LarKC

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

FP7 – 215535

# D9.7 Report of Contributions to Standardisation Activities


**Coordinator: Alexey Cheptsov**

**With contributions from: Matthias Assel, Axel Tenschert, Gregory Katsaros (HLRS), Bosse Andersson (AZ), Larizgoitia (UIBK), Yi Zeng (WICI), Zhisheng Huang (VUA), Frank van Harmelen (VUA), Vassil Momtchev (ONTO)**

**Quality Assessor: Michael Witbrock**

**Quality Controller: Alice Carpentier**



| | |
|---|---|
| Document Identifier: | LarKC/2008/D9.7/v1.0 |
| Class Deliverable: | LarKC EU-IST-2008-215535 |
| Version: | |
| Date: | |
| State: | |
| Distribution: | |

# EXECUTIVE SUMMARY

This deliverable summarises the activities related to standardisation efforts carried out during the project's lifetime. The standards adopted by the LarKC consortium as well as contribution to the existing standards are discussed. An overview on LarKC standardization efforts, including involving in standardization bodies and groups, participation in standardization meetings, and other activities is provided.

# 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** | 9.7 | **Title** | Report of Contributions to Standardisation Activities |
| **Work Package** | **Number** | 9 | **Title** | Exploitation and Standardisation |

| | | | | |
|---|---|---|---|---|
| **Date of Delivery** | **Contractual** | M 42 | **Actual** | M 42 |
| **Status** | version 1.0 | | final □ | |
| **Nature** | prototype □ report □ dissemination □ | | | |
| **Dissemination level** | public □ consortium □ | | | |

| | | | | |
|---|---|---|---|---|
| **Authors (Partner)** | Alexey Cheptsov | | | |
| **Responsible Author** | **Name** | | **E-mail** | cheptsov@hlrs.de |
| | **Partner** | HLRS | **Phone** | |

| | |
|---|---|
| **Abstract (for dissemination)** | |
| **Keywords** | Standardisation |

| **Version Log** | | | |
|---|---|---|---|
| **Issue Date** | **Rev. No.** | **Author** | **Change** |
| 06.07.2011 | 0.1 | Matthias Assel | First draft |
| 06.08.2011 | 0.2 | Alexey Cheptsov, Axel Tenschert, Gregory Katsaros | Parallelisation standards |
| | | Bosse Anderson | W3C activities |
| | | Iker Larizgoiti | Semantic plug-in description models |
| 09.05.2011 | 0.3 | Yi Zeng | E-FOAF:interest Vocabulary and its application |
| 09.08,2011 | 0.4 | | |
| 09.09.2011 | 0.5 | Zhisheng Huang and Frank van Harmelen | OWLLink and other activities |
| | | Vassil Momtchev | Ontotext's standardisation activities |
| 12.09.2011 | 0.6 | Alexey Cheptsov | Ready for Quality Assessment |
| 17.09.2011 | 0.7 | Axel Tenschert | Quality Assessment comments incorporated |
| 29.09.2011 | 1.0 | Alexey Cheptsov | Final version to be submitted to EC |

# Project Consortium Information

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

| | | |
|---|---|---|
| VRIJE UNIVERSITEIT AMSTERDAM | | Frank van Harmelen,<br>VRIJE UNIVERSITEIT AMSTERDAM,<br>Amsterdam, Netherlands,<br>Email: Frank.van.Harmelen@cs.vu.nl |
| THE INTERNATIONAL WIC INSTITUTE, BEIJING UNIVERSITY OF TECHNOLOGY | | Ning Zhong,<br>THE INTERNATIONAL WIC INSTITUTE,<br>Mabeshi, Japan,<br>Email: zhong@maebashi-it.ac.jp |
| INTERNATIONAL AGENCY FOR RESEARCH ON CANCER | | Paul Brennan,<br>INTERNATIONAL AGENCY FOR RESEARCH ON CANCER,<br>Lyon, France,<br>Email: brennan@iarc.fr |
| INFORMATION RETRIEVAL FACILITY | | John Tait,<br>INFORMATION RETRIEVAL FACILITY<br>Vienna, Austria<br>Email : john.tait@ir-facility.org |
| UNIVERSITATEA TEHNICA CLUJ-NAPOCA | | Sergiu Nedevschi,<br>UNIVERSITATEA TEHNICA CLUJ-NAPOCA,<br>Cluj-Napoca, Romania,<br>Email:<br>Sergiu.Nedevschi@cs.utcluj.ro |
| SOFTGRESS S.R.L. | | Ioan Toma,<br>SOFTGRESS S.R.L.,<br>Cluj-Napoca, Romania,<br>Email: ioan.toma@softgress.com |

# Table of Contents

## List of Tables

## List of Acronyms

| Acronym | Description |
| --- | --- |
| GAT | Grid Application Toolkit |
| RDF | Resource Description Framework |
| MPI | Message-Passing Interface |
| OS | Operating System |
| RIF | Rule Interchange Format |
| WP | Workpackage |
| WSMO | Web Service Modelling Ontology |
| WS-RF | Web Service Resource Framework |
| W3C | World Wide Web Consortium |

# 1. Introduction

The success of LarKC, which has resulted in its being adopted for a number of application scenarios both inside and outside the project's scope, imposes an essential requirement on the LarKC architecture – it must provide a development environment incorporating relevant standards that have been elaborated within the Semantic Web community. Adoption of standards is a substantial challenge for reasoning applications. LarKC overcomes this limitation by integrating distinct standards into the platform, used for building plug-ins, workflows, and applications, so that the developers are provided with built-in platform features that seamlessly provide standards support in a ready-to-use and most up-to-date form, thereby promoting the construction of solutions based on them.

To achieve this goal and to provide interoperable solutions the project needed to be aware of standardisation activities in all the relevant domains, whether it is Semantic Web, Rapid Prototyping Frameworks, or Parallel and High Performance Computing. WP9 aimed at ensuring that LarKC was built upon up-to-date standards for which there is a low probability of deprecation or replacement, to lower the adoption barriers of the LarKC platform and other developed technologies. The primary role of WP9 was to identify the standards to be incorporated into the platform and communicate them to WP5 for inclusion into the platform's framework. And, since input to standards processes is an excellent way of encouraging exploitation of the project's results, the second central objective of WP9 was supporting and reporting LarKC contributions to standardization activities.

