

RKB Explorer: Application and Infrastructure

Hugh Glaser and Ian C. Millard

School of Electronics and Computer Science
University of Southampton, UK
{hg, icm}@ecs.soton.ac.uk

Abstract. RKB Explorer is a Semantic Web application that is able to present unified views of a significant number of heterogeneous data sources regarding a given domain. We have developed an underlying information infrastructure which is mediated by ontologies and consists of many independent triplestores. Our current dataset totals many tens of millions of triples, and is publicly available through both SPARQL endpoints and resolvable URIs. To realise the synergy of disparate information sources, we have deployed tools to identify co-referent URIs, and devised an architecture to allow the information to be represented and used. This paper provides a brief overview of the RKB Explorer application, the underlying infrastructure, and a number of associated tools for both knowledge acquisition and publishing.

The screenshot displays the RKB Explorer interface. At the top left is the logo for RESIST (Resilience for Survivability in IST) Knowledge Base Explorer, Version 1.0.2. Below the logo is a navigation bar with tabs for People, Research Areas, Publications, Projects (selected), and Search. To the right of the navigation bar are links for Recently Viewed, Reset, and Help. The main content area is divided into several sections:

- Resilience for Survivability in IST:** A network diagram with a central red node labeled "ReSIST Resilience for Survivability in I...". It is connected to various other nodes, including "NEXT TTA - High-Confidence Architecture", "Network of Excellence in distributed a...", "Predictably Dependable Computing Systems", "Basic Research on Advance...", "Network of Excellence in Distributed Com...", "ReSIST Training and Dissemination Co...", "Assessment of GEN-technology-usability a...", "ReSIST WG Verif...", "Malicious- and Accidental Fault Toleranc...", "Design for validation", "European electronic delicatessen project wiki:wg_actionlist", "ReSIST WG Arch", and "ReSIST Resilience-Explicit Com...". A "Maximize" button is visible in the top right of this section.
- Detail:** A sidebar on the right providing metadata for the selected project:
 - Name:** Resilience for Survivability in IST
 - Funding source:** The European Union
 - Funding amount:** 4500000 EUR
 - Start date:** 2006-01-01
 - End date:** 2008-12-31
- People:** A list of names: Alexei Iliasov, Algirdas Avizienis, Ana Rugina, Andras Balogh, and Andras Batarazs.
- Research Areas:** A list of areas: Telecommunications, Information Processing, and Information Systems.
- Publications:** A list of titles: "Fault Injection for Dependability Validation" and "Fault Injection for Dependability Validation: A...".
- Projects:** A list of project names: "computing systems", "Design for validation", and "The esprit network of excellence in distributed...".

1 Introduction

Large-scale, multi-site, collaborative projects have complex needs in managing their activities and the artifacts and knowledge assets they create. They also have complex requirements in relating to knowledge and research throughout the world related to their business.

ReSIST (Resilience for Survivability in IST) is an EU Network of Excellence in Resilient Systems, and is one such project. From its early conceptions, it was proposed that the entire activity would be supported by a semantically-enabled knowledge infrastructure. The vision was of sets of services and applications for both acquiring and publishing knowledge, working together as a unified coherent resource. Project members and other researchers would be able to explore the knowledge created and acquired from distributed and heterogeneous resources, enabling them to discover relationships and resources that may not previously have been evident. This would no longer be a prototype, as was its predecessor developed during the AKT Project [1] *CS AKTiveSpace* [2, 3], but would be a real resource for engineers and designers of resilient systems.

The system should allow users to move seamlessly between the typical instances of general concepts (people, projects, publications, research areas) and other concepts in the application domain, such as system components and their resilience characteristics, and educational resources provided by the network members, while automatically identifying resources related to those being viewed.

We start by detailing the most obvious aspect of this system: the RKB Explorer interface, before discussing the information sources that contribute to the system, and presenting an overview of the underlying infrastructure. We finish by showing some of the other applications, identifying future directions, and discussing some conclusions.

2 The RKB Explorer

The figure above shows the single window interface of the faceted browser available at <http://www.rkbexplorer.com/explore/>. More detail is available via ‘Help’.

The main pane shows a chosen concept, along with links to other concepts of the same type that the system has identified as being related. In the screenshot, the ReSIST Project itself is under consideration, with its details on the right, and related projects are shown around it. These are chosen according to the relative weight given to ontological relationships, and the number of those relationships to each concept. The weight of the lines gives a visual ranking. They represent a project ‘Community of Practice’ (CoP) for the project. Clicking on a resource will show the detail for it, while double-clicking will add the CoP for the new resource to the pane. This will then allow a user to see how different projects are related, and see the projects that provide linkage between them.

