Towards a Multimedia Content Marketplace Implementation Based on Triplespaces

David de Francisco[1], Lyndon JB Nixon[2], and Germán Toro del Valle[1]

[1] Telefónica Research and Development,
Valladolid, Spain
`{davidfr,gtv}@tid.es`
[2] Free University of Berlin
Berlin, Germany
`nixon@inf.fu-berlin.de`

Abstract. A Multimedia Content Marketplace can support innovative business models in the telecommunication sector. This marketplace has a strong need for semantics, co-ordination and a service-oriented architecture. Triple Space Computing is an emerging semantic co-ordination paradigm for Web services, for which the marketplace is an ideal implementation scenario. This paper introduces the developed Triple Space platform and our planned evaluation of its value to our telecommunication scenario.

Keywords: telecommunications, co-ordination, marketplace, semantics, triplespaces.

1 Introduction

The Internet, and more specifically Web technologies, have brought about new technological means to explore new business opportunities. In the telecommunications sector, this has been seen as a clear challenge to diversify market coverage of companies. This extended market coverage, as well as the technology adoption of economic models such as specialization and outsourcing (through the Service Oriented Computing paradigm [1]), has meant that companies have began to collaborate to offer more complex and attractive services to their customers (B2B). These collaborative scenarios have led to new technical challenges such as data and process integration of heterogeneous sources (EAI [2]). Therefore, means for transparent communication and integration of data and processes to tackle the inherent requirements of such scenarios have to be defined.

In a previous work [3], Digital Asset Management (DAM) [4] was presented as a strategic collaborative scenario for telecommunication companies. This work identified some key requirements for the communication infrastructure that could be summarized as: (1) arbitrary number of transparent parties involved, (2) the integration of heterogenous data and message formats between them, (3) persistency of the information, (4) reliable and secure access to the information, and last but not least and (5) the support of agile business transactions, where

the communication infrastructure could handle the information in an intelligent way, acting consequently.

The aforementioned requirements demand more flexible communication and coordination paradigms beyond the ones used today, such as CORBA or Web Services, characterized by their excessive strictness to the defined interfaces of each participant. One of most innovative communication and coordination paradigms existent today is the one proposed by Triple Space Computing[1] which efficiently combines the highly distributed coordination techniques of Triple Space Computing as a distributed shared memory [5] with the full power of Semantics [6]. Triple Space Computing exposes a simple and standard interface based on the Linda co-ordination language to every client, making communication flexible.

This paradigm has been chosen to explore its applicability to solve the inherent complexity of the DAM scenario, as well as to envision new services and capabilities to be offered to the potential users of this increasingly demanded area of the telecommunication industry. On the other hand, the DAM scenario is also proposed as a validation of the Triple Space Computing paradigm itself. The business scenario is significant as: (1) it acts as a proof of concept of the proposed paradigm, (2) it serves as a way to enable the business exploitation of collaborative scenarios and, last but not least, (3) an evaluation of the designed infrastructure can serve to engender new requirements.

With these objectives in mind, the paper starts outlining the Triple Space Computing paradigm in Section 2 and continues with a presentation of the DAM scenario proposed in Section 3. The definition of the scenario will make emphasis on the relevance to the telecommunication industry and the motivation of choosing Triplespaces as the main back-end technology. Section 4 details the design and implementation of the proposed scenario, whereas Section 5 stresses the evaluation plan to demonstrate the value of the paradigm for collaborative scenarios in the telecommunication sector, stressing the business significance of the evaluation through exploitation models. Conclusions and further work close the article in Section 6.

2   Triple Space Computing and Triplespaces

Triple Space Computing [7] [5] is a novel communication and co-ordination paradigm based on the publication of semantic information (RDF triples [8]). This paradigm combines Tuple Spaces using Linda coordination [9] with Semantic Web principles [6] to provide a time, location, reference and schema decoupling to the communication among any kind of applications, with special emphasis on Web services and semantic Web services [10]. The aforementioned de-coupling aims at the design of asynchronous and transparent communications which permit semantic systems which are more flexible, fault tolerant and unaware of heterogeneity.

