterate quickly in the same "edit - refresh - view" used in JavaScript, with the added benefit of being able to debug and step through your Java code line by line.

One of the features of GWT is the use of simple RPC for the communication with the server. GWT supports an open-ended set of transfer protocols such as JSON and XML, but GWT RPC makes all-Java communications particularly easy and efficient. Similarly to traditional Java RMI, we only need to create an interface that specifies remote methods we want be able to call. When a remote method is called from the browser, GWT RPC automatically serializes the arguments, invokes the proper method on the server, and then deserializes the return value in the client code. GWT RPC makes it easy for the client and server to pass Java objects back and forth over HTTP.

In order to better understand GWT-RPC, the following picture will give an overview.

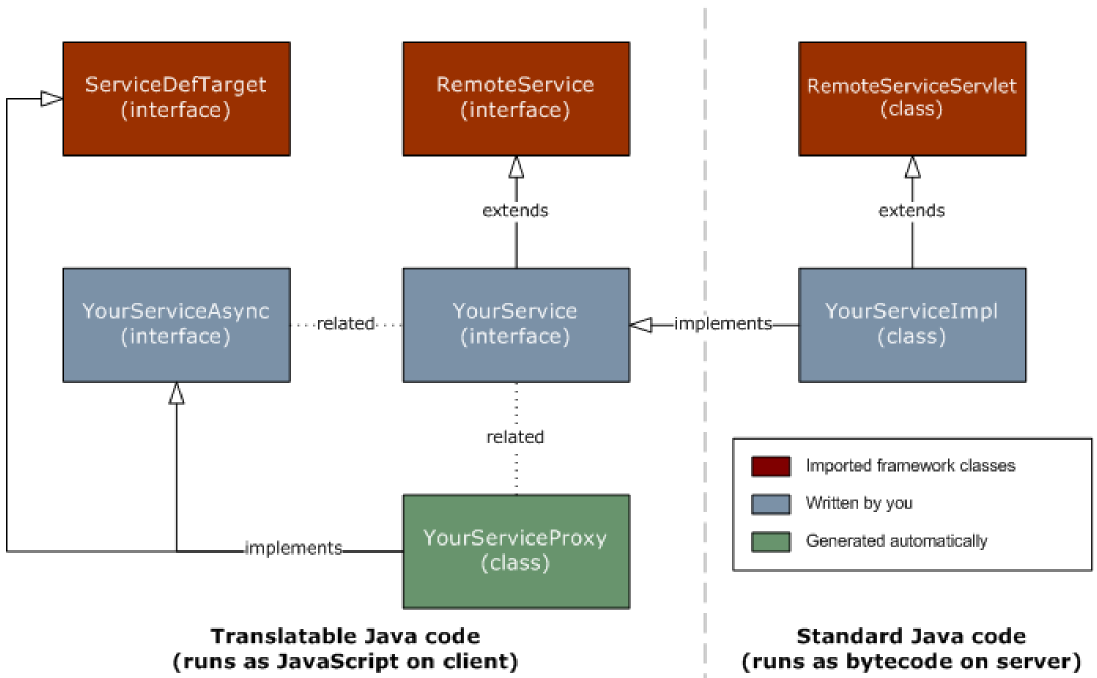


Figure 6: GWT – RPC [8]

Each service has a small family of helper interfaces and classes. In the java code, in the client packages should be defined the interfaces of the services that we want in our application. This interfaces are Service (extends `RemoteService`) which specifies a list of our RPC methods & *ServiceAsync* interface based on the Service interface which is the interface called from the client-side code. The nature of asynchronous method calls requires the caller to pass in a *callback* object that can be notified when an asynchronous call completes.

Finally, the services should be implemented in order to respond to client requests. The *ServiceImpl* classes are developed in the server side code of the GWT project. The Service implementation is based on the *servlet* architecture that extends `RemoteServiceServlet` and must implement the associated service interface.

The server front-end code (UI classes & RPC Service Interface) are implemented under the `eu.atosresearch.client` package. More specifically, the service interfaces and the associated asynchronus interfaces are allocated in `eu.atosresearch.client.service`. We have defined three different service interfaces, one per neon service. In these interfaces are detailed the different RPC methods allowed in the UI to interact with the server.