To successfully complete both identified tasks, the WP9 partners established liaisons with relevant standardization bodies to ensure that the work of LarKC was kept in line with current developments in international standards, and also that the results of LarKC feed into the standardization efforts. It identified and documented deficiencies in current standards, and made recommendations to standards bodies to influence standardization groups. Finally, it reported on standardization activities to the project board.

This deliverable is the Report on Contribution to Standardisation Activities. Hence, it is a summary of all standardization activities conducted by the project in Months 1-40. The report is structured as follows. Section 2 discusses the standards adopted by the LarKC consortium. Section 3 discusses the standards elaborated within the LarKC project itself and contributed to external standardization bodies and working groups. Section 4 discusses the activities surrounding the adoption  and contribution of standards. In Section 5, we discuss our conclusions and highlight the directions for future standards activities around an enduring LarKC infrastructure.

# 2. Adoption of Standards in LarKC

This section discusses the standards adopted by the LarKC consortium, followed by descriptions of their usage and application and by an overview of related documentation.

## 2.1. Semantic models for plug-in description

Within the LarKC platform, plug-ins are essentially pieces of software that can be run locally or remotely in order to contribute to a semantic data processing workflow. The application of semantic service description models, thus, allows for leveraging further synergies between working with plug-ins in LarKC and working with services in service-oriented architectures, by enabling code designed for use in SOAs to be more conveniently repurposed for use with LarKC.
Based on LarKC-specific requirements, a plug-in and workflow annotation language has been developed as a natural extension of previous work in and standards for Semantic Web services. For example, for LarKC annotation languages, the use of standards like RDF for the description of plug-ins and workflows has the positive effect of offering a very flexible and extensible approach to the specification of LarKC processes. The efforts at integrating SOA standards within LarKC include:

- W3C member submission WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web. (http://www.w3.org/Submission/WSMO-Lite/)
- Incorporation of results for the minimal service model that was established in the SOA4All project.
- Efforts around Linked Open Services or Linked Data Services. These works promote the use of RDF and SPARQL to describe and realize REST-based services that emphasize the Semantic Web aspects of Semantic Web services.

More details about the use of these initiatives can be found in deliverable D1.3.1 [1] and D1.3.2 [2].

## 2.2. Message-Passing Interface

The massive amount of data LarKC is designed to process requires means for effective exploitation, which is a big challenge not only because of the size but also due to the nature of this data. Even after applying incomplete reasoning techniques, elaborated by LarKC, the complexity of the reasoning algorithms over those data constitutes a major challenge for the Semantic Web applications. To cope with this challenge, several parallelisation standards have been adopted by the LarKC platform. The most promising among those standards in terms of the scalability and performance speed-up capabilities has proved to be the Message-Passing Interface (MPI).
MPI is a language independent standard [3] of the process-based implementation of parallel applications, and has become a de-facto standard for parallel applications running in supercomputing environments. MPI follows a process-oriented parallelisation, whereby processes communicate by means of messages transmitted between (a so-called “point-to-point” communication) or among (involving several or even all processes, a so called “collective” communication) the nodes. Normally, one process is executed on a single computing node. If any of the processes needs to send/receive data to/from other processes, it should call a corresponding MPI communication function. Both the point-to-point and collective communications available for MPI processes are documented in the MPI standard.
MP7I has been implemented in many programming languages, including Java, which is the main implementation language used in LarKC. One of the most successful MPI implementations is considered to be mpiJava[1][4], which came out of the HPJava project. Considering a number of requests coming from the LarKC application community, mpiJava has been adopted by LarKC as well. The development of mpiJava is currently led by HLRS, and a number of improvements to mpiJava have been made within LarKC, including an installer package, and integration with the Open MPI library. mpiJava is also provided by the BW-Grid cluster of HLRS, which is used for the parallel LarKC application deployment. Moreover, mpiJava is provided as a single installation package for the Windows OS, hence facilitating local development and debugging. Among the LarKC applications

[1] https://sourceforge.net/projects/mpijava/

implemented by means of mpiJava are the most performance critical and computation demanding statistical semantics applications[2].

To facilitate the spread-out and adoption of MPI in the Semantic Web community, LarKC has led several dissemination and exploitation activities, including:

- Setting up a development environment for mpiJava at SourceForge
- Organization of parallelisation workshops during the consortium meeting in Lyon[3] and
- during the ESWC conference[4]

### 2.3. Grid Access Toolkit

Distributed plug-in execution is the core of the LarKC architecture. To enable the seamless execution model of plug-ins in the distributed computing environments, also involving high performance and cloud-based resources, the Grid Access Toolkit (GAT) and its Java realization JavaGAT[5] has been adopted by the LarKC platform. GAT offers a unified API set that allows any software component, i.e. a plug-in, to be executed on several types of remote resource.

The architecture of JavaGAT allows this unification to be achieved by means of special "adapters", as described in D5.3.3 [5], which extend the core API to support diverse access mechanisms such as ssh or gridssh. During adoption of GAT, a number of issues have been identified, mainly on the "adapter" level. Moreover, a need for a new adaptor for accessing production high performance systems, such as those at HLRS, has arisen.

Therefore the LarKC consortium has actively contributed to GAT standardization activities, by participating in the common meetings, requirement analysis, and collaborative development.

### 2.4. AspectJ

Instrumentation of LarKC plugins, workflows and platform is done using AspectJ[6], a de-facto standard for Aspect-Oriented Programming[7] available for the Java programming language. Aspect-Oriented Programming provides methods and tools for the modularization of concerns at the level of the source code. AspectJ uses ASM to do the actual byte-code manipulation and it is closely integrated with various development environments such as Eclipse. From a conceptual point of view, AspectJ adds to java several constructs: point-cuts, advice, inter-type declarations and aspects. Point-cuts and advice dynamically affect program flow, inter-type declarations statically affect a program's class hierarchy, and aspects encapsulate these new constructs. Using AspectJ, LarKC instrumentation and monitoring injects instrumentation code into specific locations, using byte-code manipulation. The aop.xml file defines the aspects to apply on code at runtime and is read at start-up by aspectjweaver javaagent. Basic measurements are obtained by instrumenting specific classes from LarKC such as those that correspond to platform start-up, workflow creation, query execution, plugin execution, etc. Because aspect oriented programming allows manipulation of a program without interfering with its core function, it seems likely to become more widely adopted as a standard for code instrumentation, control, and monitoring.