[1] In this paper we focus on one particular realization of this paradigm as is the one developed in the context of the TripCom FP6 EU project (http://www.tripcom.org)
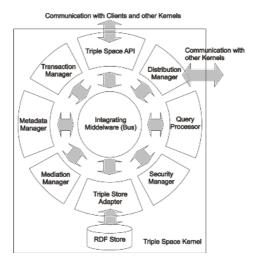
Fig. 1. Triple Space Kernel

Triple Space is the implementation of the Triple Space Computing paradigm. Triple Space is realised as distributed kernels which each host sets of spaces and which are connected by a combination of a Client-Server and a Peer-to-Peer network overlay infrastructure (the former used for direct kernel-to-kernel communication, the latter for network organisation and data look-up).

In Figure 1 the logical view of a single Triple Space kernel [11] with both kernel-internal components (circled around the integrating middleware) and kernel-external clients and other kernels (upper part of the figure) and services that may be connected to it is presented. The components that form a Triple Space kernel communicate over a kernel-internal bus system which is implemented using a tuplespace that allows all components to communicate with each other in a decoupled and co-ordinated manner.

Persistent storage of data at Triple Space kernels is provided through RDF stores which are connected to a kernel by the *Triple Store Adapter* component. The Triple Store adapter (i) abstracts from different RDF stores and their APIs and (ii) enables transparent distribution of data across a number of physical RDF stores. Apart from hosting triplespace data, kernel components can use the RDF store through the Triple Store adapter to persist configuration or runtime data. Security policies are enforced by the *Security Manager* that ensures that all data exchanges across kernel boundaries adhere to specified security policies. The *Metadata Manager* manages and provides access to a kernel's knowledge about itself (e.g. subspaces, triple access statistics) and about the global triplespace infrastructure (e.g. other kernels, clustering and routing information) in the form of semantic (RDF) metadata. The *Query Processor* is responsible for decomposing a query to parts that are satisfiable by the local data store and to parts that must be forwarded to other kernels in order to fulfil the query in its entirety, hence supporting distributed query answering in the Triple Space. The

Distribution Manager connects a single kernel to the global space infrastructure. It implements lookup functionality to find data in other spaces based on a distributed index storage system and carries out communication with remote kernels.

3   A Multimedia Content Marketplace Based on Triplespaces

This section describes the DAM contextualization, motivating why this business model is very promising for the telecommunication sector.A Multimedia Content Marketplace as a specific realization of the more general DAM scenario is presented, pointing out its functional description. Finally, the choice of the Triple Space Computing for its implementation is argued.

3.1   Contextualization

Multimedia content is a collaborative business in which several roles coexist, resulting in a very fragmented value chain (see Figure 2). *Content providers* are entities which own multimedia content (e.g. film producers). This content is usually not commercialized by owners, but by *content brokers* (e.g. film distributors). Content is exploited by *service providers* (e.g. a television company), which offer it to their *customers*. The content is delivered to customers by a *content distributor* (e.g. a telecommunication operator), which can not only deliver it, but also to provide some added value (e.g. QoS features). Content distribution often has to take into account Digital Rights Management (DRM [12]), which is provided by a *DRM provider*.
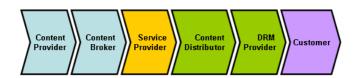

Fig. 2. Multimedia Content Value Chain

For a telecommunication company, such as Telefónica, multimedia content offers many business opportunities. The most obvious is the content distribution, where network and added-value features are already being provided by the company. Furthermore, service provision was handled by the company through the Imagenio product [2].

In the use case we will present below, we focus on the design of services based on multimedia content. The business context presented above is simplified for the business case definition. We consider a content provider as an entity which

---

[2] See http://www.telefonica.es/tol/imagenio.html (in Spanish)

offers multimedia content (being either producer or broker). A service provider is an entity which wants to offer services to their customers, making use of content offered, int he business case. Finally, a customer subscribes to these services and provide feedback about them. These roles will interact through a marketplace infrastructure provided by the marketplace owner. The next section describes the functionality our business case provides.