The service implementation or business components are located under the `eu.atosresearch.server.service.neon` package in the server side. As is described before, in order to minimize the complexity of the development, we distinguish between three different services implementation: one for the interaction with Cupboard, other for interaction with the Alignment repository and Alignment server, and finally other for the Cicero web service.

The different methods included in the different services cover the different requirements detected in the case study: search and query against a network of ontologies. Depending the RPC method, the service receives different parameters needed for query or search against the ontologies through the API / WS interface provided by the different services in the Data tier (Cupboard, Sesame, Alignment Server, Cicero).

6.2 LODD module

The current version of the prototype allows searching information related with the Nomenclature instances in Linked Data, and more specifically with Linking Open Drug Data (LODD). Among the current LODD datasets up to today we have selected Drugbank as a proof of concept. The Drugbank dataset has been integrated into the SPARQL endpoint of the LinkedLifeData[23] knowledge base at http://www.linkedlifedata.com/query. LinkedLifeData is a platform for semantic data integration trough RDF warehousing and efficient reasoning that helps to resolve conflicts in the data [7]. For the purpose of the prototype, we developed in the server implementation a module that connects with the SPARQL endpoint provided by LinkedLifeData.

The module developed in the server side of the prototype allows the user to search in Linked Data other resources related with an element of the ontology network. Moreover, the module allows to the end-user the functionality to create new candidate alignments to be added in the alignment repository between the Semantic Nomenclature ontology entities and Linked Data entities.

The code associated to Linked Data feature is implemented in the packages
`eu.atosresearch.server.json, eu.atosresearch.server.sesame and`
`eu.atosresearch.server.sparql`

[23] http://www.linkedlifedata.com/

When a new candidate mapping is created, a new issue is created in the Semantic Nomenclature Cicero space. For this purpose, from the server side is called the Cicero WS, and a login and create issue methods are invoked. This candidate mapping are visualized by the Semantic Nomenclature Ontology administrator which evaluate to include them in the local mapping repository as a formal mapping.

Possible future extensions could be based on:

- accessing other LODD datasets, such as Dailymed, linicalTrials.gov / LinkedCT, Diseasome, SIDER or Traditional Chinese Medicine;
- expose some parts of the semantic nomenclature network as a Linked Data node.

6.3 How to use Semantic Nomenclature Prototype

This appendix describes basic steps for using the second prototype of the Semantic Nomenclature prototype. This quick guide is intended for end-users in order to make easier their first interaction with the prototype and making it possible to evaluate this second prototype in terms of functionality and usability.

6.3.1 Launching the application

The second version of the Semantic Nomenclature has been released as a Web application based on GWT technology. To start the application you should use a web browser, such as Internet Explorer or Mozilla Firefox. The application has been tested with the latest versions of these Web browsers due that GWT works in the major browsers.

Currently the application is hosted in a server located in ATOS premises. In this guide we therefore provide the current URL: http://212.170.156.131:10000/semantic-nomenclature/

When the web page is loaded, the main desktop page is opened:

