Enabling the European Patient Summary Through Triplespaces

Reto Krummenacher
University of Innsbruck (DERI), Austria
reto.krummenacher@deri.org

Elena Simperl, Lyndon J. B. Nixon
Freie Universität Berlin, Germany
simperl|nixon@inf.fu-berlin.de

Dario Cerizza, Emanuele Della Valle
CEFRIEL - Politecnico di Milano, Italy
cerizza|dellavalle@cefriel.it

Abstract

One of the main items on the eHealth agenda of the European Community is the design and promotion of electronic patient summaries as an instrument to facilitate the pervasive delivery of healthcare, thus ensuring the right to patient mobility and increasing the productivity and quality of health service delivery. From a technical point of view this objective requires middleware technology which is able to cope with the stringent interoperability, multi-lingualism, security and privacy requirements arising in eHealth settings. In this paper we present triplespace computing, a coordination middleware for the Semantic Web and demonstrate its relevance to the realization of the European patient summary infrastructure.[1]

1 Introduction

The term eHealth circumscribes an inventory of technologies and tools to access, process and store information associated to healthcare related activities, from administration, scheduling and billing to clinical treatments and surgical procedures. In conjunction with the rapid advances in IT, several eHealth applications are already widely used in major healthcare institutions with attested efficiency gains. Despite this positive trend, healthcare delivery remains highly fragmented and it is difficult to integrate the various types of IT platforms. The most representative example in this respect is probably the access to patient information. The clinical information about an individual is created, processed and stored in different systems spread across several institutions. It cannot be accessed and used instantly by healthcare personnel, thus leading to additional costs – for obtaining the remotely available information or replicating particular procedures – and to a deterioration of the perceived quality of service. In order to increase the efficiency of patient care delivery, healthcare parties must be able to access and exchange patient information independently of their organizational and technological particularities. The European Commission is performing a first step in this direction by defining guidelines for the *European patient summary* (EPS) [1], an interoperability infrastructure for exchanging primary clinical information across European healthcare networks. From a technical perspective the realization of the EPS demands powerful middleware technology that guarantees ubiquitous access to distributed and multi-faceted data as well as scalability, persistency and interoperability.

Section 2 gives an overview of the EPS and specifies the requirements for an infrastructure. By help of a state of the art analysis in Section 3 we note that current (middleware) technology cannot

[1]This work is supported by the TripCom (IST-4-027324-STP) project (`www.tripcom.org`).

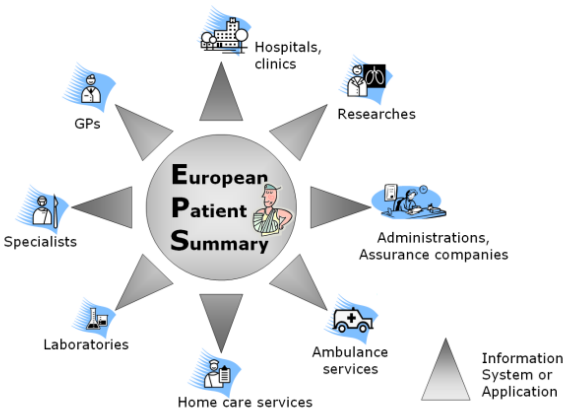

Figure 1. EPS Building Blocks

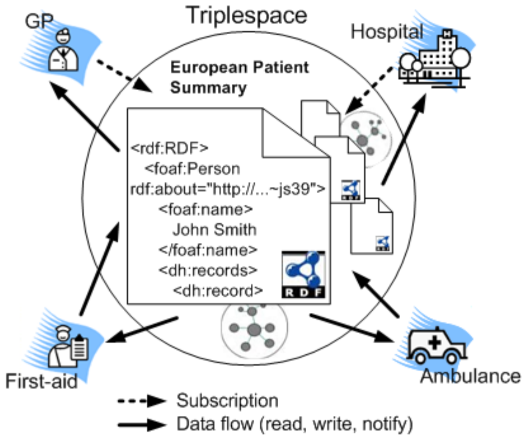

Figure 2. EPS with Triplespace

adequately support the outlined scenario. We hence propose a solution based on the combination of coordination systems and the Semantic Web - *triplespace computing* - in Section 4 and demonstrate its relevance to the EPS scenario in Section 5. Related and future work are considered in Section 6.

2 Requirements for an EPS Infrastructure

The European patient summary is a strategic initiative by the European Commission towards the realization of a European eHealth infrastructure capable of integrating information and applications in order to ensure the pervasive delivery of high quality care services. The EPS initiative should provide the core building blocks required to access and process primary clinical data in arbitrary eHealth applications across the European healthcare delivery network (Figure 1).

