Towards Service-Oriented Resource Discovery by means of Semantic Web Reasoning

Alexey Cheptsov

High Performance Computing Center Stuttgart (HLRS), University of Stuttgart, Nobelstrasse 19, 70569 Stuttgart, Germany

cheptsov@hlrs.de

Abstract. Reasoning is one of the essential tools of the modern Semantic Web. A number of applications for resource discovery on the Web such as random indexing enjoy a prominent place in face of the novel Semantic Web Reasoning trends. However, the reasoning algorithms are dealing with significant challenges when scaled up to the problem sizes addressed by the modern Semantic Web application. As such, they are not well-optimized to be applied to the emerging Internet-scale knowledge bases. We introduce a solution to building highly efficient and scalable reasoning applications based on the Large Knowledge Collider – a service-oriented incomplete reasoning platform breaking the scalability barriers of the existing solutions. We discuss the application of incomplete reasoning for the resource discovery tasks and demonstrate a service-oriented realization for the query expansion and subsetting algorithms based on the random indexing knowledge extraction technique.

Keywords: Random Indexing, Semantic Web Reasoning, Large Knowledge Collider.

1 Introduction

The large- and internet-scale data applications is a primary challenge for the Semantic Web, and in particular for reasoning algorithms, used for processing exploding volumes of data, exposed currently on the Web. Reasoning is the process of making implicit logical inferences from the explicit set of facts or statements, which constitute the core of any knowledge base. The key problem for most of the modern reasoning engines such as Jena [1] or Pellet [2] is that they can not efficiently be applied for the real-life data sets that consist of tens, sometimes of hundreds of billions of triples (a unit of the semantically annotated information), which can correspond to several petabytes of digital information. Whereas modern advances in the Supercomputing domain allow this limitation to be overcome, the reasoning algorithms and logic need to be adapted to the demands of rapidly growing data universe, in order to be able to take advantages of the large-scale and on-demand infrastructures such as high performance computing or cloud technology. On the other hand, the algorithmic principals of the reasoning engines need to be reconsidered as well in order to allow for very large volumes of data. Service-oriented architectures (SOA) can greatly contribute to this goal, acting as the main enabler of the newly proposed reasoning techniques such as incomplete reasoning [3]. This paper focuses on a service-oriented solution for constructing Semantic Web applications of a new generation, ensuring the drastic increase of the scalability for the existing reasoning applications, as elaborated by the Large Knowledge Collider (LarKC)[1] EU project.

The paper is organized as follows. In Section 2, we collect our consideration towards enabling the large-scale reasoning and its application for the resource discovery tasks. In Section 3, we discuss LarKC – a service-oriented platform for development of fundamentally new reasoning application, with much higher scalability barriers as by the existing solutions. In Section 4, we introduce some successful resource discovery applications implemented with LarKC, such as Random Indexing. In Section 5, we discuss our conclusions and highlight the directions for future work in highly scalable semantic reasoning.

2 Semantic Reasoning on the Web Scale

Despite the majority of data on the Web is available as an unstructured text, e.g. generated from the content kept in RDBM, the application areas of the modern Semantic Web spawn a wide range of domains, from social networks to large-scale Smart Cities projects in the context of the future internet [4][5]. However, data processing in such applications goes far beyond a simple maintenance of the collection of facts; based on the explicit information, collected in datasets, and simple rule sets, describing the possible relations, the implicit statements and facts can be acquired from those datasets.

Many data collections as well as application built on top of them allow for rule-based inferencing to obtain new, more important facts. The process of inferring logical consequences from a set of asserted facts, specified by using some kinds of logic description languages (e.g., RDF/RDFS and OWL[2]), is in focus of semantic reasoning. The goal is to provide a technical way to determine when inference processes is valid, i.e., when it preserves truth. This is achieved by the procedure which starts from a set of assertions that are regarded as true in a semantic model and derives whether a new model contains provably true assertions.

The latest research on the Internet-scale Knowledge Base Technologies, combined with the proliferation of SOA infrastructures and cloud computing, has created a new wave of data-intensive computing applications, and posed several challenges to the Semantic Web community. As a reaction on these challenges, a variety of reasoning methods have been suggested for the efficient processing and exploitation of the semantically annotated data. However, most of those methods have only been approved for small, closed, trustworthy, consistent, coherent and static domains, such as synthetic LUBM [6] sets. Still, there is a deep mismatch between the requirements on the

[1] http://www.larkc.eu/

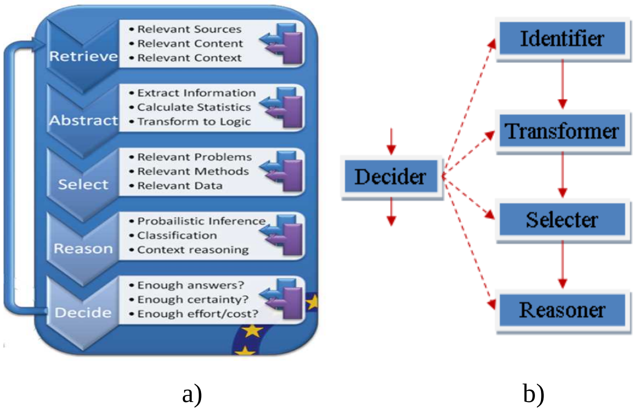
[2] http://www.w3.org/TR/owl-ref/

real-time reasoning on the Web scale and the existing efficient reasoning algorithms over the restricted subsets.