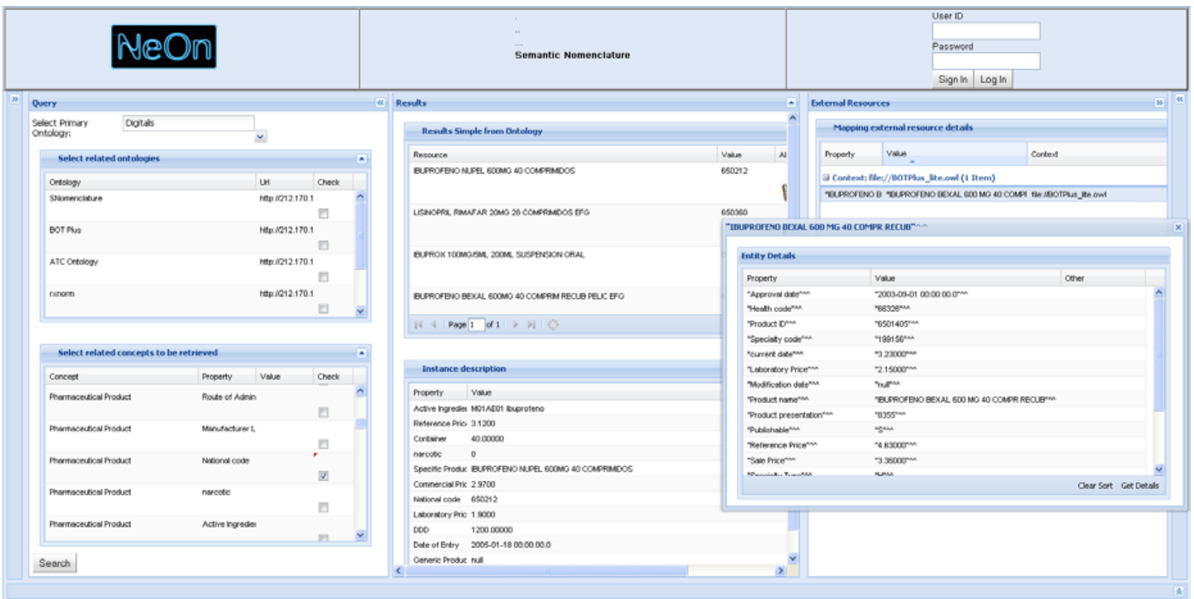


Figure 7: Semantic Nomenclature web default perspective

In the next subsections of the annex are depicted the main or most general use cases provided in the prototype.

6.3.2 Search instances against a single ontology

ACTORS

- Pharmacists, pharma experts

DESCRIPTION

The end-user executes a query against a single ontology of the Nomenclature Knowledge base (Nomenclature ontology network). The query is built based on the ontology selected and the query is composed of the selected concepts and properties.

PRECONDITIONS

- Concept and properties are checked
  - o There is a data source selected within the Query widget (SNomenclature, Digitalis, BOTPlus, ATC...).
  - o All fields belonging to the specific query are filled within Select related concepts Widget.
- The ontologies are available at the server layer and loaded in the Nomenclature Ontology Space accesible via the Cupboard WS.

TRIGGERING EVENT(s)

- Select the data source in the combo-box at the Query widget.

