Enabling Collaborative eHealth Through Triplespace Computing

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

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

Elena Simperl, Reto Krummenacher
University of Innsbruck (DERI), Austria
elena.simperl|reto.krummenacher@deri.org

Abstract

The design and promotion of electronic patient summaries as an instrument to facilitate the pervasive delivery of healthcare is emerging as a key technology in eHealth solutions. From the technical point of view this requires powerful middleware systems supporting interoperability, multi-lingualism, security and patient privacy. In this paper we present a semantic coordination model and describe how it can be used to support pervasive access to electronic patient summaries.[1]

1 Introduction

eHealth relates to a wide range of healthcare related activities being supported by computer systems and communication networks. Despite increasing uptake in major healthcare institutions, healthcare delivery remains highly fragmented and it is difficult to integrate the various types of information and IT platforms. The most representative example in this respect is probably the access to patient information. Data about an individual is created, processed and stored in different systems spread across several healthcare institutions, often without interrelation. Hence it cannot be accessed, integrated and used instantly by healthcare professionals or administrative personnel, independent of where they are and how they wish to access it, thus leading to additional costs for locating or obtaining 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 independent of organizational and technological heterogeneities. The European Commission is performing a first step in this direction by defining guidelines for the *European Patient Summary* (EPS) [5]. We aim at providing this summary in the Semantic Web language RDF, which is based on a formal semantics [8], and hence supports data validation, integration and the inference of new knowledge through well established logical reasoning approaches.

The realization of the EPS demands a powerful coordinating middleware for exchanging primary clinical information across European healthcare networks that guarantees ubiquitous access to distributed and multi-faceted data objects with a focus on scalability, persistency and interoperability. *Triplespace computing* [6] is an emerging proposal for providing Web-scale data co-ordination for information formalized using Semantic Web representation languages such as RDF. To enable this, a Linda-based coordination model is specified for knowledge co-ordination, as well as new types of tuples and tuplespaces in order to handle interpreted information with assigned truth values.

In this paper, we introduce our specification for triplespace computing and outline its use in an European Patient Summary scenario. The rest of this paper is organized as follows. Section 2 describes our proposal of a coordination model for triplespace computing. We illustrate its application in the electronic patient summary scenario in Section 3. Related and future work are considered in Sections 4 and 5, respectively.

2 Triplespace Computing

Most of the available IT applications depend largely on synchronous communication links which tightly couple the communicating agents, requiring that agents know how to reach their communicating partner (direct addressing), that they are active at the same time (temporal dependency) and that if one communication partner loses their connection, the communication is broken and possibly lost (point-to-

[1]This work has been supported by the TRIPCOM (IST-4-027324) Specific Targeted Research Project.

point).

These flaws motivated the choice of a new communication paradigm: tuplespaces [7] are shared data stores which allow distributed sharing of information - stored in ordered lists known as ”tuples” - by various devices without requiring synchronous connections. Tuplespace co-ordination is enabled by a simple yet powerful co-ordination language, Linda [4]. The Semantic Web [1] extends the Web with machine processable semantic data, allowing data exchange in heterogeneous application fields. Combining the two introduces a new communication platform that provides persistent and asynchronous dissemination of machine understandable information. We call this combined communication platform *Triplespace*: semantic information encoded in RDF triples provide a natural link from the Semantic Web and tuplespaces to triplespace computing.

To realise triplespace computing, it is necessary to revisit the definition of tuples and tuplespaces, as it is indispensable to adapt these concepts to the norms of the Semantic Web. Thereafter we concentrate on the description of the required coordination primitives in order to make the Linda operations compatible to the requirements of the Semantic Web.

2.1 Triples and triplespaces

Following the Linda paradigm a triplespace system should be able to represent *semantic information* through *tuples*. The expressivity of the information representation should be aligned to the expressivity of common Semantic Web languages, while respecting their semantics, so that tuples could be mapped to and from external Semantic Web resources. Regarding Semantic Web languages, we currently focus on RDF. RDF statements can be represented in a three fielded tuple (so-called *“triples”*) of the form *<subject, predicate, object>*. Following the RDF abstract syntax, each tuple field contains an URI (or, in the case of the object also a literal).