Whereas unlocking the full value of the scientific data has been seen as a strategic objective in the majority of ICT- related scientific activities in EU, USA, and Asia [7], the “Big Data” problem has been recognized as the primary challenger in semantic reasoning [8][9]. Indeed, the recent years have seen a tremendous increase of the structured data on the Web with scientific, public, and even government sectors involved. According to one of the recent IDC reports [10], the size of the digital data universe has grown from about 800.000 Terabytes in 2009 to 1.2 Zettabytes in 2010, i.e. an increase of 62%. Even more tremendous growth should be expected in the future (up to several tens of Zettabytes already in 2012, according to the same IDC report [10]).

The “big data” problem makes the conventional data processing techniques, also including the traditional semantic reasoning, substantially inefficient when applied for the large-scale data sets. On the other hand, the heterogeneous and streaming nature of data, e.g. implying structure complexity [11], or dimensionality and size [12], makes big data intractable on the conventional computing resource [13]. The problem becomes even worse when data are inconsistent (there is no any semantic model to interpret) or incoherent (contains some unclassifiable concepts) [14].

The broad availability of data coupled with increasing capabilities and decreasing costs of both computing and storage facilities has led the semantic reasoning community to rethink the approaches for large-scale inferencing [15]. Data-intensive reasoning requires a fundamentally different set of principles than the traditional mainstream Semantic Web offers. Some of the approaches allow for going far beyond the traditional notion of absolute correctness and completeness in reasoning as assumed by the standard techniques. An outstanding approach here is interleaving the reasoning and selection [16]. The main idea of the interleaving approach (see Fig. 1a) is to introduce a selection phase so that the reasoning processing can focus on a limited (but meaningful) part of the data, i.e. perform incomplete reasoning.



a) b)

Fig. 1. Incomplete reasoning, the overall schema (a) and the service-oriented vision (b)

As discussed before, the standard reasoning methods are not valid in the existing configurations of the Semantic Web. Some approaches, such as incomplete reasoning, offer a promising vision how a reasoning application can overcome the “big data” limitation, e.g. by interleaving the selection with the reasoning in a single “workflow”, as shown in Fig. 1a. However the need of combining several techniques within a single application introduces new challenges, for example related to ensuring the proper collaboration of team of experts working on a concrete part of the workflow, either it is identification, selection, or reasoning. Another challenge might be the adoption of the already available solutions and reusing them in the newly developed applications, as for example applying selection to the JENA reasoner [1], whose original software design doesn’t allow for such functionality. The SOA approach can help eliminate many of the drawbacks on the way towards creating new, service-based reasoning applications. Supposed that each of the construction blocks shown in Fig. 1a is a service, with standard API that ensures easy interoperability with the other similar services, quite a complex application can be developed by a simple combination of those services in a common workflow (see Fig. 1b).

Resource discovery is an essential feature of the Semantic Web, which involves tasks of decentralized and autonomous control, distributed service discovery etc. Reasoning can greatly contribute to solving these issues by for example improving the fine-grained service matchmaking, resource ranking, etc. in typical resource discovery workflows [29].

Although utilizing reasoning in the resource discovery workflows is not a new concept for the Semantic Web [17][18], there was quite a big gap in realizing the single steps of the reasoning algorithms (Fig. 1b) as a service. This was due to many reasons, among them complexity of the data dependency management, ensuring interoperability of the services, heterogeneity of the service’s functionality. Realizing a system where a massive number of parties can expose and consume services via advanced Web technology was also a research highlight for Semantic Web. An example of very successful research on offering a part of the semantic reasoning logic as a service is the SOA4ALL[3] project, whose main goal was to study the service abilities of development platforms capable of offering semantic services. Several useful services wrapping such successful reasoning engines as IRIS [19] and several others had been developed in the frame of this project. Nevertheless, the availability of such services is only an intermediate step towards offering reasoning as a service, as a lot of efforts were required to provide interoperability of those services in the context of a common application. Among others, a common platform is needed that would allow the user to seamlessly integrate the service by annotating their dependencies, manage the data dependencies intelligently, being able to specify parts of the execution that should be executed remotely, etc.