3.2   Functional Description

The business roles for the business case defined above will interact through a multimedia content marketplace. Telefónica, as a telecommunication operator, will play the marketplace owner role in this business case. Service and content providers will make businesses through the marketplace application. As a result, Telefónica will act as a business mediator among actors. In addition, customers can find, subscribe to and provide feedback about these services by interacting with the marketplace as well. Figure 3 shows an UML business case diagram where the functionality of the marketplace application is summarized.

Content providers can create content catalogues where multimedia content is offered. These catalogues can be consulted by each actor of the business case. Service providers look for content providers which can supply multimedia content, with the aim of assembling services. These searches are published as auctions. The marketplace automatically invites content providers which can supply the content required by auctions. Those which accept the auction invitation compete by offering content they own under certain conditions (e.g. price, QoS, etc.). These conditions constitute a binding bid, which means that bids can neither be withdrawn nor modified after emitted. The auction creator selects the most suitable bid after the auction concludes. The offer is then formalized in a contract
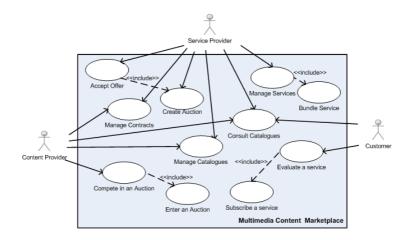

Fig. 3. DAM Marketplace UML business case Diagram

between the service and the content provider. This contract can be managed through the marketplace.

Contracts define the business terms of a service (a service can be defined by several contracts). Services are collaboratively carried out by contract(s) signers, and offered to customers. Customers can look for the most suitable services through service catalogues, and subscribe to them. As customers make use of services, they can evaluate them as well as the multimedia content they offer. Service and content evaluations can be done through the marketplace infrastructure. These evaluations provide valuable feedback to service providers, allowing a better service re-design which takes customer preferences into account.

It's important to note that the aim of the marketplace is not to be a passive business mediator, but instead to make a smart use of the heterogenous information stored. This implies the possibility of actively participate in business transactions (e.g. starting an auction if a content provider resigns). Additionally, the marketplace must mediate between the different information sources.

3.3   Suitability of Triple Space Computing

The current situation of the DAM solutions market is characterized by the existent of highly isolated and proprietary solutions of which ADAM, Apple's Final Cut Server, BrighTech's MediaBeacon, Canto's Cumulus, Chuckwalla, Nstein, OpenEdit, Phrasea, Wave Corp.' MediaBank, Widen and Xinet's WebNative Technology constitutes only a short list[3]. The isolated and proprietary nature of the existent solutions force the involved parties to anchor to some of them and to multiply efforts to support distinct platforms to offer their services.

Although some efforts have been made in the DAM standardization area like, for example, Adobe's Extensible Metadata Platform (XMP)[4], it is clear that there still exists a long way to get a universal DAM solution which efficiently deals with the inherent heterogeneity of these scenarios. The suitability of Triplespaces to implement a marketplace pattern has been motivated in [13]. Our business case is backed on this pattern, therefore we motivate now the applicability of Triplespace technology to its implementation.

As described in Section 2, Triplespaces provide de-coupled communication and co-ordination capabilities. Asynchronous communication is needed in the implementation of our marketplace, since actors might be able to interact in an auction or sign a contract without being online at the same time. Location de-coupling enables the remote communication among actors in a distributed fashion making it possible the decentralized deployment of the marketplace (e.g. with no central point of control or failure). Finally, reference decoupling ensures the transparent and private communication among actors, which do not need to explicitly know other actors to perform business transactions with them. The support of semantics ensure the schema decoupling. Business transactions and

---

[3] Refer to http://www.cmswatch.com/DAM/Vendors/ for a more detailed list and vendors' URLs.

[4] See http://www.adobe.com/products/xmp/

information retrieval can not only infer information not explicitly stated, but also mediate between heterogeneous data schemas used by actors.