RDF statements are grouped into RDF graphs (the RDF data model being graph-based). We note that a RDF graph can of course contain only one statement. Hence graphs are used as the main data structrue for communication, which implies a more expressive data model than in classical Linda.

A triplespace is defined as a container for triples which encapsulate the RDF statements. A triplespace can be divided into virtual subspaces and physically partitioned across distributed kernels. Every space is addressed using a URI and may contain multiple (sub-)spaces, while it can only be contained in at most one parent space. Consequently a tuple can be contained in a space - and implicitly in all the direct and indirect parent spaces of the original space. Communication with the triplespace can be restricted to a boundable part of the triplespace, i.e. a subspace, to allow for greater efficiency in interactions.

2.2 Coordination model

The original Linda operations, *out*, *in* and *rd*, form the basis for any tuplespace implementation. The basic tuplespace primitives have however soon proven to be insufficient in various application contexts, and implementations of tuplespace platforms based on Linda have liberally extended the coordination language for their needs. An outline of the Triplespace API is given in Table 1.

The coordination API extends Linda to support the reading and writing of sets of RDF triples (i.e., RDF graphs) instead of individual tuples. Hence the core retrieval operations, *in* and *rd*, while maintaining a version supporting ”traditional” Linda approach of returning the first matching tuple, also define versions for retrieving a RDF graph constructed from all found matches and for retrieving a RDF named graph which contains the first found match (named graphs are an extension of RDF to allow for the association of an URI with a set of RDF statements in order to identify them unambiguously).

The matching procedure for the retrieval operations is based on templates. The precise syntax and semantics of a template depends on the maturity of the space implementation and on the query languages and engines employed in this implementation; we however generally refer to it as template in order to proceed with a stable interaction model. In the current approach we use simple triple patterns that are very close to traditional Linda templates. A template is hence a tuple that contains both RDF resources and variables: *<x:TripleSpacePaper ?p ?o>*. In the future templates will consist of graph patterns (cf. Table 2) as common to most RDF query languages, e.g. SPARQL [14]. Eventually, we expect the templates to evolve to fully fledged semantic queries or rules, which invoke reasoning engines and operate on asserted and inferred triples.

Likewise, *out* can emit a RDF graph into the space, and optionally provide it with an identifier to make it into a named graph.

A noteworthy extension to the original Linda model is the incorporation of a publish-subscribe mechanism. An agent is able to subscribe to a particular type of information by providing a template and will be informed whenever some other agent placed data matching that template in the chosen space. This extends the expressiveness of the coordination model by providing a new type of interaction pattern which can not be supported by the original Linda primitives.

The last two operations depicted in Table 1 are management methods used to create and delete spaces. A space is created by giving it a new unique identifier and by possi-

out(Graph g, URI space, [URI graph, URI transaction]):void Inserts the triples included in the graph into the given space. By specifying a graph identifier a named graph is created. A transaction identifier can be provided to add the out to a given active transaction.	
rda(Template t, [URI space, URI transaction, integer timeout]):Graph Returns *one* matching triple with any triples bound to it, e.g. following the Concise Bounded Descriptions approach. The request is executed against the given space, if provided, otherwise against the virtual global triplespace. The timeout is used as means to control the blocking characteristics of rda.	
rd(Template t, [URI space, URI transaction, integer timeout]):Graph This operation generalizes rda: it returns an undetermined number of matching triples and their bound graphs; otherwise the functionality is the same.	
rdg(Template t, [URI space, URI transaction, integer timeout]):Graph Returns the entire content of a named graph that contains a matching triple; used to coordinate whole objects.	
ina(Template t, [URI space, URI transaction, integer timeout]):Graph This is the destructive version of rda.	
in(Template t, [URI space, URI transaction, integer timeout]):Graph This is the destructive version of rd.	
ing(Template t, [URI space, URI transaction, integer timeout]):Graph This is the destructive version of rdg.	
subscribe(Template t, URI space, Listener l, [URI transaction]):URI Establishes a notification mechanism for triples matching the template. Subscriptions must be expressed against a given space. In case of a match the listener (e.g., in Java a class that is called in case of an event) is notified. The operation returns a handle to the successfully registered subscription in form of a URI.	
unsubscribe(URI subscription, [URI transaction]):boolean This operation cancels a given subscription and returns true in case of successful execution.	
create(URI space, [URI parent, URI transaction]):boolean This primitive creates a new space, as subspace of parent. In case no parent space is indicated the new space is installed as direct child of the virtual global space. It returns true after successful creation.	
destroy(URI space, [URI transaction]):boolean This operation destroys the given space, its subspaces and all contained triples. Particular attention has therefore to be paid to rights management to avoid unauthorized removals.	