### 2.5. Rule Interchange Format

Rule Interchange Format (RIF)[8] is a W3C Recommendation that is used in the Semantic Web community to represent and exchange rules between rule systems. RIF comes in many flavours or dialects including RIF Datatypes and Built-Ins (DTB), RIF Core, RIF Framework for Logic Dialects (FLD), RIF Basic Logic Dialect (BLD) and RIF Production Rules Dialect (PRD). The implementation of parallel IRIS [6] developed in LarKC consumes arbitrary rules in the standardized RIF-Core format and processes them in parallel by compiling them to a series of MapReduce[9] operations after initial

[2] http://wiki.larkc.eu/LarkcProject/statisticalSemantics/parallelisation
[3] http://wiki.larkc.eu/LarKCProject/LyonMeeting11#ParallelisationWorkshop.2CTuesday01-Feb-11
[4] http://tw.rpi.edu/hpcsw2011/
[5] http://www.cs.vu.nl/ibis/javagat.html
[6] http://www.eclipse.org/aspectj/
[7] http://en.wikipedia.org/wiki/Aspect-oriented_programming
[8] http://www.w3.org/2005/rules/wiki/RIF_Working_Group
[9] http://labs.google.com/papers/mapreduce.html

optimization. Part of the implementation is an integrated RIF parser[10] that provides support for RIF rules.

[10] http://sourceforge.net/projects/rif4j

# 3. LarKC Contribution to Standards

In addition to adopting and evaluating standards, LarKC succeeded in contributions to certain standards specifications. This Chapter presents the most important such LarKC contributions to standards. Types of contributions include activities such as writing a specification, developing an extension, providing a use case, or providing experience reports or comments

## 3.1. OWLLink

VUA is a co-submitter of the OWLLink Protocol to W3C. The OWLlink protocol provides an implementation-neutral mechanism for accessing OWL reasoner functionality. OWLlink is a refinement of the DIG protocol most notably with respect to query and language expressivity. It relies on OWL 2 for the primitives of the modelling language, and is thus fully compatible with OWL. The OWLlink core covers basic reasoner management, assertion of axioms and elementary ask functionality. An extension mechanism is provided to easily add any required functionality in a controlled way to the core language. A concrete binding is based on the OWL 2 XML-Serialization OWL/XML transported via HTTP. Further defined bindings are the OWLlink HTTP/Functional binding making use of the OWL Functional syntax and the OWLlink HTTP/S-Expression binding. The DIG interface and the OWLLink protocol have been used in the development of the LarKC Reasoner Plugins.

## 3.2. C-SPARQL

In the following, we report ondiscussions the C-SPARQL group have been holding with the SPARQL 1.1 working group about "the Semantics of Aggregation in C-SPARQL and SPARQL 1.1". Essentially the same content was previously delivered in [7]. We repeat it here in order to make this document self-contained.

### 3.2.1. Introduction

In this section, we comment on the semantics of aggregation in C-SPARQL and SPARQL 1.1. Over the last two years we have developed an extension to SPARQL for continuous querying over streams of RDF (namely C-SPARQL). Central to stream processing is support for aggregates. SPARQL 1.0 has no standard support for aggregation; for this reason we have defined and implemented [8] our own support for aggregates in SPARQL which is orthogonal to the other stream processing features of C-SPARQL. In spring and summer 2010, trusting that such extension can be of general interest for the SPARQL 1.1 Working Group[11], we participated in discussions of SPARQL 1.1 support for aggregation.
In the rest of the section we first introduce support for aggregates in C-SPARQL, then we compare C-SPARQL and SPARQL 1.1 support for aggregates. We show that:

- C-SPARQL syntax for aggregates appears more compact and easily applied than the SPARQL 1.1 one,
- all SPARQL 1.1 queries with aggregates can be expressed in C-SPARQL, and
- that there are queries that can be expressed in C-SPARQL but for which it is unclear whether they can be expressed in SPARQL 1.1.

### 3.2.2. Support For Aggregates In C-SPARQL

Aggregation clauses in C-SPARQL are added at the end of the query, and have the following syntax:

```
AggregateClause --> ( "AGGREGATE {(" var "," Function "," Group ")" [Filter] "}" )*
 Function --> "COUNT" | "SUM (" var ")" | "AVG (" var ")" | "MIN (" var ")" | "MAX (" var ")"
 Group --> var | "{" var ( "," var )* "}"
```

Every aggregation clause has the following three parts:

- the first part is a new variable (i.e., a variable not in the WHERE clause or in other aggregation clauses);

[11] http://www.w3.org/2009/sparql/wiki/Main_Page

- the second part is an aggregation function (one of: COUNT, MAX, MIN, SUM, AVG); COUNT may have no argument, while the other functions take one of the variables occurring in the WHERE clause as argument and
- the third part is a set of one or more variables, which are chosen among those occurring in the WHERE clause. These variables express the grouping criteria.

Every clause may also have an optional fourth part, a FILTER clause.

### 3.2.3. Example Of Simple Support For Aggregates In C-SPARQL And SPARQL

Data:

```
@prefix : <http://books.example/> .
:auth1  :name "Alice Foo", :writesBook :book1; :book2 .
:auth2  :name "Bob Bar", :writesBook :book2 .
```

The following query counts the number of books written by an author and returns the name and the number of books.

```
SELECT ?name ?numberOfBooks
WHERE {
     ?auth :name ?name .
     ?auth :writesBook ?book .
}
AGGREGATE { (?numberOfBooks, COUNT, {?auth} ) }
```

The semantics of a query containing aggregates consists in adding new variable bindings computed by the WHERE clause to the existing regular variable bindings. For each of the new variables introduced by the AGGREGATE clauses, one new variable binding is added. The query result constructed in this way may be further filtered by a standard FILTER clause, which may refer to all the variables introduced in the WHERE and AGGREGATE clauses.
Our C-SPARQL extension is based on the conviction that in the context of RDF, knowledge should be extended rather than shrunk. Therefore, we propose to generate additional variable bindings and use them to annotate any existing variable binding that contributed to the aggregate value.