In short, Triple Space Computing seems suitable for the implementation of a highly distributed worldwide DAM solution. The aforementioned capabilities ensure a rapidly implementation of a highly functional and open DAM scenario with a minimum consumption of resources, as detailed in the following section.

4   Implementation of the Multimedia Content Marketplace

In this section we detail the design and implementation of the Multimedia Content Marketplace based on Triplespaces. We first stress out the requirements of our system. Then, we describe the high-level architecture of our proposed solution. This architecture will make use of Triplespaces to couple with the identified requirements. An example of how Triplespace functionality is used to address the auction management logic is given. Finally, we describe the ontologies used in our solution, and motivate how semantics tackle the integration and interoperability issues that occur in the scenario.

4.1   Requirements of the Proposed Solution

In this section we list the functional and non-functional requirements needed for the business case implementation. These requirements translate the key requirement identified in [3]. Providing a thorough requirement analysis is out of the scope of this article, but a more detailed analysis can be consulted in [14].

From a functional perspective, publication and retrieval of the semantic information MUST be available. These functionalities SHOULD be blocking, so that information can be queried before it's already published. A destructive consumption of the information SHOULD be also provided in order to allow the information updating. Information MUST be able to be retrieved from different sources via joint operations. Filtering conditions MUST be provided as well. Subscription and notification mechanisms MUST be available in order to coordinate business interactions. The publication of these business transactions MUST be atomic. Consistency of the information published in these cases SHOULD be provided. Actors MUST be able to manage their own information, defining where it should be published and who could access to it.

The information MUST be available and correct for the auction, contract and user management functionalities. It SHOULD be available and correct for catalogue querying. All the information published within the marketplace MUST be kept consistent. Completeness of the information retrieval MUST be ensured to a local data source, and MAY be extended to a distributed data source. Security policies MUST be updatable by actors in order to grant access rights to the information.

From the knowledge specification point of view, the system MUST be able to formally define a taxonomy which enables the classification of multimedia

content. Properties and/or rules MUST be defined in order to formalize complex relationships between multimedia assets, actors, and business transactions. All this knowledge MUST be retrieved under the retrieval requirements stated before. The system SHOULD be able to couple with heterogenous information sources, and to actively moderate business transactions by understanding the messages exchanged by actors.

4.2   High-Level Architecture

Figure 4 shows the architecture of the multimedia content marketplace. The architecture is composed of three main components. First, a Front-End (FE) component, which exposes the interfaces used by the distinct actors to interact with the marketplace. Second, a Back-End (BE) component, which abstracts the communication between actors and the Triplespace infrastructure by exposing a Web service interface. The Back-End offers the marketplace functionality (i.e: catalogue creation and queries, etc.) to actors. Finally, the underlying Triplespace infrastructure provides with the storage, communication and coordination and semantic functionality used to implement the marketplace logic.

This architecture reuses the data schemas (XML Schema[15]) being employed in another marketplace implementation. A set of ontologies defines the knowledge needed in this scenario, which will be handled by the Triplespace infrastructure. These ontologies are encoded in OWL[16] (currently OWL Lite version),and will covered in detail later. The marketplace logic is implemented in the Back-End. This logic covers the catalogue, services, contracts, users and auctions management described in Section 3. The Back-End component abstracts the complexity related RDF triples handling and the Triplespace functionality invocation .

The modular nature of the architecture allows actors to interact with the marketplace with three different schemes. First, the usage of the Front-End provided
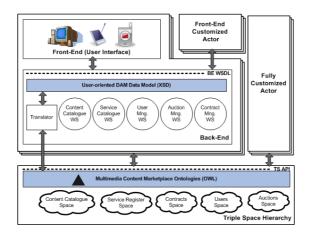

Fig. 4. Multimedia Content Marketplace Architecture

by the marketplace implementation. Second, the invocation of the Web services exposed by the Back-End, allowing actors to develop a custom interface. Finally, the direct access to the Triplespace infrastructure. This access would would be tied to the DAM marketplace data model (the set of ontologies), and the interfaces defined by the Triplespace infrastructure. It is important to point out that the most important functionality resides in the Triplespace infrastructure. Although the figure depicts five spaces, there can be several distributed kernels supporting this logical view.