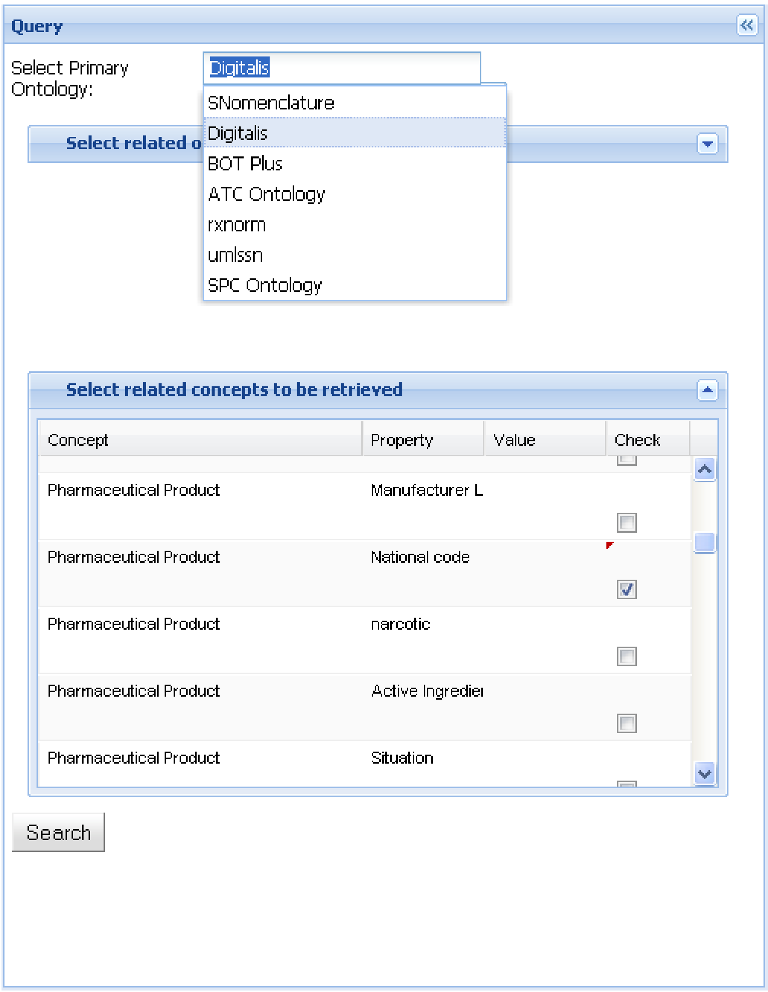


Figure 8: Query Widget

- User clicks the “Search” button at the bottom of the Query Widget. The generated SPARQL query is sent to the server side, which collects the attributes selected by the Check column and with filled restriction.

POST CONDITIONS

- Results Simple widget displays the query results.

FLOW OF EVENTS

a. BASIC FLOW

- User selects the data source in the combo of the taxonomy widget (i.e. Digitalis).

- User goes to the Select Related Concepts widget and selects the desired concept and properties to query.

    - For instance, user selects to ask by “*Pharmaceutical Product*” concept and “National Code” property.

    - User set the input keyword for the local name (i.e. “*650630*”, “” search all entities).

- User click on the Search Button.
- System displays results in the Results Simple Ontology Widget.

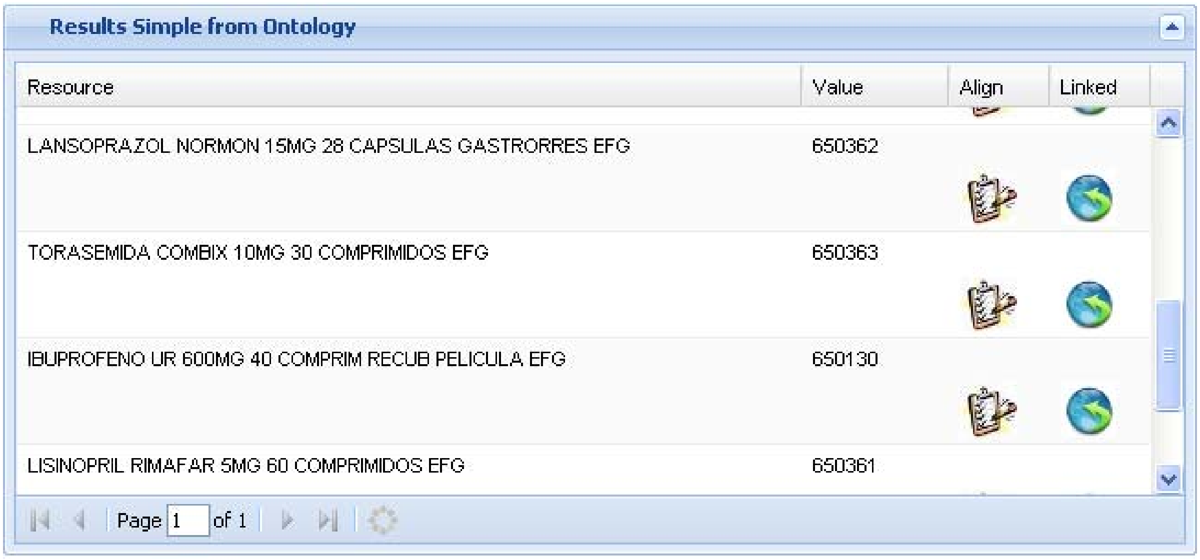


Figure 9: Results Simple Widget

- User can sort results in the Query Results Widget by clicking Resource column title or can hide/show columns by right-clicking in any column title.
- User can click on any cell showed in each result and access/open the result resource details provided by the system or search aligned results or in Linked Data.

b. ALTERNATIVE FLOW: no results found

- System returns an empty Results Simple Widget grid.

RELATED USE CASES

- Any other Simple Search Instance Query.

NOTES / ISSUES

For composed queries (various restrictions selected) it is compulsory to fill all fields of involved queries. The result will be the intersection of both results.

6.3.3 Get Entity details

ACTORS

- Pharmacists, pharma experts.

DESCRIPTION

- User obtains details of a concrete resource from the ontology network

PRECONDITIONS

- The ontologies are available at the server layer. These ontologies are loaded in the Nomenclature Ontology Space and accessible via the Cupboard WS.