Results:

| ?name | ?numberOfBooks |
|---|---|
| **"Alice Foo"** | 2 |
| **"Bob Bar"** | 1 |

This is in contrast to the conventional SQL grouping semantics that replaces all aggregated tuples with a single tuple representing the aggregate value. In this respect, we believe that our approach to aggregation is more aligned with the baseline of the SPARQL semantics.
Judging fro the example of the SPARQL 1.1 draft, the query above can be expressed in SPARQL 1.1 in the following way:

```
SELECT ?name ?numberOfBooks
WHERE {
      ?auth :name ?name .
           {
              SELECT ?auth (COUNT(?book) AS ?numberOfBooks)
              WHERE {
                      ?auth :writesBook ?book .
              }
              GROUP BY ?auth
      }
}
```

In the C-SPARQL language all the variables used in the aggregation function or in the grouping set of AGGREGATE clauses must also appear in the SELECT clause, since aggregation happens after standard SPARQL query evaluation. In SPARQL 1.1 the constraint is not specified.
Wrapping up, queries of this kind can be expressed both in SPARQL 1.1 and C-SPARQL, but C-SPARQL syntax appears more compact and handy.

### 3.2.4. Aggregates Supported In SPARQL 1.1 Are Also Supported In C-SPARQL

Given the current SPARQL 1.1 support for aggregates, it appears that all SPARQL 1.1 queries with aggregates can be expressed in C-SPARQL.
For instance, the following query is the example of SPARQL 1.1 support:

```
PREFIX  <http://books.example/>
SELECT (SUM(?lprice) AS ?totalPrice)
WHERE {
   ?org :affiliates ?auth .
   ?auth :writesBook ?book .
   ?book :price ?lprice .
} GROUP BY ?org HAVING (SUM(?lprice) > 10)
```

Such a query in C-SPARQL will be written as follows.

```
PREFIX  <http://books.example/>
SELECT ?totalPrice
WHERE {   ?org :affiliates ?auth .
   ?auth :writesBook ?book .
   ?book :price ?lprice .
}
AGGREGATE { (?totalPrice, SUM(?lprice), {?org}) FILTER ( ?totalPrice > 10) }
```

### 3.2.5. Introduction Queries That Can Be Expressed In C-SPARQL but for which It Is Unclear Whether They Can Be Expressed In SPARQL 1.1

Given the current SPARQL 1.1 support for aggregates, it is unclear whether the following C-SPARQL queries can be expressed in SPARQL 1.1

Query: the average number of books written by authors that wrote at least 5 books.

```
SELECT ?name ?book ?numberOfBooks ?averageNumberOfBooks
WHERE {
     ?auth :name ?name .
     ?auth :wrote ?book .
}
AGGREGATE { (?numberOfBooks, COUNT, {?auth} ) FILTER (?numberOfBooks > 5) }
AGGREGATE { (?averageNumberOfBooks, AVG, {?numberOfBooks} ) }
```

A possible way to express it in SPARQL 1.1 is illustrated hereafter, but no examples in the current draft show that this is possible.

```
SELECT ?name ?surname ?book ?numberOfBooks (AVG(?numberOfBooks) AS ?averageNumberOfBooks)
WHERE {
      ?auth :hasSurname ?surname .
      ?auth :hasName ?name .
      {
              SELECT ?auth (COUNT(?book) AS ?numberOfBooks)
              WHERE {
                      ?auth :wrote ?book .
      }
      GROUP BY ?auth
      HAVING (?numberOfBooks > 5)
}
```

More complex sequences of aggregation are supported in C-SPARQL, such as

- computing the number of books per author
- keeping only the authors who have published at least 5 books
- computing the total number of books grouping by affiliation
- filtering the affiliation with less than 50 books published

```
SELECT ?name ?surname ?book ?numberOfBooks ?averageNumberOfBooks, ?auth, ?organization
WHERE {
     ?auth :name ?name .
     ?auth :surname ?surname .
     ?auth :wrote ?book .
     ?auth :affiliated ?organization .
}
AGGREGATE { (?numberOfBooks, COUNT, {?auth} ) FILTER (?numberOfBooks > 5) }
```

```
AGGREGATE { (?affiliationBooks, SUM(?numberOfBooks), {?organization} )
            FILTER (?affiliationBooks > 50)}
```

A possible way to express it in SPARQL 1.1 is illustrated hereafter, but as above no examples in the current draft show that this is possible.

```
SELECT ?name ?surname ?book ?numberOfBooks
WHERE {
    ?auth :hasSurname ?surname .
    ?auth :hasName ?name .
    {
     SELECT ?affiliation (SUM(?numberOfBooks) as ?affiliationBooks)
     WHERE {
      ?auth :affiliated ?organization .
      {
       SELECT ?auth (COUNT(?book) AS ?numberOfBooks)
       WHERE {        ?auth :wrote ?book .
       }
       GROUP BY ?auth
       HAVING (?numberOfBooks > 5)
     }
     GROUP BY ?organization
     HAVING (?affiliationBooks > 50)
    }
```

In C-SPARQL, evaluation of multiple aggregation with filtering clauses is possible. For instance, one can ask for the research topics for which the Italian authors are more than the Swiss ones.

```
SELECT ?topic ?numberOfSwissAuthors ?numberOfItalianAuthors
WHERE {
     ?auth :name ?name .
     ?auth :wrote ?book .
     ?book :topic ?topic .
     ?auth :hasNationality ?nat .
}
AGGREGATE { FILTER(?nat = 'IT') (?numberOfItalianAuthors, COUNT, {?topic} ) }
AGGREGATE { FILTER(?nat = 'CH') (?numberOfSwissAuthors, COUNT, {?topic} )
FILTER(?numberOfItalianAuthors>?numberOfSwissAuthors)}
```