An outstanding effort to develop such a platform was performed in the LarKC (Large Knowledge Collider) [20] project. In the following sections, we discuss the main ideas, solutions, and outcomes of this project.

[3] http://www.soa4all.eu/

3 Large Knowledge Collider Approach

In order to create a technology for creation of trend-new applications for large-scale reasoning, several leading Semantic Web research organizations and technological companies have joined their efforts around the project of the Large Knowledge Collider (LarKC), supported by the European Commission. The mission of the project was to set up a distributed reasoning infrastructure for the Semantic Web community, which should enable application of reasoning far beyond the currently recognized scalability limitations [22], by implementing the interleaving reasoning approach. The current and future Web applications that deal with "big data" are in focus of LarKC.

The LarKC's design has been guided by the primarily goal to build a scalable platform for distributed high performance reasoning. Fig. 2 shows a conceptual view of the LarKC platform's architecture and the proposed development life-cycle. The architecture was designed to holistically cover the needs of the three main categories of users – semantic service (plug-in) developers, application (workflow) designers, and end-users internet-wide. The platform's design ensures a trade-off between the flexibility and the performance of applications in order to achieve a good balance between the generality and the usability of the platform by each of the categories of users.

Below we introduce some of the key concepts of the LarKC architecture and discuss the most important platform's services and tools for them.

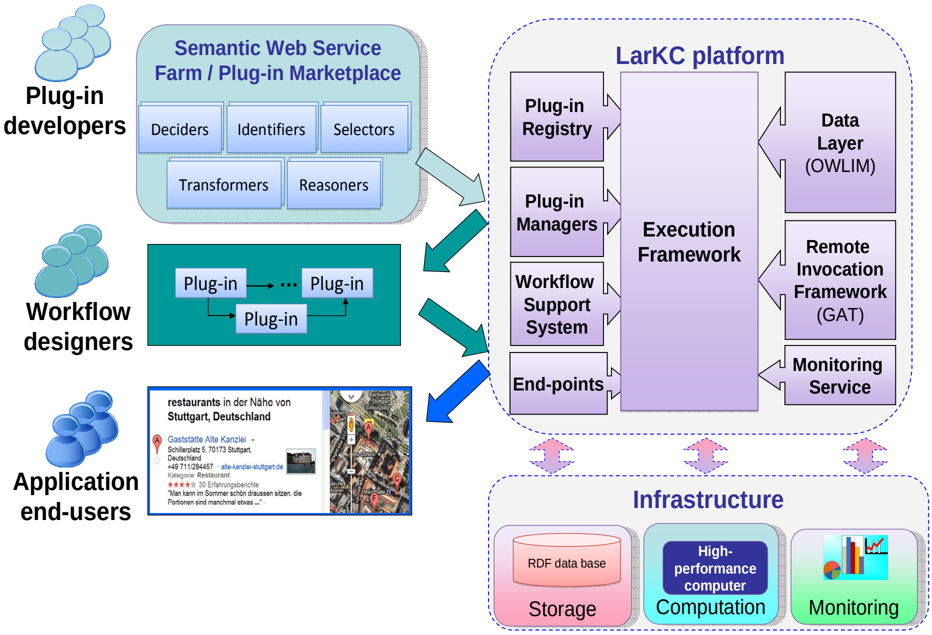


Fig. 2. Architecture of LarKC.

1. Plug-ins

Plug-ins are standalone services implementing some specific parts of the reasoning logic as discussed previously, whether it is selection, identification, transformation, or reasoning algorithm, see more at [21]. In fact, plug-ins can implement much broader functionality as foreseen by the incomplete reasoning schema (Fig. 1), hence enabling the LarKC platform to target much wider Semantic Web user community as originally targeted, e.g. for machine learning or knowledge extraction. The services are referred as plug-ins because of their flexibility and ability to be easily integrated, i.e. plugged into a common workflow and hence constitute a reasoning application. To ensure the interoperability of the plug-ins in the workflows, each plug-in should implement a special plug-in API, based on the annotation language [23]. Most essentially, the API defines the RDF schema (set of statements in the RDF format) taken as input and produced as output by each of the plug-ins. The plug-in development is facilitated by a number of special wizards, such as Eclipse IDE wizard or Maven archetype for rapid plug-in prototyping. The ready-to-use plug-ins are uploaded and published on the marketplace – a special web-enabled service offering a centralized, web-enabled repository store for the plug-ins[4].

