LarKC

The Large Knowledge Collider:
a platform for large scale integrated reasoning and Web-search

FP7 – 215535

D7a.3.2 Prototype v2

Coordinator: Vassil Momtchev
With contributions from: Deyan Peychev, Konstantin Pentchev, Todor Primov, Rostislav Hristov, Bo Andersson
Quality Assessor: Mark Greenwood
Quality Controller: Vassil Momtchev

Document Identifier:	LarKC/2010/D7a.3.2 /v1.0
Class Deliverable:	LarKC EU-IST-2008-215535
Version:	1.0
Date:	22.12.2010
State:	final
Distribution:	public

EXECUTIVE SUMMARY

This deliverable describes the second prototype version of Linked Life Data (LLD) and the application of the LarKC platform as new exploration methods for challenges such as drug discovery, genetic epidemiology of cancer and other diseases, and a carcinogenesis research. LLD aggregates more than 25 public domain databases and collects over 6 billion facts, describing all types of biomedical knowledge. The service offers a solid dataset for research and experiments of various reasoning and information extraction techniques, and at the same time applies a new type of knowledge representation methods and infrastructure to help better analyze the information relevant for the drug development process.

DOCUMENT INFORMATION

IST Project Number	FP7 - 215535	**Acronym**	LarKC
Full Title	The Large Knowledge Collider: a platform for large scale integrated reasoning and Web-search		
Project URL	http://www.larkc.eu/		
Document URL			
EU Project Officer	Stefano Bertolo		

Deliverable	**Number**	D7a.3.2	**Title**	Prototype v2
Work Package	**Number**	WP7a	**Title**	Semantic Integration for Early Clinical Development

Date of Delivery	**Contractual**	M 33	**Actual**	
Status	version 1.0		final x	
Nature	prototype x report □ dissemination □			
Dissemination level	public x consortium □			

Authors (Partner)	Vassil Momtchev, Deyan Peychev, Konstantin Pentchev, Todor Primov, Rostislav Hristov (Ontotext), Bo Andersson (AstraZeneca)			
Responsible Author	**Name**	Vassil Momtchev	**E-mail**	vassil.momtchev@ontotext.com
	Partner	Ontotext	**Phone**	

Abstract (for dissemination)	This deliverable describes the second prototype version of Linked Life Data (LLD) and the application of the LarKC platform as new exploration methods for challenges such as drug discovery, genetic epidemiology of cancer and other diseases, and a carcinogenesis research. LLD aggregates more than 25 public domain databases and collects over 6 billion facts, describing all types of biomedical knowledge. The service offers a solid dataset for research and experiments of various reasoning and information extraction techniques, and at the same time applies a new type of knowledge representation methods and infrastructure to help better analyze the information relevant for the drug development process.
Keywords	LLD, linked data, RDF, warehouse, drug development

Version Log			
Issue Date	**Issue Date**	**Issue Date**	**Issue Date**
20/12/2010	20/12/2010	20/12/2010	20/12/2010
22/12/2010	22/12/2010	22/12/2010	22/12/2010

PROJECT CONSORTIUM INFORMATION

Participant's name	Partner	Contact
Semantic Technology Institute Innsbruck, Universitaet Innsbruck		Dieter Fensel, Semantic Technology Institute (STI), Universitaet Innsbruck, Innsbruck, Austria, E-mail: dieter.fensel@sti-innsbruck.at
AstraZeneca AB		Bosse Andersson, AstraZeneca Lund, Sweden Email: bo.h.andersson@astrazeneca.com
CEFRIEL - SOCIETA CONSORTILE A RESPONSABILITA LIMITATA		Emanuele Della Valle, CEFRIEL - SOCIETA CONSORTILE A RESPONSABILITA LIMITATA, Milano, Italy, Email: emanuele.dellavalle@cefriel.it
CYCORP, RAZISKOVANJE IN EKSPERIMENTALNI RAZVOJ D.O.O.		Michael Witbrock, CYCORP, RAZISKOVANJE IN EKSPERIMENTALNI RAZVOJ D.O.O., Ljubljana, Slovenia, Email: witbrock@cyc.com
Höchstleistungsrechenzentrum, Universitaet Stuttgart		Georgina Gallizo, Höchstleistungsrechenzentrum, Universitaet Stuttgart, Stuttgart, Germany, Email: gallizo@hlrs.de
MAX-PLANCK GESELLSCHAFT ZUR FOERDERUNG DER WISSENSCHAFTEN E.V.		Lael Schooler, Max-Planck-Institut für Bildungsforschung Berlin, Germany Email: schooler@mpib-berlin.mpg.de
Ontotext AD		Atanas Kiryakov, Ontotext Lab, Sofia, Bulgaria Email: naso@ontotext.com
SALTLUX INC.		Kono Kim, SALTLUX INC, Seoul, Korea, Email: kono@saltlux.com
SIEMENS AKTIENGESELLSCHAFT		Volker Tresp, SIEMENS AKTIENGESELLSCHAFT, Muenchen, Germany, E-mail: volker.tresp@siemens.com
THE UNIVERSITY OF SHEFFIELD		Hamish Cunningham, THE UNIVERSITY OF SHEFFIELD Sheffield, UK, Email: h.cunningham@dcs.shef.ac.uk