A possible way to express it in SPARQL 1.1 is illustrated hereafter (see line 23), but as above no examples in the current draft show that this is possible. Moreover, it requires the SPARQL 1.1 engine to decide the order of execution, whereas in C-SPARQL the order is given explicitly.

```
1. SELECT ?topic ?numberOfSwissAuthors ?numberOfItalianAuthors
2. WHERE {
3.     {
4.             SELECT ?topic (COUNT(?book) AS ?numberOfSwissAuthors)
5.             WHERE {
6.                     ?auth :wrote ?book .
7.                     ?book :topic ?topic .
8.                     ?auth :hasNationality ?nat .
9.                     FILTER(?nat = 'CH') .
10.             }
11.             GROUP BY ?topic
12.     }
13.     {
14.             SELECT ?topic (COUNT(?book) AS ?numberOfItalianAuthors)
15.             WHERE {
16.                     ?auth :wrote ?book .
17.                     ?book :topic ?topic .
18.                     ?auth :hasNationality ?nat .
19.                     FILTER(?nat = 'IT') .
20.             }
21.             GROUP BY ?topic
22.     }
23.     FILTER(?numberOfItalianAuthors>?numberOfSwissAuthors)
24. }
```

Therefore, we believe that there are queries that can be expressed in C-SPARQL but not in SPARQL 1.1.

### *3.2.6. Computing Multiple Independent Aggregates at the Same Time*

As we explained in the introduction to this segment, C-SPARQL was explicitly designed for processing RDF streams. The transient nature of streams imposes the requirement to compute

multiple (possibly independent) aggregates at the same time in the same query, because assuring that two independent queries process exactly the same data is very difficult. Therefore, multiple independent aggregations are also allowed within the same C-SPARQL query, with different grouping criteria and different partitions over the same set of bindings, thus pushing the aggregation capabilities beyond those of SQL.
The following query counts the number of books written by an author, counts the number of authors per book and returns the name, the book, the number of books and the number of authors.

```
SELECT ?name ?book ?numberOfBooks ?numberOfAuthors
WHERE {
     ?auth :name ?name .
     ?auth :wrote ?book .
}
AGGREGATE { (?numberOfBooks, COUNT, {?auth} ) }
AGGREGATE { (?numberOfAuthors, COUNT, {?book} ) }
```

Results:

| ?name | ?book | ?numberOfBooks | ?numberOfAuthors |
|---|---|---|---|
| **"Alice Foo"** | b1 | 2 | 1 |
| **"Alice Foo"** | b2 | 2 | 2 |
| **"Bob Bar"** | b2 | 1 | 2 |

Judging from SPARQL 1.1 draft, the query above can be expressed in SPARQL 1.1 in the following way:

```
SELECT ?name ?surname ?book ?numberOfBooks ?numberOfAuthors
WHERE {
      ?auth :hasSurname ?surname .
      ?auth :hasName ?name .
      {
              SELECT ?auth (COUNT(?book) AS ?numberOfBooks)
              WHERE {
                      ?auth :wrote ?book .
              }
              GROUP BY ?auth
      }
      {
              SELECT  ?auth (COUNT(?auth) AS ?numberOfAuthors)
              WHERE {
                     ?auth :wrote ?book .
              }
              GROUP BY ?book
      }
}
```

### *3.3. E-Foaf:Interest*

User interests play central roles in (Semantic) Web applications. There are many user related data on the Web in the form of public user profiles, etc. Nevertheless, currently there are no unified and well accepted standards in the semantic Web community concerning the description of user interests. Since LarKC has a target of providing the search and reasoning results that users really want, we emphasize that user interests are important factors as indicators of the contextual environment of LarKC end users. Hence, we need standardization efforts that push the end user to provide user- interest related semantic data in the form that we need. In addition, use-cases such as those adopted within the LarKC project will benefit from interests-based plugins if the datasets follow the proposed vocabulary.
The e-foaf:interest Vocabulary is aimed at extending the FOAF vocabulary on user interests evaluated from different perspectives. It focuses on extending “foaf:interest” by providing more detailed vocabularies related to user interests. The efforts on e-foaf:interest Vocabulary have been introduced in LarKC D2.3.2 [9] (corresponding to version 0.1). It has been reviewed or used in several other semantic Web projects and applications. It has also been compared and integrated with other existing interests-related vocabularies. In order to make a final report on this effort, here we extract some part of the efforts from D2.3.2 and make some refinements as well as necessary changes to the previous versions.

The e-foaf:interest Vocabulary has been developed within the LarKC project, currently lead by WICI/BJUT and with participation from VUA and USFD. This is still an ongoing standardization effort, and the most updated version can be found from http://wiki.larkc.eu/e-foaf:interest .
Following LarKC D2.3.2, the E-FOAF:interest vocabulary has 3 branches that meet different levels of requirements, namely: “e-foaf:interest Basic”, “e-foaf:interest Complement”, and “e-foaf:interest Complete”. They are composed of a set of class vocabulary items and a set of property vocabulary items, as shown in Table 1. The main difference of version 0.3 from previous versions is that the content of Table 2, which reflects additional supplementary recommendation vocabularies has been added to this standardization efforts. Namely, the E-FOAF:interest vocabulary list reported in D2.3.2 is a subset of the version in this deliverable (version 0.3).

**Table 1 The e-foaf:interest vocabulary list**

| e-foaf:interest Complete | | | |
|---|---|---|---|
| **e-foaf:interest Basic** | | **e-foaf:interest Complement** | |
| **Vocabulary** | **Type** | **Vocabulary** | **Type** |
| e-foaf:interest | Class | e-foaf:cumulative_interest_value | Attribute |
| e-foaf:interest_value | Attribute | e-foaf:retained_interest_value | Attribute |
| e-foaf:interest_value_updatetime | Attribute | e-foaf :interest_longest_duration | Attribute |
| e-foaf:interest_appeared_in | Attribute | e-foaf:interest_cumulative_duration | Attribute |
| e-foaf:interest_appeare_time | Attribute | | |
| e-foaf:interest_has_synonym | Attribute | | |
| e-foaf:interest_co-occur_with | Attribute | | |