The panes in the lower half of the display show the related people, research areas, publications and projects, identified by similar ontologically informed algorithms, and are ranked by decreasing relevance. Thus the lower right-hand

pane gives a list of the related projects found in the main pane, while the lower left-hand pane shows those people involved in the currently selected project.

Clicking on the concepts listed in these panes causes the focus of the RKB Explorer to change to that concept, and thus a user could quickly move to a view of the paper ‘Fault Injection for Dependability Validation’, and see the related papers in the main pane and related concepts below. Alternatively, selecting ‘Algirdas Avizienis’ causes a similar change of focus to things related to him.

As with the graphical CoP above, the relevance presented by the system for each type of concept will depend on the structure of the ontology for each type, and the weights that have been chosen for each relation, which are changing as we refine the system. So, for example for projects, common investigators would be considered of particular importance; common associated papers of some importance; and funded under the same initiative of less importance.

3 Information Sources

Since ReSIST is concerned with Resilient Systems, it might seem that the sources targeted should be closely focused on those considered directly relevant. However, we believe this would be a mistake. The intended power of the system was that even experienced users would find information they were not expecting. Thus simply gathering information from the obvious places would only mean the system delivered information that was expected, hence we cast the net wider.

Another issue is to do with the confidence that the system would deliver an ‘even-handed’ view of the world, and that users would be able to have some confidence in this, or at least have a clear understanding of the ways in which it was angled. As a European project, it would have been inappropriate, for example, to include large-scale and detailed information from some national funding bodies, while excluding others that were not easily available. While we have detailed UK specific data, we have chosen not to use it here.

Ideally, in an active Semantic Web world, we would have simply been able to use existing knowledge sources. These sources would publish their contents against well-known ontologies, both as SPARQL endpoints and resolvable URIs. We would then use them, possibly needing some ontology translation on the way. Unfortunately, this is not yet the case. When the project started in January 2006, there were few such citizens of the Semantic Web, and so we resolved to undertake the bootstrap process ourselves.

We therefore harvested the information from the places we identified, and have made it openly available as both resolvable URIs and SPARQL endpoints, against our (AKT) ontology [4]. Each data source is held in a separate triplestore, not necessarily running on the same machine. In due course we hope that information providers can take over the ownership of their Semantic Web site.

For a system such as this, it is important that the information it provides is completely up to date. We have managed to achieve this for a small number of sources, but see the general problem being solved by the information providers taking ownership of the task of publishing the knowledge themselves.

We have gathered information from a number of different types of resource, using techniques which seemed most appropriate for each. Some were simply by web-scraping; others such as DBLP provide an XML dump which can be processed easily into RDF. Yet others provided CSV files, for which we have created simple tools to process into RDF, but also it is pleasing to report that some of the project partners were able to run these tools at their sites and upload the appropriate RDF directly to us.

Publications are at the heart of much research. We therefore looked to harvest from a number of major metadata resources. We chose the major publishers and aggregators in Computer Science, and have to date harvested some 37 million triples from Citeseer, the ACM, DBLP, and selected IEEE conferences.

Projects information comes from two major sources, totalling some 14 million triples. As a pan-European project, the information from CORDIS was mandatory. In response to developments in EU-US relations in Resilient systems, we have added all the project data from the NSF. We look forward to adding data from other large funding bodies.

Partner submitted data comes through a commitment from ReSIST members to provide information about their personnel, activities, and publications as best they can. Their ability to do this has varied with the IT systems at each site, and has led to the creation of simple tools to map data from a variety of databases or CSV exports into RDF, and to perform conversions from other formats such as BibTeX. There are more than 15 triplestores for our partners, which currently contain approximately 0.5 million triples.

Domain-specific data acquisition is important to enable experts in providing information about themselves and details of fields in which they work. We have provided bespoke form-based interfaces for acquiring specific details against ontologies that prescribe courseware materials and Resilience-Explicit computing mechanisms. In addition, we support general project activities within a wiki which utilises an early version of the Semantic MediaWiki [5] augmented with bespoke modifications to facilitate integration with our repositories.

Additional metadata has been acquired from the RISKS Digest [6], a resource with particular significance to project members, along with general location data detailing UN Location Codes such that we can easily render information utilising geographic tools such as Google Maps. Such maps have been provided to show information about the partners, as well as a visualisation of the courseware in the knowledge base.

4 Related Applications

As discussed above, the activity described here is only part of a greater vision of a knowledge-enabled infrastructure for the design, construction and deployment of resilient systems, in the context of an EU Network of Excellence. Unfortunately, space precludes the detailing of the full extent to which knowledge technologies have permeated every aspect of the project. However, we have already mentioned the use of a range of bespoke acquisition and information publishing interfaces,

combined with the integrated Semantic Wiki, and we would be happy to discuss and demonstrate these facilities at ISWC 2007. More details can be found in the appropriate project deliverable [7].

5 Information Infrastructure

5.1 Triplestores

We use 3Store [8] as our base repository, with a separate knowledge base representing each data source which we have acquired. The separate knowledge bases facilitate the system scalability, and help to provide the high query performance needed for an application of this sort, while allowing the assertion of the volumes of data (many tens of millions of triples) that we have. Each repository is complemented by a number of services and interfaces, which are accessible at <http://<repository>.rkbexplorer.com/>. These primarily include a SPARQL endpoint, direct access to RDF data through resolvable URIs, a tabulated triple browser for navigating the raw information contained, and a CRS (see below) for the triplestore, if appropriate.

5.2 Consistent Reference Service (CRS)

The way in which the RKB explorer and other applications give a unified view of tens of triplestores (knowledge bases) with tens of millions of triples, requires a well-founded method of allowing URIs to bridge between the triplestores, when they are considered to refer to the same concept.

The ReSIST activity embraces this. It includes in its architecture the deployment of a number of CRSes, which are knowledge bases of URI equivalences for the application being considered, according to appropriate criteria.

We chose to keep this knowledge separately from the main data. One reason is simply that of good engineering practice. It is easier to maintain knowledge that is being created by the CRS builder from the knowledge that is being created by the information provider. Indeed, different CRS providers will exist for the same information in an open Semantic Web world. A second reason is that a CRS is designed for a purpose, or set of purposes. Some applications might wish to consider that two concepts are the same, while this may not be the case for another application over the same knowledge. For example, in undertaking citation analysis, a paper with the same title and text that appeared both as a journal article and technical report should be considered as two separate papers. In an application such as ours, where we are considering who works with whom on what topic, it might well be more appropriate to consider that they should be treated as one resource, while still representing the separate details in a consistent fashion. As information providers of the basic information, we include a `coref:hasCRS` to the associated CRS in the RDF for a resource, so that it can be easily found, although there can be more than one CRS, corresponding to different policies.

Thus, the CRS is essentially an open service, which gives a view of URI equivalence: when presented with a URI, it returns all the URIs it considers equivalent. Note that it aims to avoid the mistake of creating a new URI; such an action would simply add further to the problem by being a new authority. It was also decided not to use `owl:sameAs`, since this is a much stronger assertion than the CRS is making. Of course, the knowledge bases themselves may still be using it where appropriate.

Allied to this CRS is the question of how to glean knowledge to put into it. Early in the activity it became apparent that the quality of this had to be high, but most importantly conservative. The idea of the whole application is to benefit from the ‘network effect’ of all the sources being viewed as one. Unfortunately, this also means that any mistakes in URI equivalence can give distorted views.

To generate knowledge for the CRS, the system uses the expected heuristic of string similarity (very conservatively), confirming the identification with the other relational uses, such as publication place (for papers), funding body (projects) and place of work (people), where these have already been the subject of equivalence identification. This means that to begin the generation of knowledge for the CRS, there is a ‘cold start’ problem, as there are almost no real URIs that are in common. This is achieved by string analysis of the titles of publication, and hence spreading to authors.

A novel aspect of this approach is that the system dynamically applies these heuristics. As a user browses, the CoPs that are calculated are essentially the knowledge required to decide if two URIs are the same: if the CoPs of two URIs with lexically similar associated titles, names, etc are similar, then we assume that they should be considered equivalent. URIs that are related are also queued for priority analysis. Thus, as users browse the system, the CRS improves, and the CRS knowledge of the resources in that topic area also improve.

Without such information of high-quality the system would provide little more than a view of disparate databases, and would in effect just be a browser for RDF. However, with this knowledge, effective information can be provided which totals more than the sum of the parts, using the meaning of the data to provide a single, consistent view over a significant number of completely separate knowledge bases.

5.3 The RKB Explorer

For the user interface for exploring the knowledge bases, as has been seen, we chose a simple and very static presentation, with one window. This was primarily because a previous attempt based on a more dynamic style, for example allowing a choice of topics, was criticised as being non-intuitive by many of our users, few of whom had any knowledge of Semantic Web technologies.

Since the general problem of distributed queries remains unsolved, the system has to implement querying as appropriate for its environment. To gather all the RDF related to a particular URI, it functions as follows.

Firstly it resolves the URI (which includes any `owl:sameAs` in that store). It then looks up the URI in the associated CRS, which can be identified from the

`coref:hasCRS` that was provided, and finds other, equivalent URIs. These can now be looked up in their CRSes, and the process continues until a complete set is found. There is also the provision for other CRSes which are not directly associated with information sources to be consulted. URIs can then be resolved directly, or if available accessed via a SPARQL endpoint for that domain.

It is also possible to consult all CRSes, but we consider this unnecessary. The CRSes we choose to trust for equivalence in these applications are either the original information providers, or ones we have chosen ourselves.

5.4 Caches

In order to achieve acceptable performance, the system has been carefully engineered. The use of 3Store provides high performance for queries, but it is necessary to take further steps. Caching is the standard way of doing this, and there are a number of caches in the system.

Firstly, URI resolution is cached. Rather than pre-generating and re-generating the RDF for every URI, incoming URIs are trapped with a dynamic 404 script, and the equivalent of a *construct* query is executed on the appropriate triplestore. As well as being returned, the results of this are then placed in a file at the URI, so that any future requests will simply be serviced by the web server.

Secondly, when the only possible action for a URI is to resolve externally, the resultant RDF is cached locally.

Finally, the calculation of a CoP represents a serious amount of RDF access and computation. The CoP results are therefore cached as they are generated.

6 The Future

In collaboration with IAI at the University of Saarbrücken, we are developing NLP tools targeted at identifying topics in resilient systems to improve the metadata describing the content of documents and to identify related clusters.

We are continuing to add data sets and knowledge. The project's Semantic Wiki is changing as the project progresses, and this is always reflected in the appropriate KB. As discussed, we are deploying interfaces for acquiring detailed metadata regarding courses and resilient mechanisms, which will enable the development of applications which can use this knowledge to advise or inform systems at run-time.

One of the challenges facing us now is the next step after the bootstrap process. We need to move the knowledge bases and infrastructure to the information providers; this is both the 'correct' thing to do in the Semantic Web, but also will mean that the information is better-maintained.

Another challenge is to start to use information that is beginning to come available, especially with the recent activity on Linked Data [9]. Because our applications are sensitive to the ontologies, this will mean the introduction of components to provide ontology mapping or dynamic translation. We are already experimenting in this area with one information provider.

Although the application presented here is situated in a particular application domain, it clearly has more general applicability. We are also working with other users to apply the system to alternative subject domains.

Finally, having created this infrastructure, we look forward to others using the knowledge bases we have curated to build exciting Semantic Web applications.

7 Conclusions

We have presented a real-world Semantic Web application that is based on large-scale information from independent sources, using an ontology to mediate between them and rank the resources when presenting consolidated results to users.

It provides a number of related applications, including the RKB Explorer, which gives an accessible and functional user interface. This, along with the usefulness of the knowledge resources have been extensively validated by the ReSIST Project partners, as reported in [7].

Since the system does not function by harvesting information into a common store, it is thus truly web-based. By employing resolvable URIs and distributed repositories to which queries can be fielded, we have created a real-world and scalable solution.

Acknowledgements

Many people have contributed directly and indirectly to this work over a number of years, including many members of the AKT and ReSIST projects. We thank them all, and in particular Harith Alani, Les Carr, Ben Dowling, Nick Gibbins, Steve Harris, Afraz Jaffri, Tim Lewy, Duncan McCrae-Spencer, Brian Randell, Benedicto Rodriguez, Nigel Shadbolt and Mikael Suominen.

This work is supported under the ReSIST Network of Excellence, which is sponsored by the Information Society Technology (IST) priority in the EU Sixth Framework Programme (FP6) under contract number IST 4 026764 NOE.

References

- [1] <http://www.aktors.org/>
- [2] Shadbolt, N., Gibbins, N., Glaser, H., Harris, S., schraefel, m.: CS AKTive Space, or how we learned to stop worrying and love the Semantic Web. *Intelligent Systems* **19**(3) (2004) 41–47
- [3] <http://cs.aktivespace.org/>
- [4] <http://www.aktors.org/publications/ontology/>
- [5] http://ontoworld.org/wiki/Semantic_MediaWiki
- [6] <http://catless.ncl.ac.uk/risks/>
- [7] Glaser, H., Millard, I.C., Anderson, T., Randell, B.: ReSIST Project Deliverable D10: Prototype knowledge base. Tech. Rept., University of Southampton. (2007)
- [8] Harris, S., Gibbins, N.: 3Store: Efficient bulk RDF storage. In: *Proceedings of the 1st International Workshop on Practical and Scalable Semantic Systems*. (2003)
- [9] <http://linkeddata.org/>