VRIJE UNIVERSITEIT AMSTERDAM		Frank van Harmelen, VRIJE UNIVERSITEIT AMSTERDAM, Amsterdam, Netherlands, Email: Frank.van.Harmelen@cs.vu.nl
THE INTERNATIONAL WIC INSTITUTE, BEIJING UNIVERSITY OF TECHNOLOGY		Ning Zhong, THE INTERNATIONAL WIC INSTITUTE, Mabeshi, Japan, Email: zhong@maebashi-it.ac.jp
INTERNATIONAL AGENCY FOR RESEARCH ON CANCER		Paul Brennan, INTERNATIONAL AGENCY FOR RESEARCH ON CANCER, Lyon, France, Email: brennan@iarc.fr
INFORMATION RETRIEVAL FACILITY		John Tait, INFORMATION RETRIEVAL FACILITY Vienna, Austria Email : john.tait@ir-facility.org

TABLE OF CONTENTS

List of Figures

List of Tables

List of Acronyms

Acronym	Description
API	Application Programming Interface
ECD	Early Clinical Development
ETL	Extraction Transformation Loading (a typical data warehouse process)
GO	Gene Ontology
KB	Knowledge Base
LarKC	Large Knowledge Collider
LLD	Linked Life Data
OBO	Open Biomedical Ontologies
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
PIKB	Pathway and Interaction Knowledge Base
RDF	Resource Descriptor Framework
RDFS	RDF Schema
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language
UMLS	Unified Medical Language System
KOS	Knowledge Organisation System
SKOS	Simple Knowledge Organisation System

1. Introduction

This deliverable is an update of the first use case prototype described in [2] and reports the latest Linked Life Data (LLD) service upgrades. LLD is the main deliverable of the WP7a "Semantic Integration for Early Clinical Development". In this work package the LarKC platform technology is used to support the development of new exploration methods for challenges such as: drug discovery, genetic epidemiology of cancer and other diseases, and a carcinogenesis research. Currently, the LLD service is publicly operational at http://linkedlifedata.com. A production service is also deployed for internal research on AstraZeneca intranet.

The efficient data integration and interoperability is the biggest challenge faced by the existing technology. The ever increasing amounts of information generated by the distributed and decentralized data providers put a constant pressure on the data driven research of early clinical development. LLD aggregates 26 public domain databases and contains over 6 billion facts, describing all types of biomedical knowledge. The service offers a solid dataset for research and experiments of various reasoning and information extraction techniques, and explores a new type of knowledge representation methods and infrastructure to help better analyze the information relevant to the drug development process.

In 2010 the public version of LLD served more than 17,500 unique users and provided a single integrated SPARQL endpoint for the most popular public biomedical databases. Its users generated more than 400,000 hits and 5GB of transferred data, using the multiple offered interfaces.

The second prototype version is focussed on delivering a very scalable and highly efficient infrastructure for RDF data warehousing. Chapter 2 presents a conceptual data integration methodology based on the RDF model and describes the latest updates and growth in the collected databases. Chapter 3 presents a data integration application that extends a standard Extra, Transform, Load (ETL) framework with RDF and LarKC data types. Finally, the document describes the next steps in WP7a use case and suggests possible directions for further improvement.

2. Dataset updates

This chapter presents the latest update on the datasets and their schemata. In the latest public LLD 0.6.1 release the number of the data sources has increased to 26, and the total number of statements has passed the 6 billion barrier - 6,217,184,106. All databases versions have been updated since the first prototype release, which lead to a 50% increase in the number of RDF statements.

2.1. Dataset statistics and growth

Table 1 lists all current data sources, their loading date and the number of triples. The latest release table and dump files can be accessed from the LLD website [1].

Data source	Load date	Number of statements	Comment
BioGRID	31.07.10	21,017,987	Molecular interactions
CellMap	24.09.10	148,350	Molecular interactions
ChEBI	04.10.10	322,960	Chemical compounds
DailyMed	13.10.10	192,578	Drug related information
DBPedia	20.07.10	494,861,175	Various information
Disease Ontology	21.10.10	144,552	Diseases
Diseasome	14.10.10	73,834	Disease-gene network
DrugBank	14.10.10	517,694	Drugs
Freebase	10.04.10	395,958,356	Various information
HapMap	17.12.10	22,462,178	SNPs
HPRD	24.09.10	1,917,111	Molecular interactions
Human Phenotype Ontology	13.10.10	84,378	Phenotypes
HumanCYC	24.09.10	300,720	Molecular interactions
IMID	24.09.10	81,659	Molecular interactions
IntAct	24.09.10	16,229,745	Molecular interactions
LHGDN	08.03.10	316,020	Disease-gene network
LinkedCT	14.10.10	7,110,670	Clinical trials
MINT	24.09.10	20,277,714	Molecular interactions
NCBI Entrez-Gene	18.10.10	140,024,275	Genes
NCI Nature	24.09.10	614,086	Molecular interactions
PubMed	18.10.10	1,360,059,619	Citations
Reactome	24.09.10	698,567	Molecular interactions
Semantic annotation	18.10.10	608,983,313	Links between the data sources
SIDER	13.10.10	106,805	Drug side effects
Symptom Ontology	15.10.10	4,163	Symptoms
UMLS	25.11.09	110,568,086	Meta-thesaurus
UniProt	04.09.10	1,833,277,354	Proteins

Table 1 Data sources statistics

We can see a dramatic increase in the RDF size in some of the actively maintained primary data sources like Uniprot (226% compared to the prototype presented in [2]). The RDF statement increase is due to the nearly double number of entries since 2009. Another contributing factor is the complication of the information schema that resulted in a slight increase in the average statements per record. On Figure 2 we can see the continuous trend of the doubling of number entries every two years. A similar trend is also observed in NCBI Entrez-Gene and Pubmed data sources, where the new entries and their meta-data improvements contributed to 40% and 69% RDF statements increase respectively. The size of UMLS data source has increased almost 10 times because of the extended relationship data model.