The architecture described has been defined following two main principles. First, to narrow the logic between the actors and the communication infrastructure in order to fully test the capabilities of the latter. Second, to keep the implementation of the actors as flexible as possible. First principle is achieved by moving almost all business logic to the actors, allowing them to directly interact with the Triplespace infrastructure. Therefore there is no need of neither a central access point to the communication infrastructure nor external synchronization with other agents, following peer-to-peer principles[17]. Second principle has been achieved by the modular definition of the reference components implemented. The marketplace owner can decide the way these components are deployed.

4.3   Integration with the Triplespace Infrastructure

In this section we describe the logic of the auction management in terms of Triplespaces primitives as an example of the marketplace implementation. This sample has been chosen as it better shows the communication and co-ordination functionalities, using Triplespaces primitives.

The Figure 5 depicts the behavior of an auction life cycle in terms of interactions between actor. A *service provider* is looking for content, and starts an auction to get a provider for this content. *Auction participants* will join the auction and perform bids, which might be validated by the auction creator. An *auction creation space* is used to publish all auction calls. Each action is carried in a separate *auction space*. The hierarchy and access policies of the spaces can be seen in the right part of the figure as well.

The storyboard of these interactions is the following:

1. Potential auction participants are subscribed to the auction creation space in order to get notified if any content being searched in any auction can be provided by themselves.
2. A service provider publishes a new auction arrangement in the auction creation space, in order to get a content.
3. Auction participants subscribed to the space whose subscriptions are matched are notified of the new auction.
4. Auction participants interested in the auction can subscribe by publishing a request in the auction creation space.
5. The service provider validates all subscriptions received from auction participants (i.e, checking internal black lists).
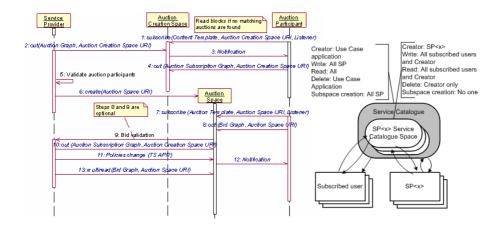
Fig. 5. Auction Management Interactions and Space Hierarchy

6. The service provider creates an Auction Space allowing all validated auction participants to write bids (no bid can be modified or deleted), as well as everyone to consult existing bids in the auction.
7. Subscription to auction management messages to get notified to things like winning bid change or auction end.
8. An auction participant writes a new bid.
9. The bid is validated by the service provider following its own validation logic.
10. If the bid is rejected, it is deleted from the Auction Space by the service provider.
11. Once the auction ends, policies of the auction space are changed, avoiding new bids to be published.
12. Auction participants are notified about the policies change.
13. The service provider reads all the bids published in the space in order to evaluate them.

4.4   Semantics of the Marketplace

Two of the core requirements of the multimedia content marketplace are the heterogeneous information integration and the support for an smart information handling, which can support an intelligent business mediation. The marketplace implementation uses a set of ontologies[5] depicted in the Figure 6, with the purpose of providing a common vocabulary to all actors.

   Multimedia content is defined in the *multimedia content ontology*. This ontology defines a taxonomy of assets. Multimedia content is usually retrieved using meta-data annotations. The formalization of semantic descriptions permits the classification of individual assets and the retrieval of inferred information as well. Next code defines a sample property using a TRREE rule:

---

[5] See http://www.tripcom.org/ontologies/dam.php

Fig. 6. Ontology Structure of the Marketplace Implementation

```
x daml:isSimilarToFilm y . [Constraint x != y]
x daml:hasActor a .
y daml:hasActor a .
x daml:hasDirector a .
y daml:hasDirector a .
------------------------
x daml:isSimilarToFilm y .
```