Table 1. The Triplespace API

TEMPLATE	DESCRIPTION
`?s a doap:Project; foaf:member ?o.`	Matches all triples where the subject is of type doap:Project and where the same subject has triples indicating the members.
`?s ?p ?o. ?o a foaf:Person.`	Matches all triples where the object is of type foaf:Person.
`?s foaf:name ?a; foaf:mbox ?b.`	Matches the triples that contain subjects for which the name and a mailbox (foaf:mbox) are indicated.

Table 2. Examples of Semantic Templates

bly attaching it as a subspace to an already existing space. The semantics of *destroy* is more complex, as the removal of a space implies the deletion of all subspaces, and of the contained data. We therefore expect that removal is only allowed to the creator of the space, or at least that it depends on restrictive security measures. Security, privacy and trust measures are entirely neglected in this paper, as they are seen to be orthogonal to the presented concepts and are developed in parallel.

The coordination model specified here has been validated in a prototypical implementation which is available on Sourceforge [2] and is continuing to be refined and extended in the EU project TripCom. A scenario to validate the triplespace computing approach in a real world setting is the eHealth scenario to which we now turn.

3 An European Patient Summary System

Several Health applications have been widely deployed in major healthcare institutions all over Europe. Anyway, healthcare delivery remains highly fragmented and it is difficult to integrate the various types of information and IT platforms. The clinical information about an individual is created, processed and stored in different systems spread across several healthcare institutions. 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. But this is a very challenging objective since it needs to deal with not only technical requirements but also with political, organizational and social aspects.

Patient summaries represent concise clinical documents that manage the most crucial information related to the

[2] `http://sourceforge.net/projects/tripcom/`

health status of citizens. For this reason they really represent a first step towards a network of complementary and heterogeneous healthcare systems.

The provision of a patient summary at European level is a strategic challenge of European Union because they represent also an enabling factor for the pervasive delivery of high-quality health-care services across Europe. But such goal demands very strongly requirements for a technological platform that can support the dimensionality of the scenario. In particular, there are about 800 millions of citizens within Europe and 1 million of care-givers. In addition, mobility of citizens must be guarantee and the access to their health information must be ensured anytime and everywhere. Moreover, a large variety of existing eHealth applications already exists and it's not possible to force those applications to adapt to a new European wide system. Indeed, it's necessary that such big system is enough flexible to adapt to the existing applications, dealing with heterogeneity issues both at level of the data exchanged and also at the level of application interfaces and protocols. Furthermore, the privacy of citizens must be respected, according to EU and specific country policies that guarantee that only authorized care givers will have access to sensitive data. Finally, healthcare authorities that are responsible of managing the data of the citizen needs to maintain their control over the data published.

An infrastructure that can meet all the previous requirements enables the provision of added-value services that are nowadays unbelievable. Just thinking to critical emergency situations where several actors operates in the same moment on several victims and there are strong needs for accessing health data and coordinating emergency operations in order to provide the most effective services.

Such very demanding requirements are met by the triplespace infrastructure and the following use case reports on a critical emergency situation where all the capabilities provided by the triplespace are used for effectively coordinating actors that pervasively access the EPS space through mobile devices.