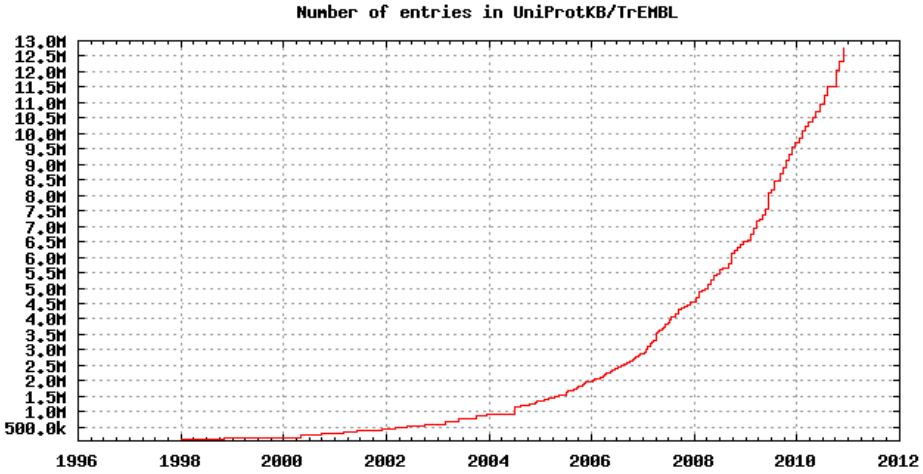

Figure 1 The number of entries in UniprotKB/TreMBL[1]

2.2. Instance alignment process

The LLD knowledge base generation process is based on few simple and effective conventions that are necessary to resolve the syntactic level and data format heterogeneity problems. The following 6 basic conventions ensure that all original database identifiers are easily retrievable and compliant with the linked data principles:

1. Preserve the original RDF structure if distributed by the owner
2. Use resolvable URIs for the data sources with no RDF distribution
3. Construct the generated URIs in the form of `lld:resource/db/type/id`
4. Identify the graph names with `lld:resource/db`
5. Name all generated predicate URIs `lld:resource/db/predicate`
6. Generate stable new URIs based on unique label that describes the resource (see Dataset provenance and updates)

In a nutshell, we encourage all content providers to maintain resolvable URIs and govern the policy for resources publishing, update and retraction. In the scenarios where the URIs are not resolvable, LLD will act as a mediator that facilitates the end users in rewriting the URIs. Once all data is loaded in a warehouse we ensure that the relevant resources are properly interlinked. The resource interlinking of the cross-data source guarantees that the redundant identifiers are associated with one of the three levels of relationship, reused from the SKOS schema [3]:

- http://www.w3.org/2004/02/skos/core#exactMatch – full resource equivalence that should be transitively propagated. It is used when there is full overlap in the identifiers and the type of resources (e.g. Uniprot and BioPAX protein sequence identifier).
- http://www.w3.org/2004/02/skos/core#closeMatch – resource equivalence limited only to the information retrieval needs. It is used when there is an overall similarity but different context of the sources (e.g. BioPAX protein sequence and an identifier of the encoding gene).
- http://www.w3.org/2004/02/skos/core#relatedMatch – limited resource equivalence that facilitates the composition of complex queries without involving string matching functions.

The semantics of the corresponding RDFS and OWL properties `rdfs:seeAlso` and `owl:sameAs` is often misinterpreted and non-consistently applied across the different RDF datasets. The `rdfs:seeAlso` property does not directly result in any inferred statements. It poses a challenge only to the query developer who should consider its scope. However, the `owl:sameAs` property

[1] http://www.ebi.ac.uk/uniprot/TrEMBLstats/

produces a massive number of implicit statements and conceptual inconsistencies, if not properly used. For instance, when two concept are merged in the data sources represented in SKOS, the statement of `owl:sameAs` is likely to result in multiple `skos:prefLabel` or `skos:inScheme` values. Another example from the LOD cloud is the assertion of an `owl:sameAs` statement between an organism specific gene and the abstract notion of a gene. Thus, a full equivalence between all gene information will be automatically generated. Hence, the LLD loading process filters all `owl:sameAs` statements and replaces them with the less engaging and safer `skos:exactMatch` predicate.

All instance-level equivalence in LLD is derived by one of the patterns presented in Figure 2, i.e. after the replacement of an existing `owl:sameAs` predicate with `skos:exactMatch`. The solid lines express the existing explicit relationships used to map the data. The dashed lines and the underlined text of the captions (e.g. used either as part of the URI or literals) designate the criteria for mapping the information. The specified mapping rules are not applicable in all cases, and are designed for specific datasets only.