This rule defines that if two film were directed by the same director and some actor played a role in both of them are similar (i.e: The Godfather I and II). As a content provider adds "The Godfather II" asset to the marketplace, the system can infer this information without being explicitly declared. This results in being able to answer next SPARQL query (prefixes are omitted):

```
SELECT ?film_name
WHERE { ?film1 rdf:type daml:Film .
    ?film2 rdf:type dam:Film .
    ?film1 daml:filmIsSimilarTo ?film2 .
    ?film1 daml:hasName "The Godfather" .
    ?film2 daml:hasName ?film_name }
```

Previous query answers the question of "Please, give me titles of films that could be similar to *The Godfather*, which I liked so much". This useful information can't be derived from a non-semantically model without having to explicitly define each film similarity, which is not feasible in a real content catalogue.

The *business transactions ontology* imports concepts from an EDIFACT ontology [18], with the aim of solving the heterogeneity problems that arise when performing a business transaction. Finally, the *marketplace ontology* defines the knowledge related to the marketplace logic, such as actors and auctions related knowledge. Its objective is to facilitate the interoperability between actors and the semi-automatic handling of auctions (i.e: automatically starting an auction when a service is terminated by a content provider).

5   Evaluation of the Marketplace Implementation

In this section we outline the evaluation plan and consequent experiments to measure the suitability of Triplespaces to implement the multimedia content
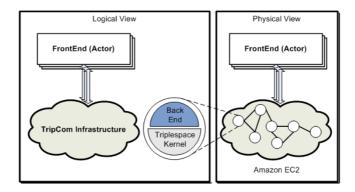
Fig. 7. Tests Architecture to Evaluate the Marketplace Implementation

marketplace. Suitability will be measured in terms of scalability and performance. Both factors are crucial from a business success perspective. Scalability ensures the deployment of a commercial scenario, with the objective of serving as many actors and customers as possible. Performance is crucial in terms of functional capabilities and response time.

We have considered some average figures based on the experiences of current services (e.g. the number of contents offered by a content provider in a real DAM application) in order to define some indicators. Indicators have been divided into scalability indicators (e.g. number of actors interacting within the marketplace) and performance indicators (e.g. content catalogue response time). Success factors have been defined for each indicator, defining expected results to be provided by Triplespaces.

These indicators will be measured through the execution of some experimental tests, whose structure is depicted in Figure 7. The tests will be deployed using the Amazon EC2 service [6]. Each node of the distributed infrastructure is a TripCom kernel [11]. This kernel will offer a marketplace Back-End implementation on its top.

Each experiment will vary the range of users and the number of nodes, and will measure each of the indicators defined. The combination of the aforementioned three coordinates is expected to be classified either as a suitable implementation or not for each individual experiment. The objective of the planned evaluation is to derive conclusions about the suitability of Triplespaces to implement a real scenario, given the expected size and performance with an affordable number of nodes. This can be seen as detecting if Triplespaces cover the technical and economic requirements to implement this real scenario.

6   Conclusions and Future Work

Multimedia content offer a lot of business opportunities that are worth to be explored by a telecommunication company. The collaborative nature of services

---

[6] See http://aws.amazon.com/ec2

based on multimedia content makes transparent communication and integration of data and processes needed. Triple Space Computing is a novel paradigm which efficiently combines the highly distributed coordination techniques of Space Based Computing with semantics. Triplespaces provide an implementation of this paradigm based on tuple spaces, extending the Linda coordination model so that RDF triples can be handled and Web services are supported.

In this article the authors have presented a business case centered on a multimedia content marketplace. Requirements which have arisen in the business case design have been presented. These requirements have motivated the use of Triplespaces and therefore semantics as the underlying communication infrastructure and knowledge representation technology respectively. While coordination capabilities of the Triplespaces have provided a de-coupled and easier implementation, semantics have been crucial to effectively address the heterogeneity of actors and the support of agile business transactions.

The ultimate objective of the prototypical implementation presented in this article is to validate Triplespace suitability for a commercial implementation of a multimedia content marketplace. In this sense, an evaluation plan has been outlined. The authors will carry out this plan with the aim of extracting meaningful conclusions about the scalability and performance indicators achieved by Triplespaces in this context.