The technical realization of the patient summary needs to build on a specialized middleware which is able to deal with the design principles specified in the European Interoperability Framework (EIF, [2]): multi-laterality, subsidiarity, multi-lingualism, and privacy. We identify several requirements for an efficient EPS infrastructure:

◇ **decentralization/distribution** is a pre-requisite for the realization of **multi-laterality** and **subsidiarity**. A highly distributed EPS infrastructure allows arbitrary healthcare parties to publish and retrieve patient information efficiently and ensures a feasible level of fault-tolerance.

◇ support for **asynchronous** and **anonymous interaction** among institutions is equally important. The coordination of information should happen independently of communication partners.

◇ to cope with the inherent heterogeneity problems (e.g. different encoding schemes and languages) and to align different eHealth systems the middleware should provide means for flexible **data and application integration**.

◇ support for appropriate **security** mechanisms is important with respect to **multi-laterality** and **privacy**.

3 Middleware Technologies Today

Middleware aims at abstracting from client and storage implementations and provides a common access structure regardless of the underlying physical and logical characteristics. Middleware systems today tend to incorporate a wide spread of additional functionalities from which applications can benefit, as they do not have to be implemented again – e.g. *security measures*, *mediation* (of data and process heterogeneity) and *coordination* of concurrent interactions. Common middleware solutions are Remote Procedure Call (RPC), object brokers such as CORBA, message-oriented middleware, as well as (Semantic) Web services. Web services are seen to be a promising and currently widely researched approach in particular for large scale systems, due to its relationship to the World Wide Web. However, Semantic Web service infrastructures do not provide support for persistent publication of data, nor the time-decoupling of messages so that data can outlive the services publishing or consuming it.

The Linda tuplespace model [8] supports these features and has been mentioned as attractive for programming open distributed applications [15]. Linda has its origins in parallel computing and was developed as a means to inject the capability of concurrent programming into sequential programming languages. It consists of coordination primitives and a shared data space (*tuplespace*) which contains data (*tuples*). A tuple is an ordered list of typed fields. The coordination primitives are a small yet elegant set of operations that permit agents to emit a tuple into the space (operation *out*) or associatively retrieve tuples either removing them (*in*) or not (*rd*). Retrieval is governed by matching rules. Tuples are matched against a template, which is a tuple that contains both literals and typed variables. A match occurs if the template and the tuple have the same length and field types and if the value of literal fields are identical. The tuple (”N70241”,EUR,22.14) will match the template (”N70241”,?currency,?amount) and bind the variables currency and amount to the values EUR and 22.14 respectively. The operations are blocking and thus provide an inherited coordination mechanism. In summary a tuplespace: a) supports **asynchrony** and **concurrency**; i.e. the producers and consumers of a tuple do not need to know one another's address nor exist concurrently; b) permits **associative addressing**, i.e. data is retrieved on the basis of the type of data requested, rather than on which specific data is referenced; and c) **separates** the coordination **model** from characteristics of the host **implementation** environment.

Hence tuplespace systems may better meet the decentralization and distribution needs of the eHealth scenario. While a number of implemented tuplespace systems - e.g. JavaSpaces [7], TSpaces [16], or GigaSpaces [9] - clearly address the communication and coordination requirements of the outlined eHealth scenario, none of them explicitly copes with the intrinsic heterogeneity of encoding systems and message types. In this context semantic technologies have become increasingly popular in the last years. Including support for the management of semantic data in the aforementioned releases would however prove difficult as they are not open architectures. They are built upon a data model (the object model of Java) which is not directly compliant with the Resource Description Framework (RDF, [10]) model underlying the Semantic Web.

The Semantic Web and eHealth systems have been identified as being complementary [3]. In particular the latter are expected to take benefit from adding semantics to unambiguously define the meaning of medical concepts, structure the patient records and mediate between heterogeneous resources.

4 Triplespace Computing

Most of the available IT applications – and in particular Web services – depend largely on synchronous communication links between information producers and consumers. Instead of following the "persistently publish and read" paradigm of the Web, traditional Web services establish tightly coupled communication cycles and send directed messages back and forth.

This motivated a new communication paradigm: Tuplespaces allow distributed sharing of information without requiring synchronous connections. The Semantic Web extends the Web with machine processable semantic data, allowing data exchange in heterogeneous application fields. Combining the two provides a communication paradigm with persistent and asynchronous dissemination of machine understandable data at its core [4], which is called *triplespace computing*: RDF triples provide a natural link from the Semantic Web and tuplespaces to triplespaces.

The communication and coordination space is tailored to the needs of machines and integrates optimized storage facilities for semantic data, mediation, and reasoning engines. Triplespace computing raises however new requirements on common tuplespace models: a) **a reference mechanism** to uniquely address resources in the global space; b) **a separation mechanism** to allow different applications to use the same terms distinguishably; and c) **a nesting mechanism** to reflect the interlinking and nesting of semantic data in graphs.