2. Workflows

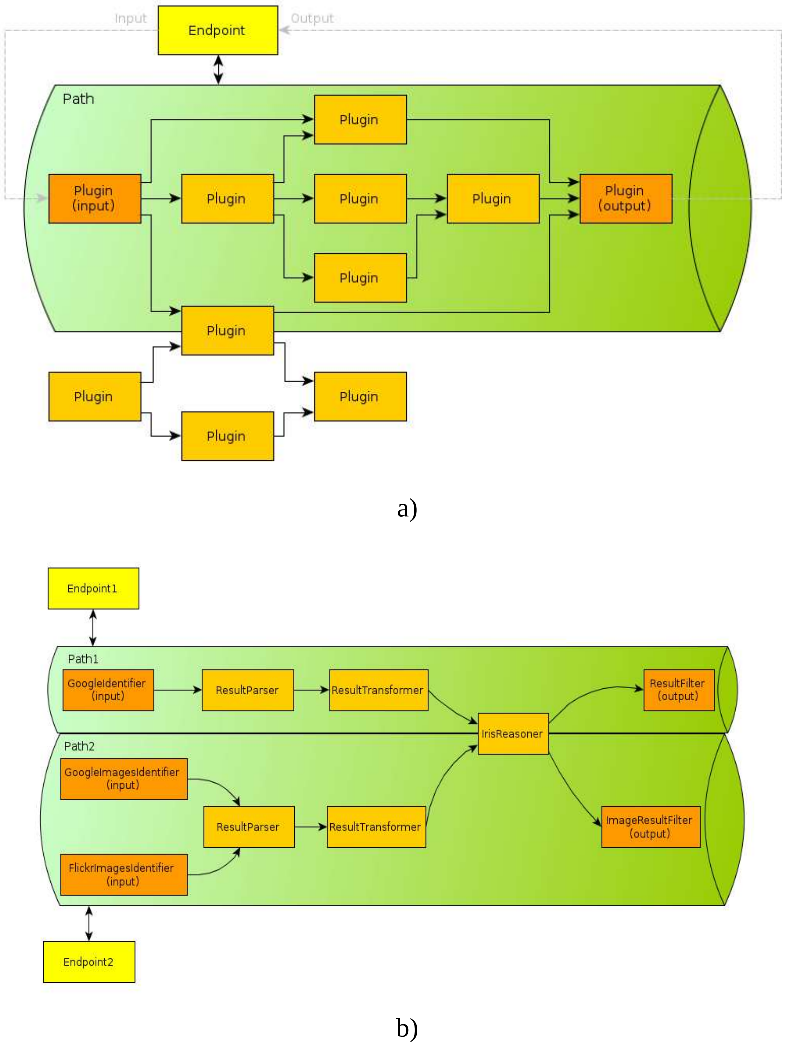
The workflow designers get access to the Marketplace in order to construct a workflow from the available plug-ins, combined to solve a certain task. In terms of LarKC, workflow is a reasoning application that is constructed of the (previously developed and uploaded on the Marketplace) plug-ins. The workflow's topology is characterised by the plug-ins included in the workflow as well as the data- and control flow connections between these plug-ins.

The complexity of the workflow's topology is determined by the number of included plug-ins, data connections between the plug-ins (also including multiple splits and joins such as in Fig. 3a or several end-points such as in Fig. 3b), and control flow events (such as instantiating, starting, stopping, and terminating single plug-ins or even workflow branches comprising several plug-ins). Same as for plug-ins, the input and output of the workflow is presented in RDF, which however can cause compatibility issues with the user's GUI, which are not obviously based on an RDF-compliant representation. In order to confirm the internal (RDF) dataflow representation with the external (user-defined) one, the LarKC architecture foresees special end-points, which are the adapters facilitating the workflow usage in the tools outside of the LarKC platform. Some typical examples of end-points, already provided by LarKC, are e.g. SPARQL end-point (SPARQL query as input and set of RDF statements as output) and HTML end-point (HTTP request without any parameters as input and HTML page as output).

For the specification of the workflow configuration, a special RDF schema was elaborated for LarKC, aiming at simplification of the annotation efforts for the work-

[4] Visit the LarKC Plug-in Marketplace at http://www.larkc.eu/plug-in-marketplace/

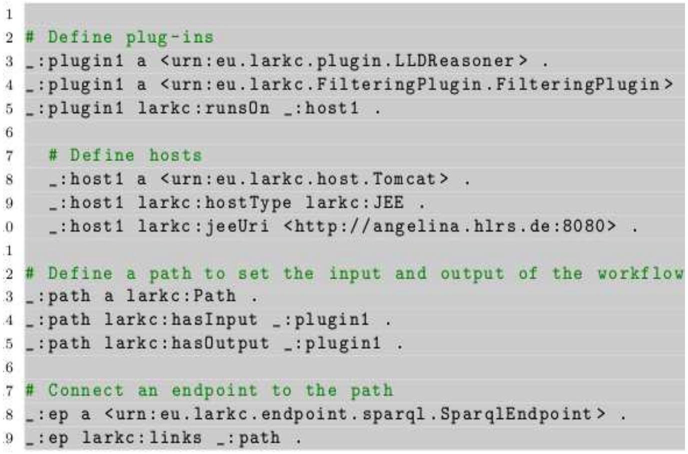
flow designers. Fig. 4a shows a simple example of the LarKC workflow annotation. Creation of the workflow specification can greatly be simplified by using upper-level graphical tools, e.g. Workflow Designer that offers a GUI for visual workflow construction (Fig. 4b) [28]. The elaborated schema makes specification of the additional features such as remote plug-in execution extremely simple and transparent for the users and can be used for tuning the front-end graphical interfaces of the applications to adapt them to the user needs.