The “e-foaf:interest Complete” serves as the complete version of the vocabulary and it is the union of the set of vocabularies from “e-foaf:interest Basic” and “e-foaf:interest Complement”.
Concrete definitions of each vocabulary has been introduced in LarKC D2.3.2. Here we are not going to review each of them, and we only extract two definitions to illustrate how we defined them in the previous deliverable.
Following lines are needed for the definition of each vocabulary listed below:

```
  <!ENTITY e-foaf  “http://www.wici-lab.org/wici/e-foaf-interest/e-foaf.owl#” >
  <!ENTITY owl  “http://www.w3.org/2002/07/owl#” >
  <!ENTITY xsd  “http://www.w3.org/2001/XMLSchema#” >
  <!ENTITY rdf  “http://www.w3.org/1999/02/22-rdf-syntax-ns#” >
  <!ENTITY foaf  “http://xmlns.com/foaf/spec/#” >
....
  xmlns:e-foaf = “http://www.wici-lab.org/wici/e-foaf-interest/e-foaf.owl#”
  xmlns:owl = “http://www.w3.org/2002/07/owl#”
  xmlns:rdf = ”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
  xmlns:rdfs = ”http://www.w3.org/2000/01/rdf-schema#”
  xmlns:xsd = ”http://www.w3.org/2001/XMLSchema#”>
  xmlns:foaf = “http://xmlns.com/foaf/spec/#”>  ….
```

Below we give some details on the definition of each vocabulary element.

*** e-foaf:interest_value** (Property)

*Definition*: “e-foaf:interest_value” represents the value of an interest. The value of a specified interest is a real number. If the Agent is interested in an interest, the interest value is greater than zero (namely a positive number), if not, the interest value is smaller than zero (namely a negative number).

```
<owl:DatatypeProperty rdf:ID=”e-foaf:interest_value”>
    <rdfs:domain rdf:resource=”#e-foaf:interest” />
    <rdfs:range  rdf:resource=”&xsd;number” />
```

```
</owl:DatatypeProperty>
```

*Note*: This property is supposed to be oriented to interest values from any perspective if the user do not care much about the types of interest values (Some possible perspectives to evaluate interests values are defined in “e-foaf:interest Complement”, namely, the perspective of “cumulative interest value”, “retained interest value”, “interest longest duration”, “interest cumulative duration”, etc. It can also be a user defined value.

*** e-foaf:interest_appeared_in** (Property)

*Definition*: “e-foaf:interest_appeared_in” represents where the interest appeared in.

```
<owl:DatatypeProperty rdf:ID=”e-foaf:interest_appeared_in”>
  <rdfs:domain rdf:resource=”#e-foaf:interest” />
  <rdfs:range  rdf:resource=”foaf:Document” />
</owl:DatatypeProperty>
```

*Note*: The motivation of this property is to help to keep the original resources where the interests come from and be reused for calculation of interest value when needed. The design of this property is inspired by “from” in Attention Profiling Markup Language [APML2011].

The following is an example that represents the appearance of the interest “Web” in one of Frank van Harmelen’s Tweets.

<http://www.w3.org/2006/03/wn/wn20/instances/wordsense-web-noun-5> <http://www.wici-lab.org/wici/e-foaf-interest/interest_appeared_in> <http://twitter.com/FrankVanHarmele/status/57763991233105920>.

In LarKC Deliverable 2.3.2, we present an illustrative example on the use of the proposed vocabulary by a use case related to “Scientific Research Interests Description” from the DBLP data. Hence, concrete use of these vocabularies can be found there.

It is very clear that the listed vocabularies in Table 1 cannot cover all necessary vocabularies that are needed to describe user interests, such as basic description on the user (e.g. name, affiliation, etc.). We follow the spirit of the Semantic Web, and many existing vocabularies are reused. As additional supplementary for D2.3.2, Table 2 listed necessary vocabularies that the E-FOAF:interest Vocabulary recommend to use when one wants to create user interests related RDF triples. Although these vocabularies are from other standards, the purpose for us to do this is to adopt public agreed standard vocabularies and reuse them as many as possible.

**Table 2 Supplementary vocabulary for the E-FOAF:interest vocabulary**

| | |
|---|---|
| http://www.w3.org/2006/03/wn/wn20/instances/synset-end_user-noun-1 | This vocabulary is used to denote that the specified user is the end user of an application. Namely, in this use case, interests related data are around this user. |
| http://www.w3.org/2002/07/owl#sameAs | This vocabulary is used to denote that two interests, or two persons, etc. are exactly the same. |
| http://purl.org/dc/elements/1.1/content | This vocabulary is used to represent the interests source contents where the user interests are originated. |
| http://linkedgeodata.org/ontology/latitude | This vocabulary is used to represent the latitude of an interest object (if any) |
| http://linkedgeodata.org/ontology/longitude | This vocabulary is used to represent the longitude of an interest object (if any) |
| http://xmlns.com/foaf/0.1/publication | This vocabulary is used to represent various publications of a specific user (e.g. papers, tweets, Facebook notes, etc.) |
| http://purl.org/dc/elements/1.1/location | This vocabulary is used to represent an agent’ location. |

The vocabulary “synset-end_user-noun-1”, it is of great importance for user-driven and user-centric LarKC applications. If the applications do not become aware of who the end user is, they cannot provide personalized services for the user. Following is an example of this vocabulary, which shows that the user Frank van Harmelen (identified by his Twitter account) is the end user of a LarKC powered application.

<http://twitter.com/FrankVanHarmele> <http://www.w3.org/2002/07/owl#InstanceOf> <http://www.w3.org/2006/03/wn/wn20/instances/synset-end_user-noun-1>.

"owl:sameAs" is important for interest integration and interest value calculation. If two interests URIs denote exactly the same interests, they need to be merged together as one. Following is an example on the use of "owl:sameAs"

<http://www.openvocabulary.info/taxonomies/dmoz/all/Top/Health/Medicine> <http://www.w3.org/2002/07/owl#sameAs> <http://www.w3.org/2006/03/wn/wn20/instances/wordsense-medicine-noun-3>.

If two URIs denote exactly the same person, reasoning process need to be performed based on "owl:sameAs" and interests related data need to be integrated together for the calculation of the specific user's interests.

<http://www.facebook.com/frank.van.harmelen> <http://www.w3.org/2002/07/owl#sameAs> <http://twitter.com/FrankVanHarmele>.

