TripCom

Triple Space Communication

FP6 – 027324

Deliverable

D8A.1

TripCom Requirements Analysis and Architecture Profile for EAI Applications

David de Francisco Marcos
Daniel Martin
Thorsten Scheibler
Daniel Wutke
Andreas Harth
Martin Murth
Elena Paslaru Bontas Simperl

April 9, 2007

Executive Summary

This deliverable presents an EAI use case to demonstrate Tripcom technical capabilities in a Digital Asset Management business scenario. The deliverable presents a state of the art of current EAI solutions in order to point out possible benefits of applying Tripcom to such scenarios. In order to show an example of such application, a concrete use case focused on the implementation of a marketplace business model for the aforementioned DAM context is presented. This scenario is analyzed, giving the implementation guidelines and the validation indicators that should guide the implementation refinement. Finally, some conclusions are presented to the reader, and further technical documentation is presented for consulting purposes.

Document Information

IST Project Number	FP6 – 027324	**Acronym**	TripCom
Full Title	Triple Space Communication		
Project URL	http://www.tripcom.org/		
Document URL			
EU Project Officer	Werner Janusch		

Deliverable	**Number**	8A.1	**Title**	TripCom Requirements Analysis and Architecture Profile for EAI Applications
Work Package	**Number**	8A	**Title**	Use Case 1 - Enterprise Application Integration

Date of Delivery	**Contractual**	M12	**Actual**	31-Dec-06
Status	version 0.1		final ☐	
Nature	prototype ☐ report ☒ dissemination ☐			
Dissemination Level	public ☐ consortium ☒			

Authors (Partner)	David de Francisco Marcos (TID), Daniel Martin (USTUTT), Thorsten Scheibler (USTUTT), Daniel Wutke (USTUTT), Andreas Harth (NUIG), Martin Murth (TUW), Elena Paslaru Bontas Simperl (FUB)			
Resp. Author	David de Francisco Marcos (TID)		**E-mail**	ddf268@tid.es
	Partner	TID	**Phone**	+34(983) 548267

Abstract (for dissemination)	EAI technologies are backed by middleware solutions whose aim is to ease the integration of business applications inter and intra enterprises in a powerful, flexible and elegant way. For some emerging business contexts like multimedia content trading, the dynamic and collaborative scenarios are based on ad-hoc business relationships. These relationships make the coordination and semantic capabilities desirable in order to provide such a flexible integration.
Keywords	Enterprise Application Integration (EAI), Digital Asset Management (DAM), Space Based Computing, Triple Space, Marketplace Business Model, Enterprise Integration Patterns

Version Log			
Issue Date	**Rev No.**	**Author**	**Change**
2007-02-15	1	David de Francisco	TOC defined
2007-02-18	2	David de Francisco	Merged sections 2 and 3 from previous working documents
2007-02-26	3	David de Francisco	Merged TID subsections from sections 4 and 5 and annexes from previous working documents
2007-02-28	4	Daniel Martin, Daniel Wutke	Added EAI architectures section
2007-03-01	5	Andreas Harth	Added section 6
2007-03-01	6	Thorsten Scheibler	Added section 4.3
2007-03-01	7	Martin Murth	Added section 4.4
2007-03-02	8	David de Francisco	Filled deliverable template, introduction, conclusions and reviewed the document
2007-03-15	9	All	First review changes merged and document structure changed
2007-03-16	10	Andreas Harth	Merged performance requirements section with new document structure
2007-03-16	11	All	Merged changes suggested by the quality assessor's review