a)

b)

Fig. 3. Examples of LarKC workflows: a) workflow with non-trivial branched dataflow (containing multiple splits/joins), b) workflow with multiple end-points



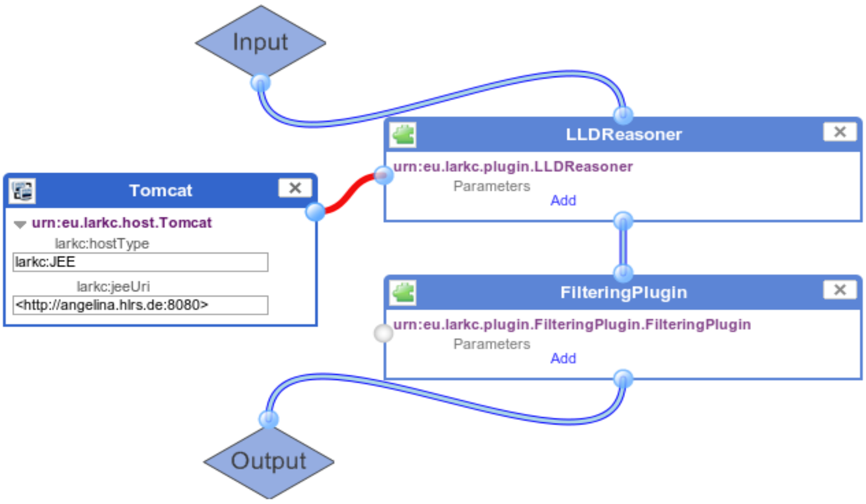
```
# Define plug-ins
_:plugin1 a <urn:eu.larkc.plugin.LLDReasoner> .
_:plugin1 a <urn:eu.larkc.FilteringPlugin.FilteringPlugin>
_:plugin1 larkc:runsOn _:host1 .

  # Define hosts
  _:host1 a <urn:eu.larkc.host.Tomcat> .
  _:host1 larkc:hostType larkc:JEE .
  _:host1 larkc:jeeUri <http://angelina.hlrs.de:8080> .

# Define a path to set the input and output of the workflow
_:path a larkc:Path .
_:path larkc:hasInput _:plugin1 .
_:path larkc:hasOutput _:plugin1 .

# Connect an endpoint to the path
_:ep a <urn:eu.larkc.endpoint.sparql.SparqlEndpoint> .
_:ep larkc:links _:path .
```

a)



larkc:hostType
larkc:JEE

larkc:jeeUri
<http://angelina.hlrs.de:8080>

b)

Fig. 4. Further example of LarKC workflows: a) RDF schema for workflow annotation, b) Workflow Designer GUI with the specification of the remote host

3. Applications

Workflows are already standalone applications that can be submitted to the platform and executed by means of such tools as Workflow Designer discussed above. Nevertheless, workflows can also be wrapped into much more powerful user interfaces, adapted to the needs of the targeted end-user communities, e.g. Urban Computing [24], and using LarKC as a back-end engine. The service-oriented approach makes possible hiding the complexity of the LarKC platform, by enabling its whole power to the end-users through such interfaces. We present an exemplarily LarKC application in Section 4.

4. Platform services

All above-described activities related to plug-in creation, workflow design, and application development are facilitated by an extensive set of the platform services, as shown in Fig. 2. A detailed description of the main LarKC services can be found in our previous publication [21].

4 Application Scenario – Random Indexing

Random indexing [25] is a distributional statistic technique used in resource discovery for extracting semantically similar words from the word co-occurrence statistics in the text data, based on high-dimensional vector spaces (Fig. 5).

Random indexing offers new opportunities for a number of large-scale Web applications performing the search and reasoning on the Web scale [26]. Prominent application using random indexing is subsetting (Fig. 6a) and query expansion (Fig. 6b).