The vocabulary "dc:content" is used to represent the original content of the interest source. Following is an example for a tweet of the user Frank van Harmelen.

<http://twitter.com/status/Text_101037553347600384> <http://www.w3.org/2000/01/rdf-schema#label> "?@Richard Cyganiak ?@grechaw "Turtle, it's a published working draft! http://www.w3.org/TR/turtle/"".

User interests sometimes are associated with locations. In this kind of scenario, each interest of this kind has a latitude and a longitude. Here we borrow vocabularies from the LinkedGeoData ontology. An illustrative example is described as follows:

<http://data.semanticweb.org/organization/shanghai-jiao-tong-university-china>
<http://linkedgeodata.org/ontology/latitude> "31.063522".
<http://data.semanticweb.org/organization/shanghai-jiao-tong-university-china>
<http://linkedgeodata.org/ontology/longitude> "121.453857".

User interests may be extracted from various sources that the user published, while each of them can be considered as a special type of publication from the user. Hence, we choose to use "foaf:publication" to represent these publications.

<http://twitter.com/FrankVanHarmele> <http://xmlns.com/foaf/0.1/publication> <http://twitter.com/FrankVanHarmele/status/57763991233105920>.

A specific user or other kinds of agent may have some location terms that are used to specify where they are. Here we use "dc:location", and following is an illustrative example related to Frank van Harmelen extracted through Twitter API.

<http://twitter.com/FrankVanHarmele> <http://purl.org/dc/elements/1.1/location> "Amsterdam".

Recently, the proposed E-FOAF:interest vocabulary as well as its supplementary vocabulary have been used in the active academic visit recommendation application powered by the LarKC platform (The application is an interests-driven recommendation system based on Semantic Web Dog Food data and Twitter data, etc. More details can be acquired in LarKC D2.7.3).

### *3.4. Tagging and Grouping in RDF Datasets*

The LarKC data layer is a core platform component responsible for providing RDF middleware capabilities like passing by reference information between plug-ins. Such a usage scenario is the selector plug-in, which has to subset the complete RDF graph and efficiently pass it to the next plug-in in the workflow's path. One prominent extension of the RDF data model is the usage of *named graphs* defined in **Fehler! Verweisquelle konnte nicht gefunden werden.**. The quadruple support is now widely recognized by the industry and all major RDF repositories currently implement it in order to overcome the issues of reification and support the management of *datasets*, integration of information from multiple data sources. **Fehler! Verweisquelle konnte nicht gefunden werden.** proposes semantics for the add and remove statement operations of **<S,P,O,G>** quadruples and motivates the further extension of the RDF model with *tripleset* also known as labelled set of statements.

The data layer implements the specified semantics [10] and has provided a full reference implementation since Release 1.0.

# 4. LarKC Standardization Activity

This section discuss the standardization activities conducted by the LarKC consortium's partners including some not described in the previous sections.

## *4.1. Involvement in Standardisation Bodies and Groups*

This section discusses the involvements of the LarKC members in different standardisation bodies and main achievements in them. The following table (Table 3) summarizes the partner involvement in the standardisation bodies discussed below.

**Table 3 LarKC partner involvement in standardisation bodies**

| Partner | Standardisation body | Short description |
|---|---|---|
| UIBK | W3C (RIF WG) | The Rule Interchange Format (RIF) Working Group was chartered by the World Wide Web Consortium in 2005 to create a standard for exchanging rules among rule systems, in particular among Web rule engines. RIF focused on exchange rather than trying to develop a single one-fits-all rule language because, in contrast to other Semantic Web standards, such as RDF, OWL, and SPARQL. |
| AZ | W3C | The W3C World Wide Web Consortium (http://www.w3.org/) is an international community that develops open standards to ensure the long-term growth of the Web. A number of standards recommended and elaborated by W3C are adopted by LarKC. |
| HLRS | MPI-Forum | The MPI Forum (www.mpi-forum.org) is an open group with representatives from many organizations that define and maintain the MPI (Message-Passing Interface) standard. The MPI standard is one of the major assets towards scaling up the Semantic Web applications. |
| VUA | OWLLink | The OWLlink protocol (http://www.owllink.org/) provides an implementation-neutral mechanism for accessing OWL reasoner functionality. OWLlink is a refinement of the DIG protocol most notably with respect to query and language expressivity. It relies on OWL 2 for the primitives of the modeling language, and is thus fully compatible with OWL. The OWLlink core cover basic reasoner management, assertion of axioms and elementary ask functionality. An extension mechanism is provided to easily add any required functionality in a controlled way to the core language. A concrete binding is based on the OWL 2 XML-Serialization OWL/XML transported via HTTP. Further defined bindings are the OWLlink HTTP/Functional binding making use of the OWL Functional syntax and the OWLlink HTTP/S-Expression binding. |
| WICI/BJUT | Web Intelligence | The Web Intelligence Consortium (WIC) |

| | | |
|---|---|---|
| | Consortium | (http://wi-consortium.org/) is an international, non-profit organization dedicated to advancing world-wide scientific research and industrial development in the field of Web Intelligence (WI). It organizes and promotes standardizations for implementing Artificial Intelligence on the Web. |

Below we discuss the main standardization bodies listed in Table 3.

#### 4.1.1. W3C

There is a wide range of W3C standards applicable to LarKC and, therefore, several different groups are of interest for the project, such as the Health-Care and Life Sciences interest group (HCLS).
AstraZeneca participates in the Health-Care and Life Sciences interest group (HCLS), which develops, advocates for and support the usage of Semantic Web technologies in health-care, translational medicine, and bio-sciences. AstraZeneca participate in the following initiatives of W3C:

- The Linking Open Drug Data initiative (LODD), which aims at using state-of-the-art semantic link discovery techniques for interlinking publicly available datasets with drug information (Linked data) and demonstrate how researchers in life science, as well as physicians and patients can take advantage of the connected data sets.
- The Translational Medicine task force (TMO), which aims to demonstrate how information-based translational medicine activities can be made easier and more effective using semantic web technologies. TMO's main activity is to develop a prototype knowledge base that captures terminology and expert knowledge as formal ontologies for which automated reasoning becomes possible, and have the ontology act as a global schema for the growing amounts of open, linked biomedical data that are part of the Semantic Web.