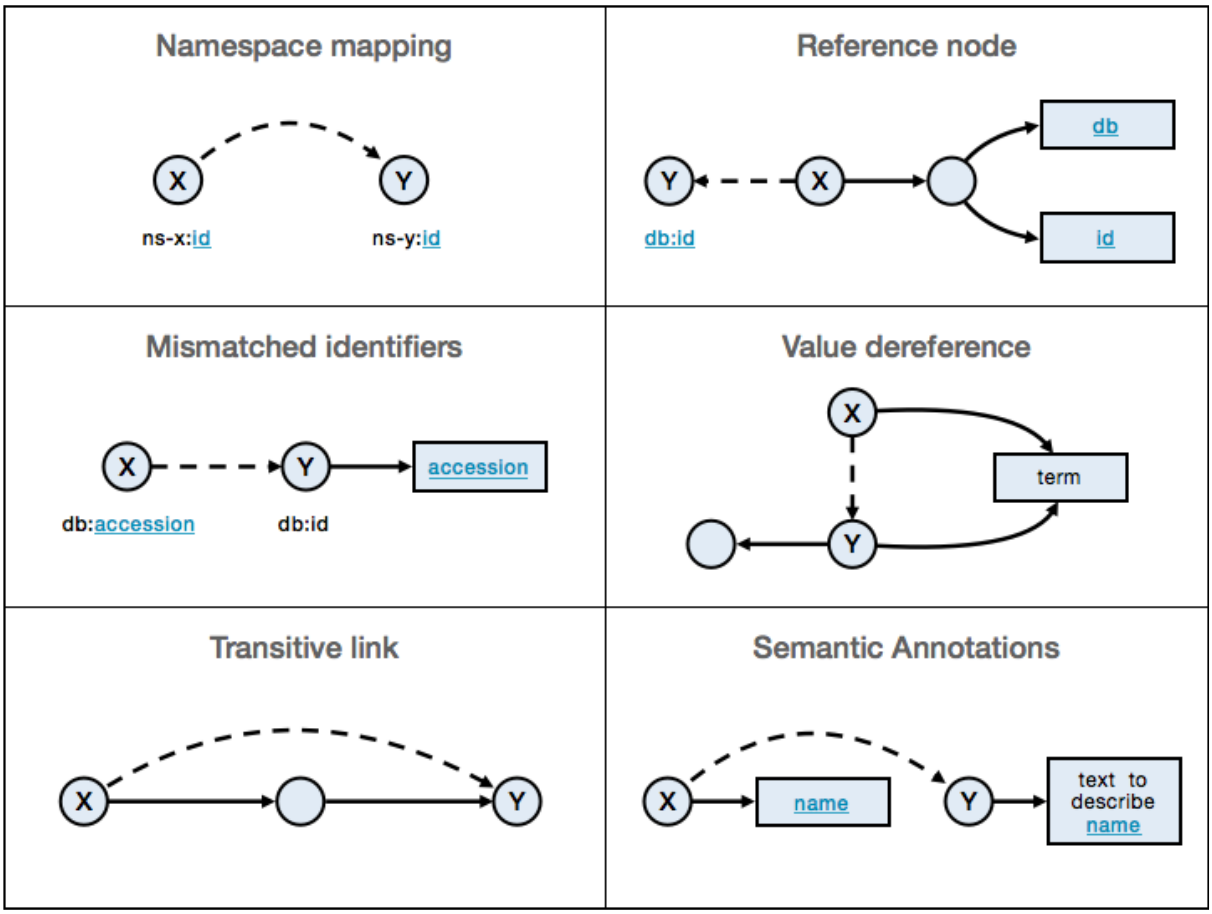

Figure 2 Patterns to align instance level identity over linked data

Table 2 represents all statistics for new explicitly added cross data source mappings. The mappings are now automatically generated as part of the extraction, transformation and loading (ETL) process described in Chapter 3. For latest information about the explicitly generated cross-data source mappings, please refer to [1] data sources page.

Source dataset	Destination dataset	Linked Data Mapping Rule	Number of connections	Semantic relationship
BioPax GO	UMLS concepts	Reference Node	301,655	skos:relatedMatch

	(GO)			
UMLS concepts (MeSH)	PubMed MeSH terms	Value Dereference	25,306	skos:exactMatch
Pubmed	UniProt citations	Namespace Mapping	871,999	skos:exactMatch
UMLS concepts (NCBI Taxonomy)	UniProt organisms	Value Dereference	412,297	skos:exactMatch
BioPax Proteins	UniProt	Reference Node	148,941	skos:exactMatch
DrugBank targets	UniProt	Namespace Mapping	1,384	skos:exactMatch
DrugBank targets (via HGNC references)	EntrezGene (via HGNC references)	Namespace Mapping	1,617	skos:relatedMatch
UniProt GO terms	UMLS concepts (GO)	Mismatched Identifiers	51,024	skos:exactMatch
BioPax Citations	Pubmed	Reference Node	1,219,394	skos:exactMatch
UMLS concepts (NCBI Taxonomy)	EntrezGene organisms	Mismatched Identifiers	5,303	skos:exactMatch
EntrezGene GO terms	UMLS concepts (GO)	Mismatched Identifiers	30,578	skos:exactMatch
BioPax Proteins	Entrez Gene	Reference Node	36,783	skos:closeMatch

Table 2 Cross data source mapping statistics

2.3. New visualization features

The returned RDF data is very difficult to be explored, even by expert users. A very common problem is the need to see nested information hidden behind an URI. Since LLD 0.5 release, the service has offered a new, more user-friendly views of the rendered information, depending on its type. The number of the customized (i.e. no RDF triples) views is growing quickly and it covers multiple information types such as: gene, proteins, terminology concept, drugs, article, etc.