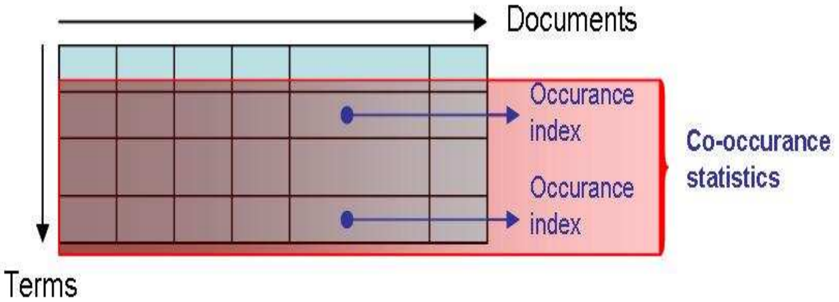
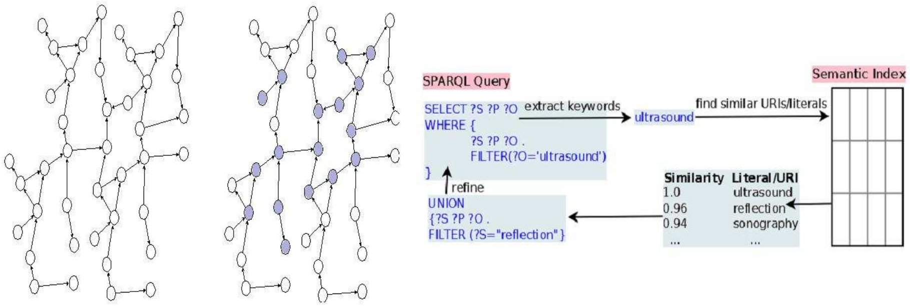


Fig. 5. Schema of the co-occurrence statistical analysis of text corpora.



```
SELECT ?S ?P ?O
WHERE {
        ?S ?P ?O .
        FILTER(?O='ultrasound')
}
```

```
UNION
{?S ?P ?O .
FILTER (?S="reflection"}
```

Similarity	Literal/URI
1.0	ultrasound
0.96	reflection
0.94	sonography
...	...

a) b)

Fig. 6. Application of Random Indexing: a) subsetting b) query expansion.

Query expansion [30] is used in information retrieval with the aim to expand the document collection returned as a result to a query, thus covering the larger portion of the documents. Subsetting (also known as selection) [31], on the contrary, deprecates the unnecessary items from a data set in order to achieve faster processing. Both presented problems are complementary, as change properties of the query to best adapt it to the search needs.

The main complexity of the random indexing algorithms lies in the following:

- High dimensionality of the underlying vector space.

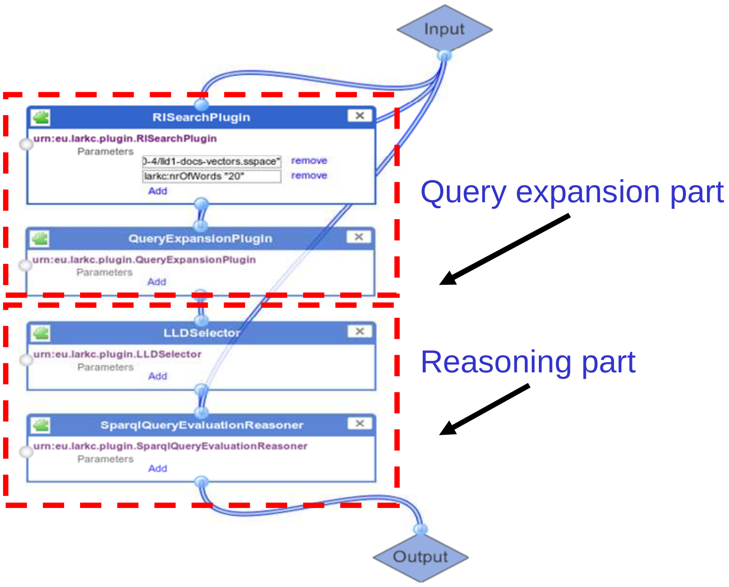
A typical random indexing search algorithm performs traversal over all the entries of the vector space. This means, that the size of the vector space to the large extent defines the search performance. The modern data stores, such as Linked Life Data or Open Phacts consolidate many billion of statements and result in vector spaces of a very large dimensionality. Random indexing over such large data sets is computationally very costly, with regard to both execution time and memory consumption. The latter is of especial drawback for use of random indexing packages on the mass computers. So far, only relatively small parts of the Semantic Web data have been indexed and analyzed.

- High call frequency.

Both indexing and search over the vector space is typically a one-time operation, which means that the entire process should be repeated from scratch every time new data is encountered.

The implementation as a LarKC plug-in allows random indexing to take advantages of the LarKC data and execution model, being seamlessly integrated with the other plug-ins and building up a common workflow. This allows random indexing to be coupled with reasoners to improve the resource discovery algorithm. On the other hand, the reasoning process can also benefit from the integration, for example by using random indexing to expand the initial query and improve the quality of the obtained results, such as shown in Fig. 7.