Mr. Christian is an English citizen that is spending his holidays in Northern Italy and Austria. While traveling by bus along the highly frequented highway Modena-Brennero, he is suddenly getting involved in a major accident. The bus overturns near Bozen, in South Tyrol, and many of the travelers are injured, some even severely. Mr. Christian suffers an open fracture of the leg and shows symptoms of shock.

Due to the seriousness of the accident and 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 (also the nearby Italian-speaking region of Trentino is involved). All medical staff has access to the EPS through mobile devices. This allows them to in-

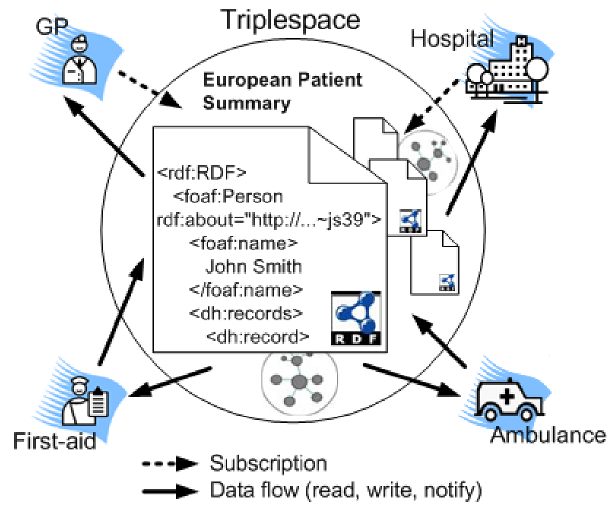

Figure 1. Realization of the EPS with Triplespace

stantly gain access to relevant information about their patients in order to provide the best possible treatment on the spot. Moreover, the EPS built over Triple Space permits the different units to collaboratively treat the victims and to synchronize their activities.

Rescuer Roman, the South-Tyrolese rescue worker that first finds Mr. Christian, searches for his clinical data in the EPS system using his PDA device. This can be done after having identified Christian in the system through some unique ID, for example the passport number which Christian is thankfully carrying on him.

```
rd(?foaf:Person, id:passNr, "1234...") ->
p:js39 rd(p:js39, dh:records,b ?dh:record) ->
uhid:026253645
```

The PDA of the rescuer has been easily integrated with the EPS since the triplespace levers over RDF language to provide semantic interoperability techniques that help in solving heterogeneity issues among existing eHealth applications and eHealth standards. Given the transmission of sensitive data in such a context, the system encrypts and decrypts data in its transmission between the user and the triplespace making use of a key that is only allocated to Roman's current interactions [2].

According to European, English and Italian policies, he 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. The information requested by Roman is presented to him in German as the application running on his mobile device can query terms inside the EPS which follow standardized medical terminologies using the appropriate language setting, as they are defined and official translations for their terms in all European languages exist.

```
rd(uhid:026253645, dh:diagnosedWith, ?dh:Contagion)
    -> icd:b24
rd(icd:b24, rdfs:label, ?string)
    -> "Sindrome da immunodeficienza acquisita"
```

Under normal conditions, Roman would provide Mr. Christian with a dose of the analgesic morphine (N02AA01 in the Anatomical Therapeutic Chemical Classification System). According to Christian's summary, he repeatedly showed allergic reactions to morphine and the first aid assistant prefers to administer oxycodone (N02AA05) that does not trigger the same consequences. Furthermore, Roman takes care of stopping the bleeding of Mr. Christian's broken leg.

```
rd(uhid:026253645, dh:allergicTo, ?inn:Analgesic)
    -> atc:N02AA01
out(uhid:026253645, dh:rcvdTreatment, atc:N02AA05)
```

After logging all the treatments administered to Mr. Christian and his GPS position in the EPS, he takes care of other casualties. Other rescuers and ambulance doctors can now read the information published by the rescuer, becoming aware that Mr. Christian requires further medical cares.