Figure 3 presents the default view of an article and lists the authors, title, abstract, and its type.

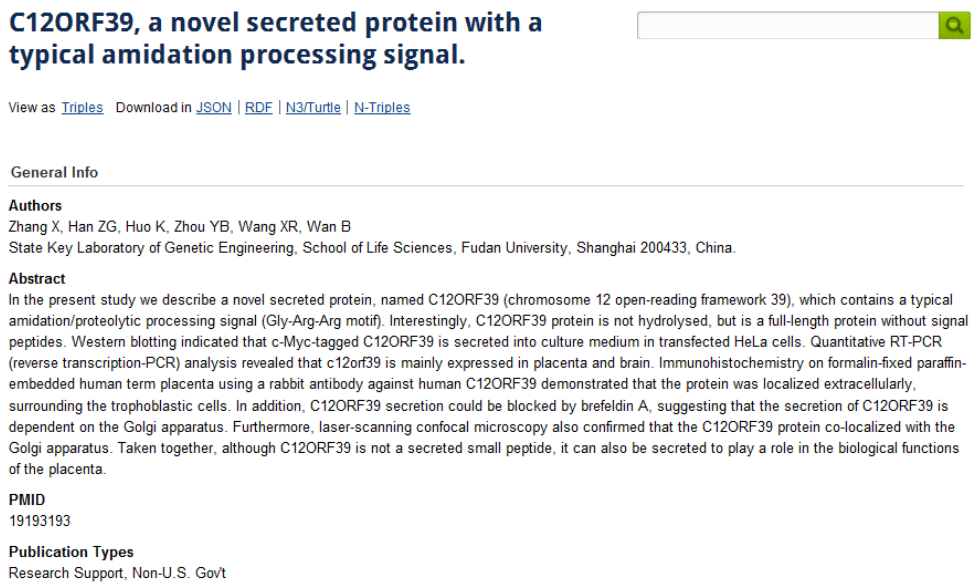

Figure 3 A screenshot of an article view

Another common problem is the visualization of a terminology concept mentioned by multiple coding systems. The screen shot in Figure 4 presents a user interface where one can see the concept names and their alternative names, mentioned by the different terminologies. It is also possible to explore all broader and narrower concepts. The inferred and the explicit links are displayed differently.

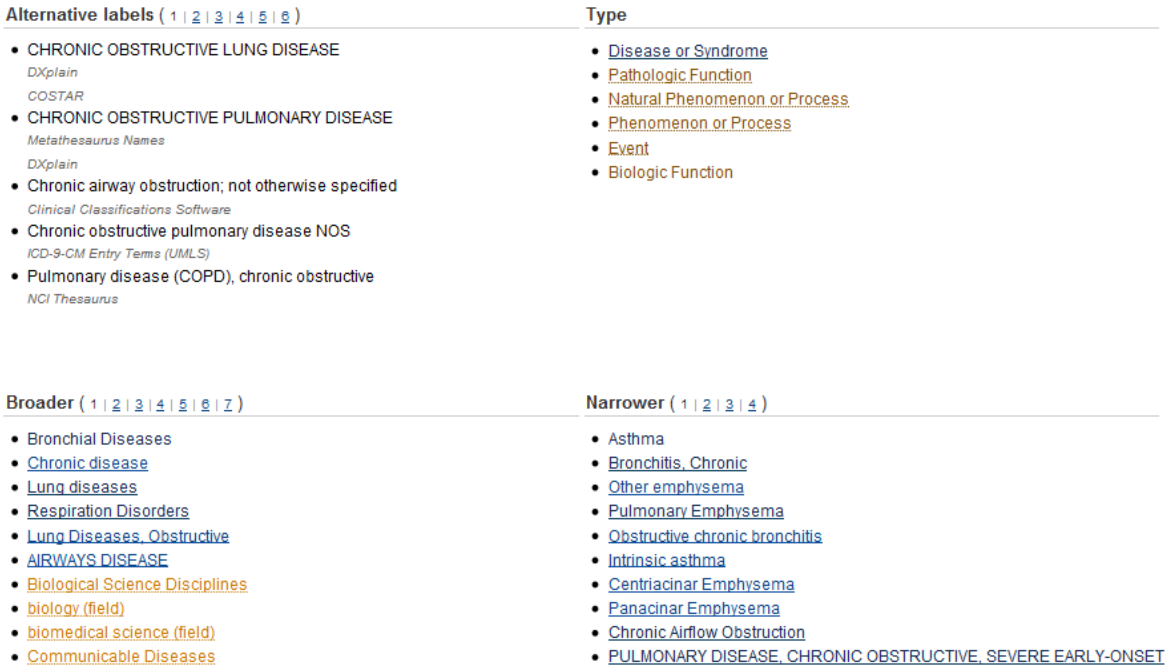

Figure 4 A screen shot of a terminology concept

The protein sequence view in Figure 5 shows the extracted relevant meta-data of all related resources such as keywords and interactions and how they are presented in an easy to navigate interface.