Acknowledgements

References

1. Papazoglou, M.P.: Service oriented computing: Concepts, characteristics and directions. In: WISE 2003: Proceedings of the Fourth International Conference on Web Information Systems Engineering, Washington, DC, USA, vol. 3. IEEE Computer Society, Los Alamitos (2003)
2. Linthicum, D.S.: Enterprise application integration. Addison-Wesley Longman Ltd., Essex (2000)
3. de Francisco, D., Pérez, N., Foxvog, D., Harth, A., Martin, D., Wutke, D., Paslaru Bontas Simperl, E.: Towards a digital content services design based on triple space. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 163–179. Springer, Heidelberg (2007)
4. Austerberry, D.: Digital Asset Management, 2nd edn. Focal Press (2006)
5. Riemer, J., Martín-Recuerda, F., Ding, Y., Murth, M., Sapkota, B., Krummenacher, R., Shafiq, M.O., Fensel, D., Kühn, E.: Triple space computing: Adding semantics to space-based computing. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 300–306. Springer, Heidelberg (2006)
6. Lee, B.T., Hendler, J., Lassila, O.: The semantic web. Scientific American (May 2001)

7. Fensel, D.: Triple-Space Computing: Semantic Web Services Based on Persistent Publication of Information. In: Aagesen, F.A., Anutariya, C., Wuwongse, V. (eds.) INTELLCOMM 2004. LNCS, vol. 3283, pp. 43–53. Springer, Heidelberg (2004)

8. W3C: Rdf specification (February 2004), http://www.w3.org/TR/rdf-primer/

9. Gelernter, D.: Generative communication in linda. ACM Trans. Program. Lang. Syst. 7(1), 80–112 (1985)

10. Fensel, D., Lausen, H., de Bruijn, J., Stollberg, M., Roman, D., Polleres, A.: Enabling Semantic Web Services: The Web Service Modeling Ontology. Springer, Heidelberg (2007)

11. Martin, D., de Francisco, D., Krummenacher, R., Moritsch, H., Wutke, D.: An architecture for a qos-aware application integration middleware. In: Proceedings of the 11th International Conference on Business Information Systems. Lecture Notes in Business Information Processing. Springer, Heidelberg (2008)

12. Uhl, A., Pommer, A.: Image And Video Encryption - From Digital Rights Management To Secured Personal Communication. Springer, Heidelberg (2005)

13. de Francisco, D., Elicegui, J.M., Martin, D., Wutke, D., Murth, M.: Using triple spaces to implement a marketplace pattern. In: Procceedings of the 1st Space Based Computing as Semantic Middleware for Enterprise Application Integration Workshop in 1st European Semantic Technology Conference, Viena, Austria (May 2007), SpaceBasedComputing.org

14. de Francisco Marcos, D., Martin, D., Scheibler, T., Wutke, D., Harth, A., Murth, M., Simperl, E.P.B.: Tripcom requirements analysis and architecture profile for eai applications (April 2007), http://tripcom.org/docs/del/D8B.1.pdf

15. W3C: W3c xml schema definition language (xsdl) 1.1 part 1: Structures (August 2007), http://www.w3.org/TR/xmlschema11-1/

16. W3C: Owl web ontology language overview (February 2004), http://www.w3.org/TR/owl-features/

17. Minar, N., Hedlund, M., Shirky, C., O'Reilly, T., et al.: Peer to Peer: Harnessing the Power of Disruptive Technologies. O'Reilly and Associates Inc., USA (2001)

18. Foxvog, D., Bussler, C.: Ontologizing edi semantics. In: Roddick, J.F., Benjamins, V.R., Si-said Cherfi, S., Chiang, R., Claramunt, C., Elmasri, R.A., Grandi, F., Han, H., Hepp, M., Lytras, M., Mišić, V.B., Poels, G., Song, I.-Y., Trujillo, J., Vangenot, C. (eds.) ER Workshops 2006. LNCS, vol. 4231, pp. 301–311. Springer, Heidelberg (2006)