Only shortly thereafter Dr. Anna, an American ambulance doctor working in Trentino, and her team take over the care of Mr. Christian. From his latest EPS entry (now presented in English thanking to the terminology mediation), they notice the medication and treatment he already received and a description of the injury published by Rescuer Roman: Open wound in lower leg.

```
rd(uhid:026253645, dh:hasInjury, ?dh:Injury)
    -> icd:s81
rd(icd:s81, rdfs:label, ?string)
    -> "open wound in lower leg"
rd(uhid:026253645, dh:rcvdTreatment, ?dh:Medicine)
    -> atc:N02AA05
rd(atc:N02AA05, rdfs:label, ?string)
    -> "oxycodone"
rd(uhid:026253645, dh:rcvdTreatment, ?dh:Treatment)
    -> dh:CottonGauze
```

Dr. Anna decides that Mr. Christian needs to be hospitalized and she publishes a new emergency case in the EPS with the location of the accident. Local hospitals monitor the space using the publish/subscribe mechanism and are alerted of an emergency case in their vicinity. The ambulance crew then monitors the space for removals, as the hospital consumes the emergency call of Dr. Anna when they have the necessary capacities. Not only can no other hospital erroneously allocate resources to Mr. Christian, but also the ambulance crew can know which hospital they should bring their patient to.

```
ambulance:
out(uhid:026253645, dh:emergencyCase, loc:....)

hospital:
notify(?foaf:Person, dh:emergencycase, loc:....)
    -> uhid:026253645

hospital:
in(uhid:026253645, dh:emergencyCase, loc:....)
```

While Mr. Christian is on the way to the emergency room in Trento, Dr. Erica, the emergency doctor, can already take action to initiate the treatment at the local hospital. Dr. Erica and her team access the EPS of Mr. Christian and study the clinical information about his current health status (added by Dr. Anna and Rescuer Roman) and his past medical history. In that way they are ready to receive Mr. Christian and can treat him faster and more accurately.

Back home in England, the general practitioner Dr. Gabriela, that is responsible for Mr. Christian, will be informed automatically by the EPS through her eHR about the changed medical status of the patient (this is due to the subscription made by the GP on her cared citizens). This notification contains, in the English language of the general practitioner and in the correct format treated by her eHR, all the necessary information about the emergency recovery, the specific medical problems and the treatment received by Mr. Christian.

4 Related Work

A number of approaches are developing in the field of triplespace computing. A review of current activities has been carried out in the interests of identifying commonalities and differences [12]. 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 conceptualisations and realizations. sTuples has the first instance of a semantic tuplespace but it was limited to exchanging OWL (ontology) documents in tuple fields [9], just as other space implementations have supported the exchange of XML documents [15].

The Triple Space Computing proposal [6] has spawned a number of initiatives. In the TSC project a first attempt is made to extend the initial proposal with a concrete proposed architecture in which the coordination model is enriched with publish-subscribe capabilities and transactions [3]. cSpaces was born as an independent initiative to extend Triple Space Computing with more sophisticated features and to study their applicability in different scenarios apart from Web Services [11]. The work in this paper is also inspired from this proposal, and differs from earlier work in that it concentrates on conceptualizing triplespaces from scratch – while previous efforts extended existing systems

– and is focused on supporting a number of vital communication scenarios such as the eHealth case given here.

Parallel to this, Semantic Web Spaces [16] have been presented which aims to act as a communication middleware for a Semantic Web of heterogeneous, distributed intelligent agents. Its focus is on the communication of Semantic Web clients rather than Web Services. However we expect that both initiatives can learn from one another in dealing with the challenges of implementing semantic data exchange on top of the Linda coordination model.

The benefits of applying tuplespace technology to the healthcare sector has been acknowledged in [10]. The authors describe a tuplespace realized with the use of RFID technology and elaborate on its usage in eHealth scenarios. However, their work does not consider the use of semantic data nor an extension of the coordination model.