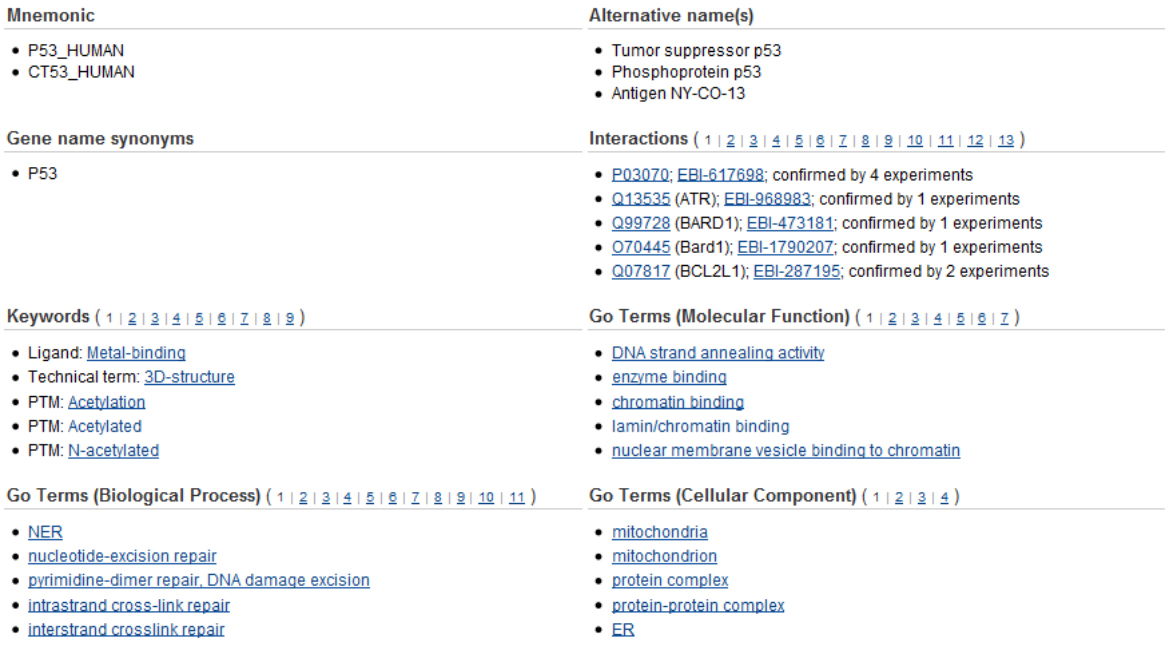

Figure 5 A screen shot of complex protein sequence meta-data

3. Data transformers code generation

The LarKC platform is designed to be a flexible plug-in architecture, [4]. The task of transforming various bits of information into RDF has a central position in the WP7a "Semantic Integration for Early Clinical Development". To enable the full potential of the LarKC plug-in infrastructure, composed by various selectors, deciders and reasoners, all relevant data first should be represented in RDF format. This chapter describes the software behind the Linked Life Data updates process and the way it is integrated into the LarKC platform.

The RDF and linked data technologies are considered as an excellent platform for data integration in the drug discovery and translational medicine, [5], [6], [7]. Every RDF integration methodology can be classified somewhere between the two classical data integration approaches – federation and warehousing. The data federation approach uses distributed and heterogeneous information systems without replicating the actual information. All schema level heterogeneity is resolved in query time by adapters, configured by various mapping languages. Data warehousing takes the opposite approach by replicating all information in a single centralized system, and resolves all information heterogeneity problems loading time. This process is known as extract, transform and load (ETL). In the context of RDF data model we have found many applications and languages, trying to address the problem of the data federation. Some good examples are the declarative schema level languages like D2RQ [8] and VOSSQL2RDF [10]. Recently, W3C recognized the great importance of reaching better data interoperability across different applications, and initiated a joint effort for the standardisation of common mapping language – R2RML, [9]. The federation approach seems an appealing and efficient integration option, but it is unable to address the major requirements of the LLD service: a) integration of high-number of different data sources and b) very quick query response. The warehousing approach makes a predictable query response time for all types of searches possible, including a very specific one like the unbound predicate. Another major advantage is the highly-efficient implementation of forward-chaining reasoners for the materialization (or simply indexing) of trivial facts.

3.1. TalenD Open Studio Introduction

Because there is no data warehousing ETL tools that support RDF, we decided to use the readily available components of a popular open-source ETL platform. TalenD Open Studio[2] is a leading open-source data integration and ETL development framework that includes an integrated development environment (IDE). The IDE also offers a GUI for designing data transformation workflows and copying data across different systems. The designed job within the IDE can be exported as Java code, by code generation and further wrapping in other libraries.