Project Consortium Information

Acronym	Partner	Contact
Leopold Franzens University Innsbruck http://www.deri.at	LFUI	Prof. Dr. Dieter Fensel Digital Enterprise Research Institute (DERI) Innsbruck, Austria E-mail: dieter.fensel@deri.org
National University of Ireland, Galway http://www.deri.ie	NUIG	Dr. Laurentiu Vasiliu Digital Enterprise Research Institute (DERI) Galway, Ireland Email: laurentiu.vasiliu@deri.org
University of Stuttgart http://www.iaas.uni-stuttgart.de/	USTUTT	Prof.Dr. Frank Leymann Inst. für Architektur von Anwendungssystemen (IAAS) Stuttgart, Germany E-mail: frank.leymann@informatik.uni-stuttgart.de
Vienna university of Technology http://www.complang.tuwien.ac.at/	TUW	Prof.Dr. eva Kühn Institut für Computersprachen Vienna, Austria E-mail: eva@complang.tuwien.ac.at
Free University Berlin http://www.ag-nbi.de/	FUB	Prof. Dr.-Ing. Robert Tolksdorf AG Netzbasierte Informationssysteme Berlin, Germany E-mail : tolk@inf.fu-berlin.de
Ontotext Lab, Sirma Group Corp. http://www.ontotext.com/	ONTO	Atanas Kiryakov, Vassil Momtchev, Ontotext Lab, Sirma Group Corp. Sofia, Bulgaria E-mail: vassil.momtchev@ontotext.com
Profium OY http://www.profium.com/	Profium	Dr. Janne Saarela Profium OY Espoo, Finnland E-mail: janne.saarela@profium.com
CEFRIEL SCRL. http://www.cefriel.it/	CEFRIEL	Davide Cerri CEFRIEL SCRL. Milano, Italy E-mail: cerri@cefriel.it
Telefonica I+D http://www.tid.es/	TID	Noelia Pérez Crespo Telefonica I+D Madrid, España E-mail: npc@tid.es

Table of Contents

List of Abbreviations

ANSI ASC X12 American National Standards Institute Accredited Standard Committee X12
API Application Programming Interface
B2B Business To Business
BPEL Business Process Execution Language
CD Content Distributor
CP Content Provider
DAM Digital Asset Management
DRM Digital Rights Management
DSL Digital Subscriber Line
EAI Enterprise Application Integration
ebXML Electronic Business eXtensible Markup Language
EDA Event Driven Architecture
EDI Electronic Data Interchange
EII Enterprise Information Integration
EIP Enterprise Integration Patterns
ESB Enterprise Service Bus
ETL Extraction, Transformation and Load
eTOM Enhanced Telecommunication Operations Map
FIFO First In First Out
HTTP HyperText Transfer Protocol
MOM Message Oriented Middleware
ODS Operational Data Store
QoS Quality of Service
RDF Resource Description Framework
RFC Request For Comments
ROI Return On Investment
RPC Remote Procedure Call
SOA Service Oriented Architecture
SP Service Provider
TS Triple Space
TSC Triple Space Computing
UML Unified Modeling Language
UN/EDIFACT United Nations / Electronic Data Interchange for Administration, Commerce and Transport
VAT Value Added Tax
WS Web Service
WSML Web Service Modeling Language
W3C World Wide Web Consortium
XML eXtended Markup Language
XSLT eXtensible Stylesheet Language Tansformation

1 Introduction

The purpose of Enterprise Application Integration (EAI) is the "unrestricted sharing of data and business processes among any connected applications and data sources in the enterprise" [23], where the term "business process" describes a sequence of activities to be carried out to reach a well defined goal (which can be either a material product or a piece of information) [21].

Typically, business processes even inside an enterprise are not realized using a single monolithic application, but involve multiple independent applications from different vendors based on different platforms which need to be glued together.

In the following sections, a short introduction of EAI in general is given and an overview of practices which are employed both in intra-enterprise and inter-enterprise application integration is provided, making use of so called integration patterns [18]. How Triple Space can improve or complement existing EAI solutions is also emphasized(chapters 2 and 3).

In order to illustrate such Triple Space advantages within EAI area, a use case from the Digital Asset Management business context is presented (chapter 4), giving the guidelines that will be used to implement this use case in the further work performed in D8A.2 (chapter 5). Validation indicators will also be given in order to evaluate the use case aimed objectives (chapter 6).