LarKC is the technology that not only enables the large-scale reasoning approach for the already existing applications, but also facilitates their rapid prototyping with low initial investments, leveraging the SOA approach through the unique platform solutions. Furthermore, LarKC delivers a complete eco-system where the researches from very different domains can team up in order to develop new challenging mashup-applications, e.g. for the resource discovery, hence having a dramatic impact on a lot of problem domains.


Input
RISearchPlugin
urn:eu.larkc.plugin.RISearchPlugin
Parameters
remove
remove
Add
QueryExpansionPlugin
urn:eu.larkc.plugin.QueryExpansionPlugin
Parameters
Add
LLDSelector
urn:eu.larkc.plugin.LLDSelector
Parameters
Add
SparqlQueryEvaluationReasoner
urn:eu.larkc.plugin.SparqlQueryEvaluationReasoner
Parameters
Add
Output
Query expansion part
Reasoning part

)-4/lld1-docs-vectors.sspace"

larkc:nrOfWords "20"

Fig. 7. Realization of query expansion in the Linked Life Data reasoning workflow.

5 Conclusions

We proposed a technology that allows a resource discovery process to be enhanced by integration with the reasoning. The technology is based on the Large Knowledge Collider (LarKC). LarKC is very promising platform for creation of new-generation semantic reasoning applications. The LarKC's main value is twofold. On the one hand, it enables a new approach for large-scale reasoning based on the technique for interleaving the identification, the selection, and the reasoning phases. On the other hand, through over the project's life time (2008-2011), LarKC has evolved in an outstanding, service-oriented platform for creating very flexible but extremely powerful applications, based on the plug-in's realization concept. The LarKC plug-in marketplace has already comprised several tens of freely available plug-ins, which implement new know-how solutions or wrap existing software components to offer their functionality to a much wider range of applications as even originally envisioned by their developers. Moreover, LarKC offers several additional features to improve the performance and scalability of the applications, facilitated through the parallelization, distributed execution, and monitoring platform. LarKC is an open source development, which encourages collaborative application development for Semantic Web. Despite being quite a young solution, LarKC has already established itself as a very promising technology in the Semantic Web world. Some evidence of its value was a series of Europe- and world-wide Semantic Web challenges won by the LarKC applications. It is important to note that the creation of LarKC applications, including the ones discussed in the paper, was also possible and without LarKC, but would have required much more (in order of magnitude) development efforts and financial investments.

We believe that the availability of such platform as LarKC will make a lot of developers to rethink their current approaches for resource discovery as well as semantic reasoning towards their tighter coupling and wider adoption of the service-oriented paradigm.

6 Acknowledgment

This work has been performed for the LarKC project (http://www.larkc.eu), partly funded by the European Commission's IST activity of the 7th Framework Programme (ICT-FP7-215535).

7 References

1. McCarthy, P.: Introduction to Jena. IBM Developer Works, http://www.ibm.com/developerworks/xml/library/j-jena/
2. Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., Katz, Y.: Pellet: a practical owl-dl reasoner. Journal of Web Semantics, http://www.mindswap.org/papers/PelletJWS.pdf
3. Fensel, D., van Harmelen, F.: Unifying Reasoning and Search to Web Scale. IEEE Internet Computing, 11(2), 96--95 (2007).
4. Broekstra, J., Klein, M., Decker, S., Fensel, D., van Harmelen, F., Horrocks, I.: Enabling knowledge representation on the Web by extending RDF schema. Proceedings of the 10th international conference on World Wide Web (WWW '01), ACM, 467--478 (2001).
5. Donovang-Kuhlisch, M.: Smart City Process Support and Applications as a Service – from the Future Internet. Future Internet Assembly 2010, http://fi-ghent.fi-week.eu/files/2010/12/1430-Margarete-Donovang-Kuhlisch.pdf (2010)
6. Guo, Y., Pan, Z., Heflin, J.: LUBM: A Benchmark for OWL Knowledge Base Systems. Web Semantics, 3(2), 158--182 (2005)
7. High Level Expert EU Group: Riding the wave - How Europe can gain from the rising tide of scientific data. Final report, October 2010, http://ec.europa.eu/information_society/newsroom/cf/document.cfm?action=display&doc_id=707
8. Thompson, B., Personick, M.: Large-scale mashups using RDF and bigdata. Semantic Technology Conference (2009)
9. Hustadt, U., Motik, B., Sattler, U.: Data Complexity of Reasoning in Very Expressive Description Logics. Proc. IJCAI 2005, Edinburgh, UK, July 30–August 5 2005. Morgan Kaufmann Publishers, 466--471 (2005)