3.2. RDF Components Extension

For the needs of LLD update process we implemented 7 new components that allow the usage of RDF data:

- `tRDFParser` – parses an RDF file and streams a sequence of triples
  - Configuration parameters
    - RDF data format
    - Input RDF file
  - Input: none
  - Output: A stream of RDF triples
- `tRDFWriter` – reads a stream of triples and outputs RDF file

[2] http://www.talend.com/

    - Configuration parameters
        - RDF data format
        - Output data file
    - Input: A stream of RDF triples
    - Output: none
- `tOWLIMInput` – reads data from OWLIM/LarKC data layer instance and stream it
    - Configuration parameters
        - In process or remote RDF repository
        - RDF Storage path
        - Path to Sesame TTL configuration file
        - SPARQL query to be executed
        - Use an existing connection (see tOWLIMConnection)
    - Input: none
    - Output: A stream of RDF variable bindings
- `tOWLIMOutput` – writes/deletes a sequence of triples passed by a stream
    - Configuration parameters
        - In process or remote RDF repository
        - RDF Storage path
        - Path to Sesame TTL configuration file
        - Use an existing connection (see tOWLIMConnection)
        - Insert/delete operation
    - Input: RDF triples
    - Output: none
- `tOWLIMConnection` – a helper component that enables the OWLIM/LarKC data layer connection reuse
    - Configuration parameters
        - In process or remote RDF repository
        - RDF Storage path
        - Path to Sesame TTL configuration file
    - Input: none
    - Output: none
- `tOWLIMConnectionClose` – a helper component that flushes and closes an OWLIM connection
    - Configuration parameters
        - Component identifier to close
    - Input: none
    - Output: none
- `tSPARQLConnection` – reads from SPARQL endpoint and streams triples
    - Configuration parameters
        - SPARQL endpoint URL
    - Input: none
    - Output: A stream of RDF variable bindings

Figure 6 presents a simple workflow that resolves "Reference Node" patterns presented in Figure 2. The `tSPARQLEndpoint_1` component streams all *X*, *id* and *db* values selected with a SPARQL query. The `tSPARQLEndpoint_2` component loads all *Y* values and passes them to a `tMap` (a standard ETL component) that joins the results and produces triples of type: **`<X> skos:exactMatch <Y>.`**

The designed job can be exported as code generated Java classes or directly as Java ARchive (JAR) file, which than can be packed as LarKC data transformer plug-in.

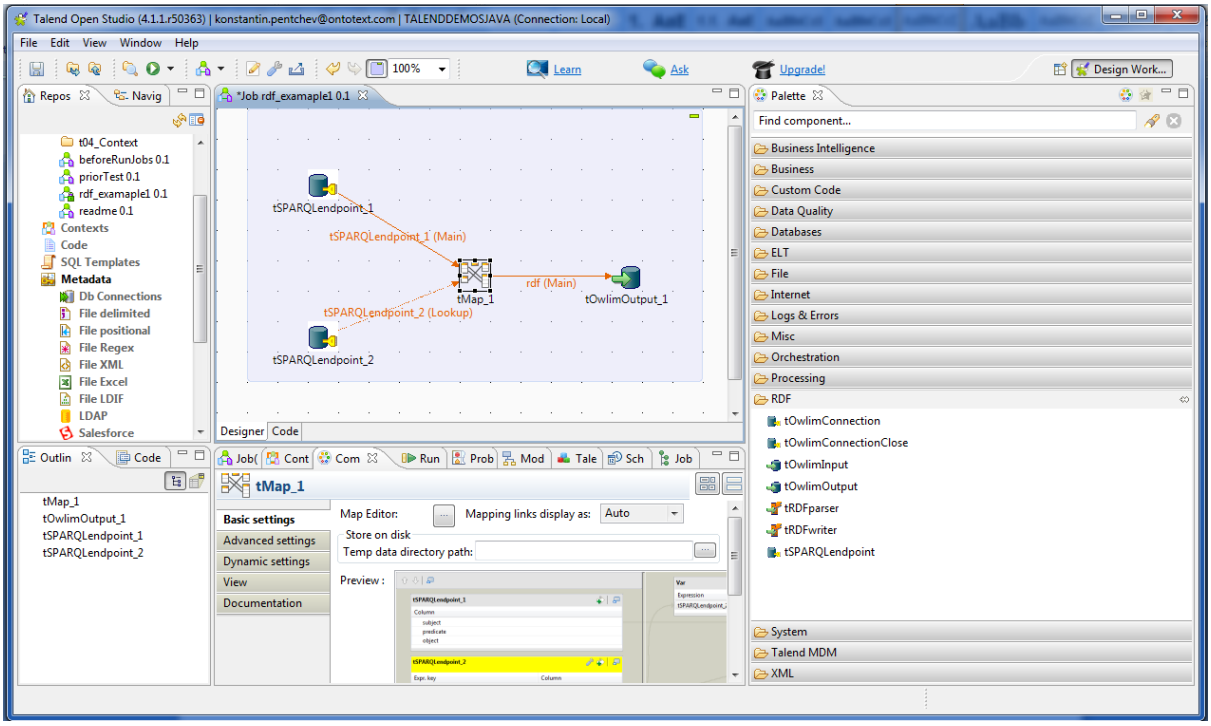

Figure 6 Job for the resolution of the Reference Node pattern.

3.3. TalenD Studio Performance Overheads

The performance of the TalenD components was evaluated on a Windows 7 32-bit system, with Intel® Core™ i5 CPU M520 @ 2.40 Ghz and 4GB RAM.

The `tSPARQLEndpoint` component uses a low-level XML API to send HTTP get requests containing SPARQL queries to a remote SPARQL endpoint. This simple TalenD job is designed to query 100k statements from the LLD repository and write them to a delimited file. The process completes with an average speed of 12,371 statements per second.