5 Conclusions and Future Work

In this paper we presented the usage of the tuplespace paradigm as Semantic Web middleware for realizing the emerging European patient summary. Tuplespaces are a good alternative to common information management and interaction models on the Web, since they allow agents to publish and retrieve information in an uncoupled manner in terms of space and time.

Current middleware technologies do not cover some significant aspects related to interoperability, coordination and scalability. We propose an extension of the electronic patient summary scenario into triplespace computing. The Linda model for coordination is suited to this scenario, as it provides the basic requirements of the system: a common data store, support for multiple agents and their interaction, coordination of that interaction and decoupling from time and space.

The functionality of the system is also abstracted into external agents who interact with the data store. This not only is a basis for modularizing the EPS system and hence supporting reusability and updatability, but also makes system knowledge directly available to any interested (and access enabled) agents. Simple agent operations (reading some knowledge from the system) are then standardized (through Linda) and supported from the tuplespace platform without requiring any specific functionality to be executed from the EPS system.

Patient data, as well as associated coding systems and exchange message types, are represented using machine-understandable representation languages (such as RDF(S)[8], OWL[13]) and formal ontologies. This permits automatic mediation between heterogeneous data formats and reasoning over the knowledge of the system to deduce new information which is of use to the healthcare providers.

Our scenario will, in the next step, be integrated with the first Triple Space prototype in order to demonstrate and validate the coordination of European Patient Summary data through triplespace computing. Furthermore, we will continue to specify how patient summaries are modelled and manipulated in RDF/OWL and the prototype will be extended to support richer querying and scalable distribution.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 5 2001.

[2] C. Bryce, M. Oriol, and J. Vitek. A Coordination Model for Agents Based on Secure Spaces. In P. Ciancarini and A. Wolf, editors, *Proc. 3rd Int. Conf. on Coordination Models and Languages*, volume 1594, pages 4–20, Amsterdam, Netherland, 1999. Springer-Verlag, Berlin.

[3] C. Bussler, E. Kilgarriff, R. Krummenacher, F. Martin-Recuerda, I. Toma, and B. Sapkota. D21. v0.1 WSMX Triple-Space Computing, June 2005.

[4] N. Carriero and D. Gelernter. Linda in context. *Commun. ACM*, 32(4):444–458, 1989.

[5] Commission of the European Communities. An action plan for a European e-Health area (COM (2004) 356 final), 2006.

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

[7] D. Gelernter. Generative Communication in Linda. *ACM Trans. Prog. Lang. Syst.*, 7(1):80–112, 1985.

[8] P. Hayes and B. McBride. RDF Semantics. Available at http://www.w3.org/TR/rdf-mt/, 2004.

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

[10] M. Mamei, R. Quaglieri, and F. Zambonelli. Making tuple spaces physical with rfid tags. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 434–439, New York, NY, USA, 2006. ACM Press.

[11] F. Martin-Recuerda. Towards cspaces: A new perspective for the semantic web. In *1st International IFIP WG 12.5 Working Conference on Industrial Applications of Semantic Web (IASW '05)*, Jyvaskyla, Finland, 2005. Springer IFIP Book Series.

[12] F. Martín-Recuerda, L. Nixon, and E. Paslaru Bontas. Knowledge Web Deliverable D2.4.8.1 Technical and ontological infrastructure for Triple Space Computing, January 2006.

[13] P. F. Patel-Schneider, P. Hayes, and I. Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. Available at http://www.w3.org/TR/owl-absyn/, 2004.

[14] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. W3C Working Draft, October 2006.

[15] R. Tolksdorf and D. Glaubitz. Coordinating Web-based Systems with Documents in XMLSpaces. In *Proceedings of the Sixth IFCIS International Conference on Cooperative Information Systems (CoopIS 2001)*, number LNCS 2172, pages 356–370. Springer Verlag, 2001.

[16] R. Tolksdorf, E. Paslaru-Bontas, and L. Nixon. A coordination model for the Semantic Web . In *Proceedings of the 21st ACM Symposium on Applied Computing, Track "Coordination Models, Languages and Applications"*, 2006.