TRIGGERING EVENT(s)

- User clicks any cell in any result row of the Results Simple widget.

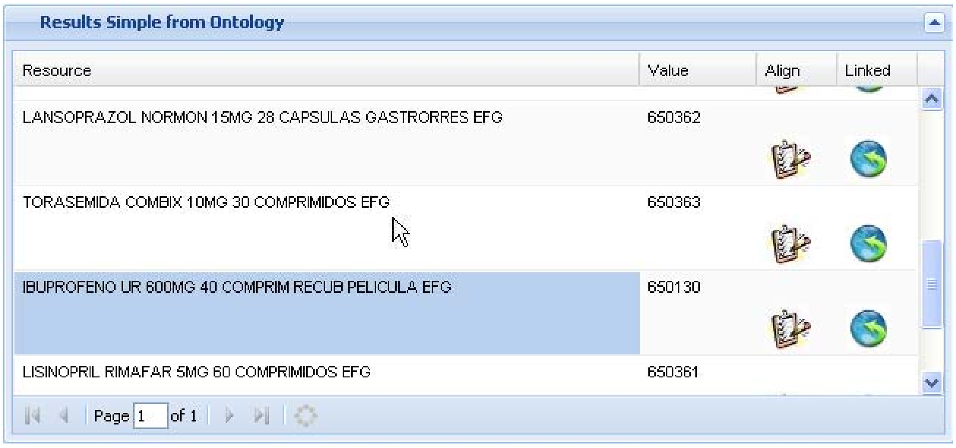


Figure 10: Select Nomenclature entity

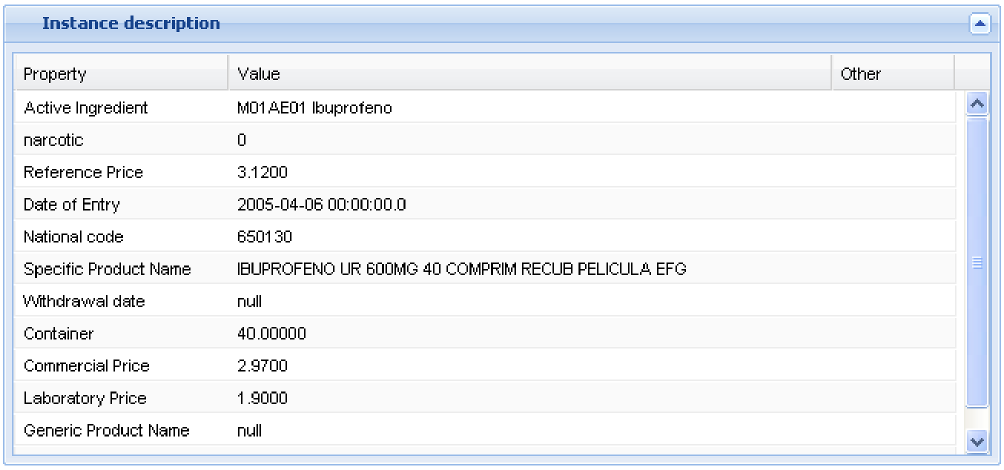
POST CONDITIONS

- System displays the entity details in the Instance description widget.

FLOW OF EVENTS

a. BASIC FLOW

- User goes to the Result Simple Widget at the top-right of the page.
- User clicks on the “IBUPROFENO UR 600MG 40 COMPRIM RECUB PELICULA EFG” cell in the resource row
- System displays results in the Instance description Widget.


Instance description

Property	Value	Other
Active Ingredient	M01AE01 Ibuprofeno	
narcotic	0	
Reference Price	3.1200	
Date of Entry	2005-04-06 00:00:00.0	
National code	650130	
Specific Product Name	IBUPROFENO UR 600MG 40 COMPRIM RECUB PELICULA EFG	
Withdrawal date	null	
Container	40.00000	
Commercial Price	2.9700	
Laboratory Price	1.9000	
Generic Product Name	null	

Figure 11: Instance description widget

b. ALTERNATIVE FLOW: no results found

- System returns an empty grid result.

RELATED USE CASES

- Any other Instance details Search.