NOTE: Some of the information presented in this deliverable have been extracted from an accepted article which will be published in the next 10th international Business Information Systems (BIS) conference, written from this WP working team.

2 EAI state of the art

Linthicum[23] describes four types of EAI levels:

Data level It's goal is to favour the data reusing and sharing among enterprises and within an enterprise, between different departments.

Message level After defining common means for data sharing, applications within an enterprise need to communicate this date between different applications, exchanging information. This level focus on the data exchange.

Process level Data sharing and transportation affects to business processes implemented in the company. This level focuses on the integration of business processes executed in both intra and inter enterprise environment.

User interface level It's goal is to improve the access for different IT functionalities of the enterprise through user interface sharing, giving a uniform view of these IT functionalities.

In this deliverable we will focus on the first three levels, since TripCom is aligned with them. Hohpe [18] presents six common types of enterprise application integration, which can be mapped to previous EAI levels described.

To retrieve a certain piece of information, users often have to access multiple information systems. *Information Portals* can be used to integrate information from different sources into a single user interface, thus providing the user with a consistent look upon the data. Typically different business systems have to access the same business entity (e.g. customer information, billing data) which is stored in different local data stores. Updates to one data store have to be replicated to all other data stores in order to ensure consistency (*Data replication*).

Certain business functions which are common to multiple applications (e.g. is item in stock, retrieve the address of a customer) can be exposed to applications as *shared business functions*. This is the generic use case for Web service technology. Business functions can be encapsulated as virtual components [22] and shared as Web services.

Service-oriented architectures (SOA) [34] are built upon services, which are essentially shared business functions with added interface descriptions. The service descriptions are published in a service directory which can be used by service requestors to discover the services (i.e. business functions) they need. The services can be wired (or integrated) by means of service composition (e.g. workflow technologies) to new applications. *Distributed business processes* can be used to coordinate the interaction between distributed applications. Occasionally it is feasible to provide a business partner with access to certain shared business functions to enable interactions between enterprises (*business-to-business integration*).

All these EAI fields can be grouped into three EAI levels: data, message and processes.

2.1 Data level EAI

Data-level EAI tries to overcome highly-coupled distribution between many databases present in enterprise applications and application logic in a simple and efficient way.

In terms of EAI, coupling it's the binding level between logic and data, logic with logic and data with data. The less coupling level, the most reusable application logic is and the easier the integration between applications is. Data-level EAI tries to raise cohesion between data sources of the enterprise in order to share information between databases and applications, without regard for application and database changes.

This cohesion is needed because it's a common practice in enterprises to split up the storage of different business entities among different silos (i.e: clients may be stored in a CRM silo and payments can be stored in a general ledger system, but those concepts should be related in a billing process).

Data-level EAI is a usual entry point for EAI due to a great number of tools and techniques existing to integrate the information from database to database. Besides, it requires few significant changes to application logic or database structure as well as less cost to enterprises which are willing to integrate their systems [23].

One difficulty when solving a data-level EAI problem is the large amount of kind of databases (relational, object-oriented, multidimensional and legacy systems) and data schemas within the enterprise. Given the complexity and difficulty in accomplishing this desired end, it is also necessary to take into account both the model and nature of data bases.

Data-level solutions can be backed on batched integration (like in ETL solutions explained in Subsection 2.1.1) or on online integration (like some ODS solutions explained in 2.1.3.

2.1.1 Extraction, Translation and Load (ETL)

ETL uses traditional database middleware and database replication software in order to share information between different application databases using one-to-one, one-to-many, or many-to-many configurations based on a point to point topology. This approach is frequently used in migration and synchronization projects. This is a database to database data integration approach, which usually employs data warehouse based architectures.

When the same data schema is used for all databases involved in the integration process, only the exchange and load of the information between all databases is needed. However, when not all databases use the same data schema, information must be also transformed in order to translate it (maybe on the fly) to the source data schema.