10. McKendrick, J.: Size of the data universe: 1.2 zettabytes and growing fast, ZDNet.
11. Della Valle, E., Ceri, S., van Harmelen, F., Fensel, D.: It’s a streaming world! Reasoning upon rapidly changing information. IEEE Intelligent Systems, 24(6), 83--89 (2009)
12. Fensel, D., van Harmelen, F.: Unifying Reasoning and Search to Web Scale. IEEE Internet Computing. 11(2), 96--95 (2007)
13. Cheptsov, A., Assel, M.: Towards High Performance Semantic Web – Experience of the LarKC Project. inSiDE - Journal of Innovatives Supercomputing in Deutschland, 9(1), 72--75 (2011)
14. Huang, Z., van Harmelen, F., Teije, A.: Reasoning with inconsistent ontologies. Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI'05, 454--459 (2005)
15. Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Stumme, G., Sure, Y., Tane, J., Volz, R., Zacharias, V.: KAON - Towards a Large Scale Semantic Web. Tjoa, Proceedings of the Third international Conference on E-Commerce and Web Technologies, 304--313 (2002)
16. Huang, Z.: Interleaving Reasoning and Selection with Semantic Data. Proceedings of the 4th International Workshop on Ontology Dynamics (IWOD-10), ISWC2010 Workshop (2010)
17. Deelman, E., Gannon, D., Shields, M., Taylor I.: Workflows and e-Science: An overview of workflow system features and capabilities. Future Generation Computer Systems, 25(5) (2009)
18. Gil, Y., Ratnakar, V., Fritz, C.: Assisting Scientists with Complex Data Analysis Tasks through Semantic Workflows. Proceedings of the AAAI Fall Symposium on Proactive Assistant Agents, Arlington, VA.
19. IRIS - Integrated Rule Inference System - API and User Guide, http://iris-reasoner.org/pages/user_guide.pdf
20. Fensel, D., van Harmelen, F., Andersson, B., Brennan, P., Cunningham, H., Della Valle, E., Fischer, F., Huang, Z., Kiryakov, A., Lee, T., Schooler, L., Tresp, V., Wesner, S., Witbrock, M., Zhong, N.: Towards LarKC: A Platform for Web-Scale Reasoning. Proceedings of the 2008 IEEE international Conference on Semantic Computing ICSC, 524--529 (2008)
21. Assel, M., Cheptsov, A., Gallizo, G., Celino, I., Dell'Aglio, D., Bradeško, L., Witbrock, M., Della Valle, E.: Large knowledge collider: a service-oriented platform for large-scale semantic reasoning. Proceedings of the International Conference on Web Intelligence, Mining and Semantics (2011)
22. Assel, M., Cheptsov, A., Gallizo, G., Benkert, K., Tenschert, A.: Applying High Performance Computing Techniques for Advanced Semantic Reasoning. eChallenges e-2010 Conference Proceedings. Paul Cunningham and Miriam Cunningham (Eds). IIMC International Information Management Corporation (2010)
23. Roman, D., Bishop, B., Toma, I., Gallizo, G., Fortuna, B.: LarKC Plug-in Annotation Language. Proceedings of The First International Conferences on Advanced Service Computing – Service Computation 2009 (2009)
24. Della Valle, E., Celino, I., Dell’Aglio, D.: The Experience of Realizing a Semantic Web Urban Computing Application. T. GIS, vol. 14, iss. 2, 163--181 (2010)
25. Sahlgren, M.: An introduction to random indexing. Proceedings of Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering TKE 2005, 1--9 (2005)

26. Jurgens, D., Stevens, K.: The S-Space Package: An Open Source Package for Word Space Models. Proceedings of the ACL 2010 System Demonstrations, 30--35 (2010)
27. Assel, M., Cheptsov, A., Czink, B., Damljanovic, D., Quesada, J.: MPI Realization of High Performance Search for Querying Large RDF Graphs using Statistical Semantics. Proceedings of the 1st Workshop on High-Performance Computing for the Semantic Web (HPCSW2011), co-located with the 8th Extended Semantic Web Conference, ESWC2011, Heraklion, Greece, May 29 (2011)
28. Le Phuoc, D., Polleres, A., Morbidoni, C., Hauswirth, M., Tummarello, G.: Rapid semantic web mashup development through semantic web pipes. Proceedings of WWW2009 Research Track (2009)
29. Ruta, M.: If objects could talk: novel resource discovery approaches in pervasive environments.
http://www.iaria.org/conferences2010/filesUBICOMM10/MicheleRuta_NexTech2010_Keynote_Speech-2.pdf
30. Efthimiadis, E.: Query Expansion. Martha E. Williams (ed.), Annual Review of Information Systems and Technology (ARIST), v31, 121--187 (1996)
31. Quilitz, B., Leser, U.: Querying Distributed RDF Data Sources with SPARQL. Proceedings of the 5th European Semantic Web Conference (ESWC2008)