6.3.4 Linked Data Interaction

ACTORS

- Pharma experts, pharmacists, ontology experts

DESCRIPTION

User obtains more added information from a ontology entity, searching in the Linked of Drug Data aligned resource. The popup shows the end-user which dataset are available (Drugbank, Dailymed) and opens a new web page with the resources retrieved from Linked Data. Moreover, allow the user to create candidate mappings between the Nomenclature entity and the Linked Data entity.

PRECONDITIONS

- The selected resource has significant properties and values to query Linked Data.
- The external Linked Data SPARQL endpoint is available. .

TRIGGERING EVENT(s)

- The user can clicks in the Linked Data icon in any row of the Result Simple widget.

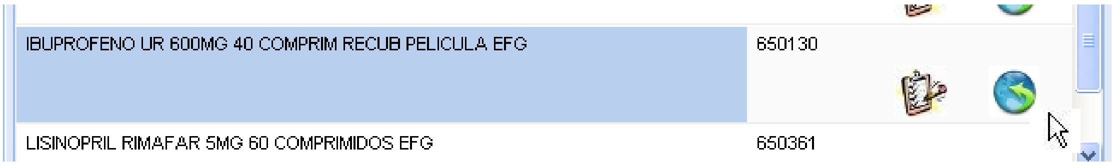


Figure 12: Click on the Linked Data icon

- The user clicks on Search button in the Linked Data Popup

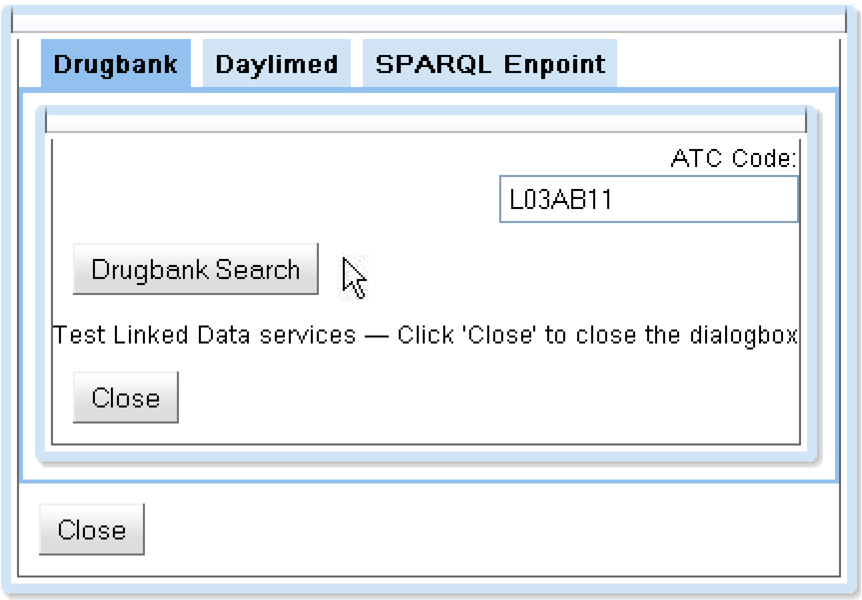


Figure 13: Linked Data feature Popup

- The user clicks on the add alignment button and then in the create alignment

Figure 14: Create a new candidate alignment

POST CONDITIONS

- System displays the Linked Data resources associated to the selected Nomenclature resource in a new web page
- System creates a new candidate mapping in the local alignment repository
- System creates a new issue in the Semantic Nomenclature project in Cicero wiki

FLOW OF EVENTS

a. BASIC FLOW

- User clicks in the Linked Data icon of any result in the Result Simple Widget.
- The Linked Data Popup is showed with a significant Nomenclature resource property related with LODD. The user clicks on the Search Button.
- A new web page is opened in the browser. This web page loads the Simile template structure and shows the Linked Data resources found in LODD.