According to the TripCom project, the global information space is organized in a tree of non-overlapping spaces. This structuring ends in increased scalability by its natural grouping of related data and its users; e.g. a space per regional authority with subspaces per hospital. The semantic data is then published and read from a particular space rather than the global triplespace. The semantic data itself is represented by three fielded tuples that model the $<$subject,predicate, object$>$-triples of RDF. In RDF triples are grouped in graphs; this structure is taken over to triplespace computing. Hence, an agent does no longer have to read and write single tuples, but rather whole constructs of semantic data. The use of RDF graphs implies also a more expressive retrieval scheme than traditional template matching because triples cannot be distinguished from one another in terms of contents, as they have all the same number of fields and field types. We therefore need to extend the retrieval primitives to support RDF-specific query languages that reflect the semantics of the data.

Another noteworthy extension will be the incorporated public-subscribe mechanism. An agent may subscribe to particular information by use of semantic templates and is informed whenever relevant data is published in the space.

5 Realization of the EPS Scenario

As shown by our middleware discussion, current technologies do not cover some significant aspects related to interoperability, coordination and scalability in open, autonomous systems. In this section we propose a grounding of the EPS scenario into triplespace computing. Triplespace computing is envisioned to be well suited to the scenario, as it provides the basic requirements of the system: a common data store, support for multiple agents and coordination of their interactions, and decoupling in time and reference. The functionality of the EPS system is abstracted into external agents. This is not only a basis for modularization and hence supporting reusability and updatability of data, but also makes system knowledge directly available to any interested (and access-enabled) agent. Simple agent operations (reading and writing) are then standardized and supported by the triplespace without requiring any specific functionality to be executed from the EPS system. Patient data, as well as associated coding systems and message types, are represented using machine-understandable representation languages and formal ontologies. This permits automatic mediation between heterogeneous data formats and reasoning over the information of the system to infer new and likely crucial knowledge.

To illustrate how the patient summary integrates with triplespace computing we designed a hypothetical use case of an English citizen traveling to Italy: John Smith has booked an all-inclusive coach trip. On day three of their trip the group gets involved in a major accident on the highly frequented motor way Milano-Venezia. The bus overturns and many of the travelers are injured, some even severely. John Smith suffers an open fracture of the leg and shows symptoms of shock.

Due to the high number of victims the volunteer first aid corps and most ambulances of the region are sent for, which calls upon complex coordination work. All medical staff has access to the EPS through mobile devices. This allows them to instantly gain access to relevant information about their patients in order to provide the best possible treatment on the spot. Moreover, the patient summary permits the different units to collaboratively treat the victims and to synchronize activities.

Andrea Bonardi, the rescue worker that first finds John, searches for his clinical data in the EPS by use of a unique identifier – John's passport number – and requests his patient record (Figure 2). The request is transformed to a template-based read. Templates could be based on triple patterns, as done in this paper with Notation3: a "`.`" terminates a triple, a "`;`" introduces another property of the same subject, and the "`,`" symbol introduces another object with the same predicate and subject.

```
rd(?p id:passNr "F1234"; dh:records ?record.)   -> ?record = uhid:2625365
```

According to EC policies Andrea is allowed to read all necessary information about allergies, immunizations, currently prescribed medication and contagious diseases. The data protection laws restrict first aid workers and ambulance doctors from consulting further details of the patient summary. This is controlled in the space by appropriate Access Control Lists which can restrict access on the basis of predefined roles [6]. The information requested by Andrea is presented to him in Italian. The application running on his mobile device queries terms inside the EPS which follow standardized medical terminologies using the appropriate language setting and for which official translations exist.

Under normal conditions Andrea would provide John Smith with a dose of the analgesic morphine (N02AA01 in the Anatomical Therapeutic Chemical Classification System). According to John's record he however repeatedly showed allergic reactions to morphine and Andrea prefers to administer oxycodone (N02AA05).

```
rd(uhid:2625365 dh:allergicTo ?analgesic.)      -> ?analgesic = atc:N02AA01
out(uhid:2625365 dh:rcvdTreatment atc:N02AA05.)
```

Furthermore Andrea takes care of stopping the bleeding of John's broken leg. After logging all the treatments in the EPS, he asks a bystander to keep a firm eye on John until the ambulance picks him up. In the meantime Andrea takes care of other casualties. Shortly thereafter the ambulance doctor and her team assume John. From his latest EPS entry they notice the medication and treatment he already received and a description of the injury published by Andrea: Open fracture of shaft of femur (UMLS C0159836).[2]

```
rd(uhid:2625365 dh:hasInjury ?injury; dh:rcvdTreatment ?treatment.)
    -> ?injury = umls:C0159836, ?treatment = atc:N02AA05
```

6 Related and Future Work

A number of approaches are emerging in the field of semantic tuplespace computing. Besides the core commonality of coordinating the exchange of semantic data, generally RDF, in a tuplespace, these approaches differ in their aims and hence their conceptualizations and realizations. sTuples [11] is the first instance of a semantic tuplespace but is limited to exchanging OWL documents [13] in tuple fields.