The HCLS interest group, in particular the TMO and LODD initiatives, share a common interest with LarKC's use cases 7a and 7b on linking and reusing (publicly available) drug information. The LinkedLifeData knowledge repository has been presented and discussed with the HCLS interest group.
The UIBK partner is involved into the work of W3C-RIF working group, which is focused on defining a standard for exchanging rules among rule systems, in particular among Web rule engines. UIBK's collaboration has been focused on developing tools to support this standard, more precisely on RIF4J[12], a reasoning engine for RIF-BLD that provides a Java object model for RIF-BLD and supports the parsing and serialization of RIF-BLD formulas. Furthermore, it provides a prototype implementation of a RIF-BLD consumer based on the Datalog engine IRIS. This tool is of special interest for LarKC because it enables the reuse of rules among infrastructure plug-ins that perform reasoning following different techniques or algorithms.

#### 4.1.2. WIC

Web Intelligence Consortium (WIC) holds more than 18 branch organizations and special interests groups located in different countries or regions world wide (China, Japan, USA, Australia, Poland, Korea, Canada, etc.). In each branch organizations, groups with similar research interests are brought together in the form of joint projects, visiting programs, etc. Standardization is one of the efforts that the organization is dedicated to.
The LarKC partner, International WIC Institute (WICI)/Beijing University of Technology proposed and leads the E-FOAF:interest standardization effort (reported in Section 3.2 of this deliverable) within the LarKC project. As an affiliated member of WIC, the International WIC Institute (WICI)/Beijing University of Technology also submitted this effort as a candidate for Knowledge

[12] http://rif4j.sourceforge.net/

Science and Problem Solving standardization to the Web Intelligence Consortium. Now it is a recommended standard of the Web Intelligence Consortium[13].

### 4.2. Participation at Standardisation Meetings

The following meetings related to the standardisation activities conducted by LarKC and described in sections above have been attended by the LarKC partners.

- HLRS
  - Annual MPI forums
  - Annual D-Grid meetings as well as meetings of the GAT working group (in the frame of German Grid association)
- VUA
  - Regular OWLLink meetings
  - FOAF forums
  - W3C Semantic Web Mailing lists
- W3C Workshop "RDF Next Steps”, 2010 Stanford

[13] http://www.wi-consortium.org/wicweb/html/tools.html

# 5. Conclusions

This deliverable contains the final version of the report on LarKC standardisation activities. It concentrates on successful LarKC collaboration with standardisation bodies, project experiences and contributions. In particular, the document contains descriptions of standards adopted in LarKC including experiences and recommendations, specific contributions to standards performed by LarKC partners, and a summary of the general involvement of LarKC partners in standardisation activities during the whole project lifetime.

The LarKC consortium's members were not only active in adoption of the existing standards such as "traditional" RDF and OWL but also contributed towards creation, enhancement and promotion of the new standards. The prominent example of the latter is C-SPARQL, which is a SPARQL extension for streaming operations support.

Moreover, LarKC has striven to establish new standards for the Semantic Web applications such as MPI. MPI is very promising in terms of the performance and scalability expectations for the parallel applications implemented with it. The identified standards have been integrated in the design and realization of the LarKC platform's architecture, thus seamlessly enabling those standards for the LarKC's plug-ins, workflows, and applications, which was the essential requirement of almost all application scenarios.

The standardization activities beyond the project's scope were also in focus of the LarKC partners, who contributed within standardization bodies spanning a wide range from the W3C working groups dealing with the future of internet application to the MPI forum aiming among others at opening the high performance infrastructures to the Semantic Web applications of future.

We believe that the reported standardization activities done within the project's initial scope will also be continued beyond the LarKC project proper. The success gained in LarKC will definitively serve as a source of guiding principles for several standards to be elaborated, including the above-mentioned C-SPARQL and MPI.

## 6. References

[1] Barry Bishop, Dumitru Roman, Barry Bishop, Dumitru Roman. D1.3.1 Initial LarKC Plugin Annotation Language and Methodology, LarKC Deliverable 2010.
[2] Ioan Toma, Irene Celino. D1.3.2 Improved LarKC Plugin Annotation Language and Methodology, LarKC Deliverable 2011.
[3] The MPI standard version 1.1. Available at http://www.mcs.anl.gov/research/projects/mpi/mpi-standard/mpi-report-1.1/mpi-report.htm
[4] Getov, V., Gray, P., Sunderam, V.S.: MPI and Java-MPI: contrasts and comparisons of low-level communication performance. ACMIEEE 1999 Conference Supercomputing, pp. 1--16, ACM/IEEE (1999)
[5] Matthias Assel, Alexey Cheptsov, Blasius Czink, Christoph Fuchs, Norbert Lanzanasto, Vassil Momtchev, Spyros Kotoulas, Luka Bradesko, Blaz Fortuna, Ioan Toma: D5.3.3 Final LarKC architecture and design, LarKC Deliverable 2011.
[6] Florian Fischer, Ioan Toma, Valer Roman, Adrian Marte, Iker Larizgoitia. D4.4.2 Implementation of Rule-based Reasoning Plug-in, LarKC Deliverable 2011.
[7] Emanuele Della Valle, Davide Barbieri, Daniele Braga, Stefano Ceri. D3.7 Implemented plug in for Data Stream management approaches based on aggregation techniques, LarKC Deliverable 2011.
[8] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, Michael Grossniklaus: C-SPARQL: a Continuous Query Language for RDF Data Streams. Int. J. Semantic Computing 4(1): 3-25 (2010)
[9] Jose Quesada, Yi Zeng, Ralph Brandao, Lael Schooler, Stefan Otte, Yi Zeng, Yan Wang, Zhisheng Huang, Ning Zhong, Danica Damljanovic. D2.3.2 Cognitive memories components v2, LarKC Deliverable 2010.
[10] Florian Fischer, Uwe Keller, Atanas Kiryakov, Zhisheng Huang, Vassil Momtchev, Elena Simperl, Dieter Fensel, Dumitru Roman. D1.1.3 Initial knowledge representation formalism, LarKC Deliverable 2010.