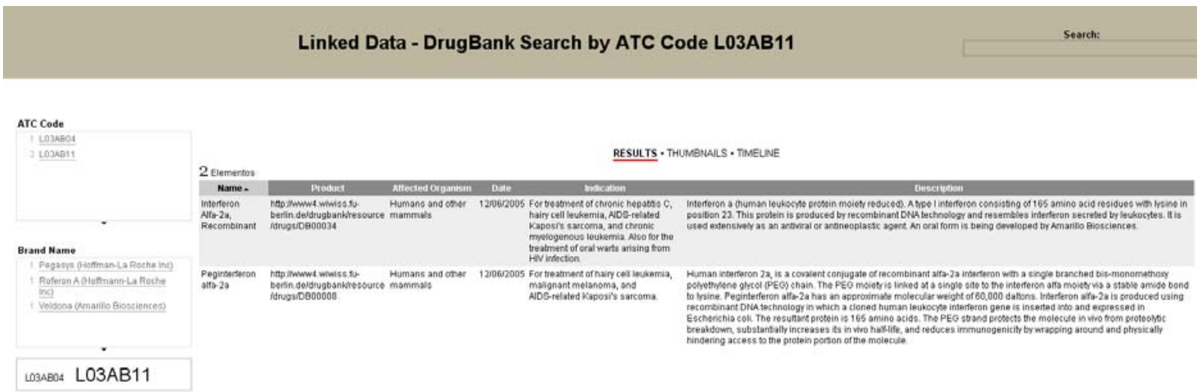


Figure 15: Linked Data Nomenclature web page

- The user interacts with the web page that displays LODD resources. In the thumbnails perspective the user clicks on the “Perform an alignment” link.

Figure 16: Perform an alignment in S. Nomenclature - Linked Data feature

- A new popup is showed with a form. This form includes the Nomenclature resource URI and Linked Data resource URI with two combobox. User select which type of mapping will be created (owl:equivalentClass for concepts; owl:sameAs for instances). The other combo allows the user to select the context of the new mapping (e.g. Digitalis – Drugbank, Digitalis-Dailymed, Botplus-Drugbank...)

- User clicks the "Create an alignment" button and a new alignment is created in the local alignment repository of the system. Also a new issue is created in the Nomenclature project Cicero wiki.

b. ALTERNATIVE FLOW: no results found

- The Linked Data SPARQL endpoint is not available

- The Nomenclature resource has not significant properties related with Linked Drug of Data.

RELATED USE CASES

- Any other Aligned or Mapped resource search.

6.3.5 Retrieve Aligned resources results

ACTORS

- Pharmacist, pharma experts

DESCRIPTION

User obtains more added information of a concrete Nomenclature resource, searching in the aligned resources of the different ontologies of the ontology network. The aligned widget shows aligned results from the Nomenclature network and the user can obtain more details from other datarsets for its selected resource.

PRECONDITIONS

- The selected resource has significant alignments in the Alignment repository.

- The Alignment repository is available

TRIGGERING EVENT(s)

- The user can clicks on the Aligned icon in any row of the Result Simple widget (e.g. IBUPROFEN BEXAL 600MG 40 COMPRIM RECUB). The Aligned results widget is filled with aligned resources found in the alignment repository.

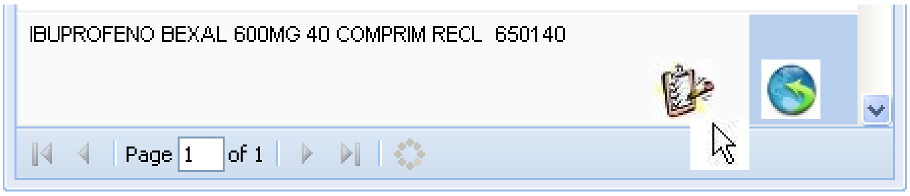


Figure 17: Click on Aligned icon

- The aligned results are grouped depending of the context of the mapping (Botplus ontology, Digitalis-Drugbank...). User clicks on any cell in the results showed in the Aligned results widget. A new widget is opened in the Semantic Nomenclature prototype desktop. In this case, in the “IBUPROFEN BEXAL 600MG 40...” cell.