Semantic Web Spaces [14] have been presented as a middleware for a Semantic Web of heterogeneous, distributed intelligent agents with a focus on the communication of Semantic Web clients.

[2]The Unified Medical Language System (UMLS) is an integrated collection of more than 100 medical terminologies and classification systems.

In parallel, the triplespace computing proposal in [4] has spawned a number of initiatives; besides the TripCom project. In the TSC project a first attempt is made to extend the initial proposal with a concrete architecture in which the coordination model is enriched with publish-subscribe capabilities and transactions [5]. cSpaces is an independent effort to extend the original triplespace ideas with more sophisticated features and to study their applicability in different scenarios apart from Web services [12]. The TripCom project is also inspired by [4] and differs from earlier work in that it concentrates on conceptualizing and implementing triplespaces from scratch – while previous efforts extended and combined existing systems – and is focused on supporting a number of vital communication scenarios such as this eHealth case.

This paper presents the usage of triplespace computing as Semantic Web middleware for realizing the emerging European patient summary. First, tuplespaces are a good alternative to common information management and interaction models, since they allow agents to publish and retrieve information in decoupled ways. Second, triplespaces with integrated semantic technology support (mediation, reasoning and formalizations) allow knowledge-driven data management which is expected to increase scalability and to improve the reliability and search efficiency – a core requirement of eHealth settings. Moreover the TripCom project emphasizes on the integration of security and privacy measures in order to ensure the trustworthiness of triplespaces and hence the EPS. However, the realization of semantically-enhanced tuplespaces means not only enabling semantic data to be represented in terms of tuples, but also the revision of the classic Linda model with respect to the meaning of its primitives, and with respect to distribution, dynamics and scalability issues. In particular the latter tasks are subject to ongoing and future work of the TripCom project.

References

[1] Commission of the EC. An action plan for a European e-Health area - COM(2004) 356, 2004.

[2] Commission of the EC. European Interoperability Framework for Pan-European eGovernment services, 2005.

[3] E. Della Valle, D. Cerizza, I. Celino, A. Dogac, G.B. Laleci, Y. Kabak, A. Okcan, O. Gulderen, T. Namli, and V. Bicer. eHealth Use Case - Semantic Web Services in eHealth. In R. Studer, St. Grimm, and A. Abecker, editors, *Semantic Web Services: Concepts, Technology and Applications*. Idea Group Publishers, USA, 2007.

[4] D. Fensel. Triple-Space Computing: Semantic Web Services Based on Persistent Publication of Information. In *INTELLCOMM*, pp. 43–53, 2004.

[5] D. Fensel, R. Krummenacher, O. Shafiq, E. Kuehn, J. Riemer, Y. Ding, and B. Draxler. TSC - Triple Space Computing. *e&i Elektrotechnik und Informationstechnik*, 124(1/2), 2007.

[6] D. Ferraiolo and R. Kuhn. Role-Based Access Control. In *15th NIST-NCSC Conf.*, pp. 554–563, 1992.

[7] E. Freeman, K. Arnold, and S. Hupfer. *JavaSpaces Principles, Patterns, and Practice*. The Jini Technology Series. Addison-Wesley Longman Ltd., 1999.

[8] D. Gelernter. Generative Communication in Linda. *ACM Transactions on Programming Languages and Systems*, 7(1):80–112, 1985.

[9] GigaSpaces(TM) Technologies Ltd. GigaSpaces Platform - White Papers. `http://www.gigaspaces.com/`.

[10] P. Hayes and B. McBride. RDF Semantics. W3C Recommendation, 2004.

[11] D. Khushraj, O. Lassila, and T.W. Finin. sTuples: Semantic Tuple Spaces. In *MobiQuitous*, pp. 268–277, 2004.

[12] F. Martín-Recuerda. Application Integration Using Conceptual Spaces (CSpaces). In *1st Asian Semantic Web Conf.*, pp. 300–306, 2006.

[13] D.L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. W3C Recommendation, 2004.

[14] L.J.B. Nixon, E. Paslaru Bontas Simperl, O. Antonenko, and R. Tolksdorf. Towards Semantic Tuplespace Computing: The Semantic Web Spaces System. In *22nd ACM Symposium on Applied Computing*, March 2007.

[15] D. Rossi, G. Cabri, and E. Denti. Tuple-based Technologies for Coordination. *Coordination of Internet Agents: Models, Technologies, and Applications*, pp. 83–109, 2001.

[16] P. Wyckoff, S. McLaughry, T. Lehman, and D. Ford. TSpaces. *IBM Systems Journal*, 37(3):454–474, 1998.