ETL architectures have evolved deriving in three generation of ETL solutions which finally have make use of data warehouses and have evolved placing the transformation logic more and more located in a middleware layer instead of early proprietary transformation logic for each legacy system [35]:

2.1.2 Federated database EAI approach

Federated database EAI uses a virtual database model, which acts as a logical unifier of distinct databases with any brand, model or data schema which share no resources, creating a single, unified, logical view of the federated databases. This virtual database model exists only in software and is mapped to any number of connected physical databases using LAN connections. This way, enterprise applications get a logical unified view of the enterprise information through the horizontally partitioned data across the servers involved in the integration domain, providing a transparent view of all heterogenous databases [15].

The main advantage of this approach is that it relies on a middleware layer which shares information between applications and can easily used for further applications. Although there might be any performance factors to take into account when implementing a federated model, it provides the user a uniform view of several information sources, which is not often possible due to physical restrictions in data-warehousing approach [15]. Database federation can also be a base for Operational Data Stores.

2.1.3 Enterprise Information Integration using Operational Data Stores (ODS)

An EII solution is basically a translator. It translates from the language used by the application when submitting a query to all of the languages and interfaces supported by information sources that can answer that query, this way using all data sources as if they were only one central data source but not using a virtual database like in the federated database approach.

This solution also coordinates and optimizes interactions with back-end information sources, resolves any semantic inconsistencies thus providing semantic integration level and delivers a consolidated result to the requesting application in a format it understands [Pand05]. The information is extracted from enterprise applications and integrated into a data-hub using real-time processes or batch processes depending on how critical the information being processed is.

ODS architectures provides some benefits [8] which make them suitable to be applied to integrated reporting, record systems and multiple updating applications.

2.2 Message level EAI

Messaging is a fundamental technology for EAI, providing asynchronous and reliable communication, platform/language independency and time autonomy. It allows applications to be wired together in a disconnected, loosely coupled manner, providing many advantages over traditional RPC based solutions. It is based on 3 major principles: send and forget, store and forward and guaranteed delivery.

A message channel is a logical address defined in the Message Oriented Middleware (MOM) which identifies a connection between applications. Message producers and consumers exchange information by sending messages to and receiving messages from the channel.

2.2.1 Point-to-Point integration

Pont-to-Point integration describes the simplest possible integration architecture. Each component directly communicates with another component, requiring $O(n^2)$ message transformations for n participating applications. The Messaging middleware directly passes the messages to the target application, without going through intermediary components as in the hub and spoke or bus based styles.

Figure 2.1 taken from [26] compares Point-to-Point to Hub and Spoke integration.

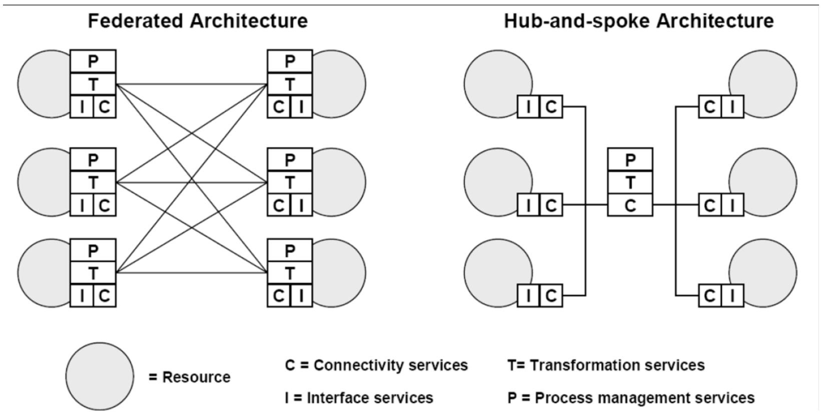

Figure 2.1: Point-to-Point vs. Hub and Spoke EAI architectures.