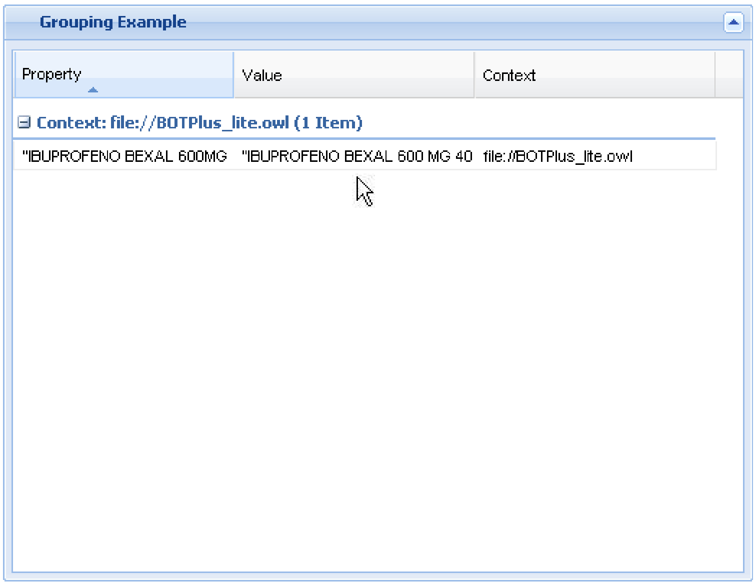


Figure 18: Aligned Results Widget

- User clicks on “Get details” button in Aligned details widget

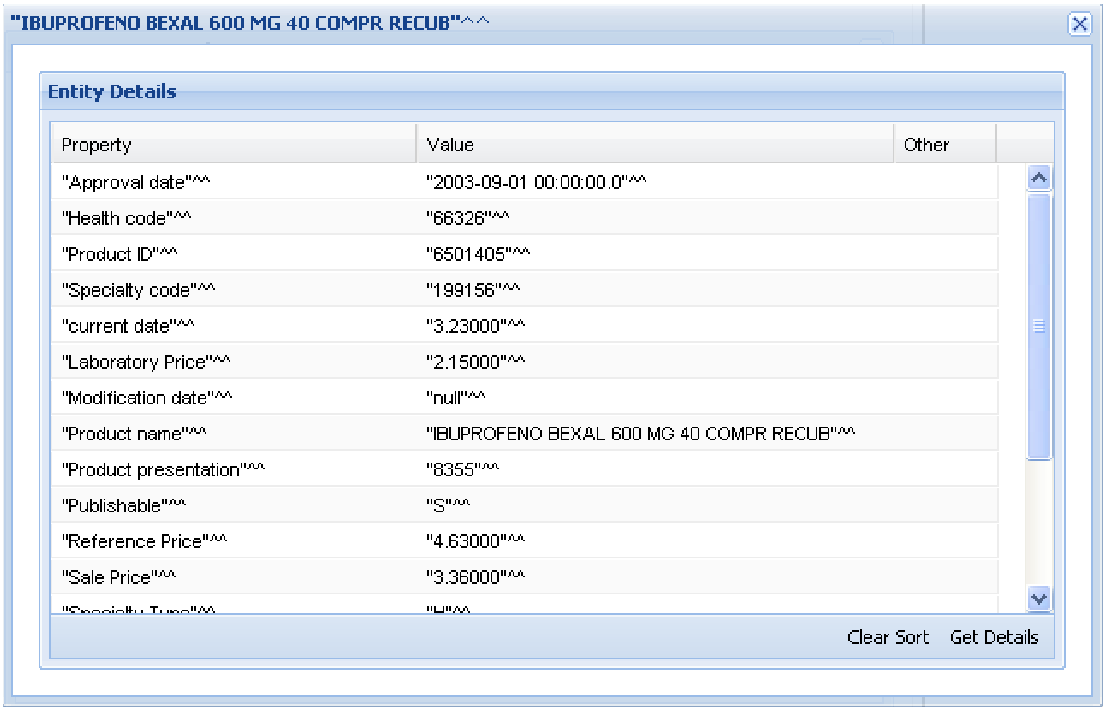


Figure 19: Aligned resource details Widget

POST CONDITIONS

- System displays the aligned resources associated to the selected Nomenclature resource in Aligned results widget
- System displays the aligned resource details in the Aligned details Widget

b. ALTERNATIVE FLOW: no results found

- The Alignment Repository is not available
- The Nomenclature resource has not significant alignments in the repository.

RELATED USE CASES

- Any other Aligned or Mapped resource search.