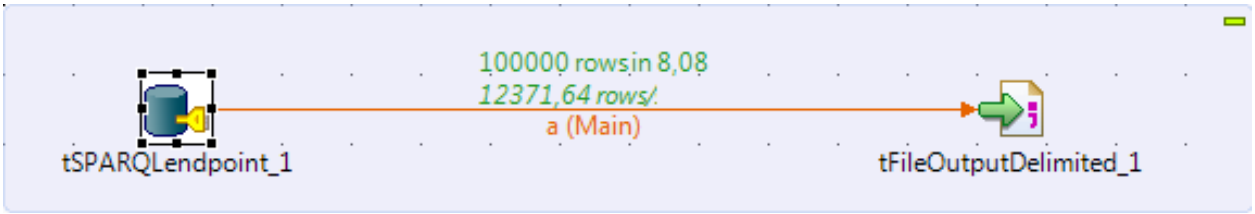

Figure 7 The execution time for a simple RDF processing job, implemented with TalenD

To measure if any significant overhead was present in the specific implementation or due to the code-generating nature of TalenD, the same job was designed and implemented using standard Java code. The execution completes with an average speed of 12,698 statements per second.

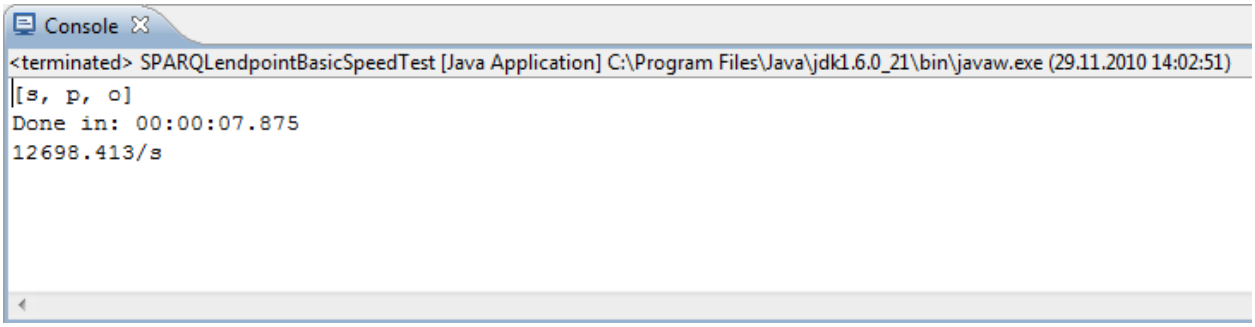

Figure 8 The execution time for a simple RDF processing job, implemented manually

The performance overhead of a code generated TalenD RDF job is less than 3% slower than the performance of the written java code. We consider this performance overhead negligible and comparable to the network performance fluctuations.

4. Future Work and Conclusion

This document presents a second update of the WP7a software prototype. The LLD service shows excellent scalability in terms of data scale and number of supported users. The upgrade of the LLD update process, using data transformers generated by a graphical user interface, lowers the integration effort to add new data sources and allows simpler automatic updates. The service has demonstrated very good stability and has its regular users. A constraining factor for a wider adoption is the expert-level interface and the limited number of available end-user applications such as Relfinder[3] and Context-aware Linked Life Data search[4].

Future work will be considered in two main directions:

- Better tools for computerized interpretation of the information by non-expert users – the SPARQL query interface provides a very powerful but also challenging option for the end-users. The development of new queries is a time consuming process requiring technical, biomedical and schema expertise. This might be improved by introduction of a new integration meta-schema describing causative relationships, by integration of active learning (WP3) and better selection (WP2) methods, and finally by improvements in the existing user interface such as Relfinder.
- Gather feedback and improve the existing information – the present interface does not support mechanisms for correcting data by the end user. The process is currently executed only during the knowledge base design. The development of a feedback mechanism will improve the existing information and make the evaluation of different information retrieval and selection methods easier.

The LLD prototype is available at http://linkedlifedata.com with all the latest software version and additional resources.

[3] http://linkedlifedata.com/relfinder
[4] http://www.wici-lab.org/wici/context-aware-LLD

References

[1] Linked Life Data Service - http://linkedlifedata.com
[2] Momtchev V. et al., D7a.3.1 Prototype v1
[3] Miles A., and Bechhover S., SKOS Simple Knowledge Organization System Reference, available at: http://www.w3.org/TR/skos-reference/
[4] Gallizo G., et al., D5.3.2 Overall LarKC architecture and design v1
[5] Gardner S. Ontologies and semantic data integration, Drug Discovery Today, Vol. 10, No. 14. (15 July 2005), pp. 1001-1007.
[6] Zhao J., et al., The Way to Go for Biological Data Integration, Data Integration in the Life Sciences (2009), pp. 47-54.
[7] Langerger A., et al, A Semantic Web Middleware for Virtual Data Integration on the Web, The Semantic Web: Research and Applications (2008), pp. 493-507.
[8] Bizer C., et al. The D2RQ Platform v0.7 - Treating Non-RDF Relational Databases as Virtual RDF Graphs, available at: http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/
[9] Das S., et al, R2RML: RDB to RDF Mapping Language, W3C Working Draft 28 October 2010
[10] Erling O., Mapping Relational Data to RDF in Virtuoso, White paper available at: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSSQLRDF