Triple Space Computing: A framework for the new communication paradigm

Reto Krummenacher, Francisco Martin-Recuerda
DERI, University of Innsbruck, Austria
firstname.lastname@deri.org

Martin Murth, Johannes Riemer
Vienna University of Technology, Austria
mm | jr@complang.tuwien.ac.at

Ying Ding, Dieter Fensel
Electronic Web Service GmbH, Innsbruck, Austria

ABSTRACT
Triple Space Computing (TSC) is an emerging communication and coordination paradigm tailored to the Semantic Web and Semantic Web services. In this paper we shortly describe the core ideas behind TSC and the initial efforts of an Austrian research project that aims at realizing the vision of Triple Space Computing.

Keywords
Triple Space Computing, Semantic Web services, service interaction and coordination

1. INTRODUCTION
Semantic Web services promise seamless interoperability of data and applications on a semantic level, thus turning the Web from a world-wide information repository for human consumption only to an infrastructure of distributed computation. Appropriate semantic descriptions of Web services and intelligent mechanisms working upon this, need a solid basement in terms of the underlying semantically enabled communication technologies. Triple Space Computing (TSC) which inherits the publication-based communication model from Tuple Space computing, extending it with semantics, provides solutions in that direction [2]. Instead of sending messages back and forth, applications will communicate by writing and reading RDF triples in a shared persistent information space, the Triple Space.

The TSC project is a research project aiming at developing such a new middleware infrastructure with special support for the Semantic Web and Semantic Web services. TSC is an Austrian national funded project running for 3 years. Currently it is at its early stage. The outcome of the TSC project will be a generic framework and prototype implementation for a Triple Space Computing environment. In this paper we report on the initial ideas of the framework and future plans.

2. TSC CONCEPTUAL MODEL
The TSC framework is based on the evolution and integration of several well-known technologies: Tuple Space computing [3], shared object space [7] and Semantic Web technologies (in particular RDF). It defines the use and extensions of these technologies with regards to their conceptual foundations towards Triple Space Computing.

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

The first foundation of the TSC framework is Tuple Space computing. The Tuple Space model not only decouples the information exchange in reference, time and space, but also offers a high-level abstraction, namely the communication via the reading and writing of tuples in a space, where a tuple is an ordered set of typed fields. Applying this paradigm offers the advantage of removing the complexity of message-based systems currently used for building Web services. It also offers advantages in terms of reduced development costs, simplicity, extensibility, easy debugging at runtime, and recovery due to persistent storage in the information space.

The second foundation is the application of Web design principles, thus further decreasing the deficiencies of message-based communication and hence improving the scalability of the system. The Web technologies add some additional features that are lacking in current Tuple Spaces: (1) URIs as a unique, well-defined reference mechanism, (2) namespaces as a separation mechanism of information chunks by qualified names, and (3) interlinking of resources by use of foreign URIs as hyperlinks.

The third foundation is the semantics of communication. The semantic descriptions in TSC are based on RDF triples and handled as Named Graphs [1]. Although RDF lacks expressivity for more complex ontology specifications, the triple model [5] provides a simple, but valuable approach for annotating information. Thus, it is considered to be sufficient for prototyping TSC. In more advanced implementations, richer data models than nested triples may be applied — e.g., OWL or rule languages.

Like Tuple Space computing, the basic Triple Space paradigm based on 'persistent publish and read' has a major limitation from the perspective of a client: an application which wants to read a concrete triple or set of triples has to interrupt the main process flow or run a concurrent thread that periodically checks if relevant data is available. Figure 1 illustrates an example of a simple producer-consumer interaction: Process B is the consumer and searches data before it is available in the space. Process A publishes the data. On the left side, process B queries the space (and blocks the main flow) until data is available. On the right side, process

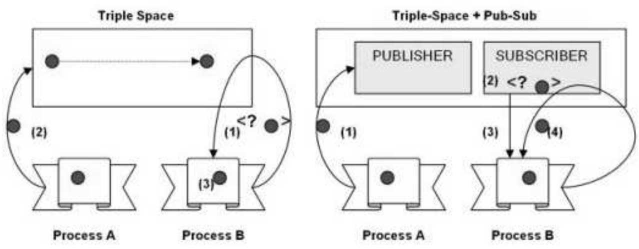

Figure 1: Publish-Subscribe over Triple Spaces.

B is subscribed to the data, and whenever some matching triples are available a notification is sent that indicates that the desired data can be retrieved.

The combination of the publish-subscribe paradigm and TSC defines a more flexible coordination model that extends the idea of a Web-like 'persistent publish and read' communication infrastructure for Web services to an even more decoupled 'persistent publish and subscribe' model.

3. TSC ARCHITECTURE

The envisioned TSC architecture is based on a hybrid architecture that combines P2P and client/server models in a so-called super-peer system [8]. This configuration ends up in a two-tiered system: the upper-tier is composed of a powerful and static server network, while the lower-tier consists of clients that might only be temporarily available and possibly possess limited computational resources (called heavy clients). Embedded devices like smart phones or PDAs, referred to as light clients, will access the Triple Space remotely using the backbone of servers (Figure 2). In that way the TSC framework is not only envisioned to improve service-oriented architectures, but also mobile and even ubiquitous applications [6].

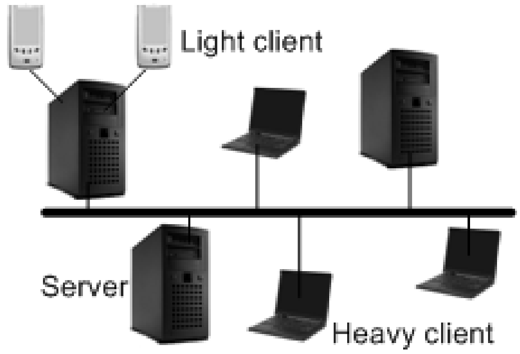

Figure 2: TSC architecture overview.

Heavy clients and servers run a Triple Space kernel (light clients do not) that provides coordination, security and data handling (data mediation, querying and storage) services. The CORSO (Coordinated Shared Objects) framework serves as the starting point to build the TSC kernel. CORSO is a platform for the management of distributed applications in heterogeneous IT environments based on communication via shared objects [7]. The CORSO implementation will thus be extended to support the mediation, replication and communication of semantic data. Yet Another RDF Store (YARS, [4]) is the RDF repository and query engine used by the kernels of the planned prototype to store, manipulate and query the data that each space contains. The idea is however to abstract the coordination service from the underlying data storage infrastructure and to allow the use of arbitrary data stores without altering the core implementation of the TSC kernels.

4. CONCLUSIONS AND FUTURE PLAN

In this paper, we presented initial ideas of a framework for Triple Space Computing, a new communication and coordination infrastructure for the Semantic Web and Semantic Web services. The project responsible for this work still has two years to go. During these two years, we will provide a consolidated TSC architecture and interfaces for the cooperation amongst service components and for the TSC infrastructure as a whole. In particular we will focus on the use of data mediation and query engine components, on data replication, security and privacy mechanisms and investigate how standard service architectures (SOA) can be better applied in TSC. In the end, a running prototype will be provided and the usability will be tested via a case study on how TSC can enhance communication and process coordination of a Semantic Web service execution environment like WSMX (www.wsmx.org).

5. ACKNOWLEDGMENTS

The authors thank the TSC project consortium. The project is funded by the Austrian Federal Ministry under the FIT-IT project grant 809249.

6. REFERENCES

[1] J.J. Carroll, Ch. Bizer, P. Hayes and P. Stickler. *Named Graphs.* Journal of Web Semantics 3(4), 2005.

[2] D. Fensel. *Triple-based Computing.* DERI Research Report: DERI-TR-2004-05-31, May 2004.

[3] D. Gelernter. *Generative Communication in Linda.* ACM Transactions on Prog. Lang. and Systems, 1985.

[4] A. Harth, M. Magni and St. Decker. *Scalable Distributed RDF Storage Infrastructure.* DERI Lion Deliverable 1.02, June 2005.

[5] G. Klyne and J. J. Carroll (eds.). *Resource Description Framework (RDF): Concepts and Abstract Syntax.* W3C Recommendation, February 2004.

[6] R. Krummenacher and Th. Strang. *Ubiquitous Semantic Spaces.* Conf. Supplement of the 7th Int'l Conf. on Ubiquitous Computing, September 2005.

[7] E. Kühn. *Virtual Shared Memory for Distributed Architectures.* Nova Science Publisher, 2001.

[8] B. Yang and H. Garcia-Molina. *Designing a Super-peer Network.* IEEE Int'l Conf. on Data Engineering, 2003.