2.2.2 Hub-and-Spoke integration

Hub-and-spoke describes an integration architecture, where all applications send messages to a message oriented middleware (MOM). A message hub is a middleware that builds on top of a MOM, by adding message routing and transformation functionality [18]. The key benefit is the introduction of a target application independent, neutral message format reducing the number of required message transformations (source to target message format) from $O(n^2)$ (as in Point-to-Point integration) to $O(n)$.

2.2.3 Bus-based integration

Bus-based integration describes an architectural style, where the applications to be integrated communicate via messages sent to an Enterprise Service Bus (ESB). The service bus serves as an intermediary which mediates between different applications, providing features like message routing, message transformation, event handling and integration logic [5] in form of reusable services with a common interface description. Often ESBs are based on message oriented middleware (MOM), XML, XSLT and Web services. In contrast to hub-and-spoke integration an ESB is a system which consists of multiple services, each of them providing distinct functionality.

2.2.4 Messaging protocols within EAI

One crucial component in EAI systems is the format in which messages are encoded. Typically, such systems allow to specify their own message formats. However, since some message types are very common in business, such as purchase orders and shipping orders,there are several standards for business-oriented message types. Standards are especially important for Electronic Data Interchange (EDI) messages which are exchanged between systems and used as the interface to other businesses, suppliers, and customers. Some of the most widely used standards for specifying EDI message payload include ANSI ASC X12[1], UN/EDIFACT[2], and RosettaNet[3].

[1]http://www.x12.org/
[2]http://www.unece.org/trade/untdid/welcome.htm
[3]http://www.rosettanet.org/

Typically, EDI messages are used for transmission of documents across enterprises and therefore across EAI systems. Each message is translated from the internal representation of the originating system to the common standard. Encoding standards such as X12 and EDIFACT are oblivious to transmission protocols used to interchange such documents. They do not define how interchange partners shall establish the required communications link to exchange EDI data. Protocols are defined to transmit EDI messages via a variety of channels, most notably recently Internet protocols have been used, such as E-Mail[4] or HTTP[5].

2.3 Process level EAI

In the previous sections we have seen how to integrate data from heterogeneous sources (query-only), and how to exchange messages between components (either representing a query or a state change).

Use cases in reality are a bit more complex than just passing messages. For example, for a purchasing use case, typically a series of steps are involved. Traditionally, the business processes were embedded in the application logic [7]. Embedding business processes in application logic makes it difficult to adapt quickly to changing business environments. Especially when integrating workflows across enterprises, business processes which are hardcoded in a procedural language are too difficult and costly to maintain.

Workflow management technology has been applied for quite some time to support organizational processes that depend on information systems and human resources [29]. Prominent models of encoding the contol flow of an application in a declarative way are rule-based systems. (e.g. productions rule systems, Event-Condition-Action rule systems). E.g. MS BizTalk uses a rule engine for executing the business processes models. Other popular formalisms in the reasearch community for specifying processes are Petri Nets, UML Activity Diagrams, or Abstract State Machines. BPEL is the current state-of-the-art in industry for specifying and executing processes.

The idea in current research projects and upcoming products is to decouple the communication between services by introducing mediation components (I, C, and T in Figure 2.1), which translate between different message formats and vocabularies during run-time, and a discovery component, which selects appropriate services at run-time given the criteria specified during design-time of the process.

Newer systems, especially those based on an SOA paradigm, distinguish between orchestration and choreography. Orchestration pertains to functionality internal to the service, i.e. the pattern of interactions a Web service agent must follow to achieve a goal Choreography is concerned with specifying external interfaces, i.e. the sequence of interactions between a Web service and its user.

[4]http://www.ietf.org/rfc/rfc3335
[5]http://www.ietf.org/rfc/rfc4130

2.4 Added value of Tripcom in existing EAI solutions

2.4.1 Triple Space application to data-level EAI

Tripcom can enhance Enterprise Information Integration areas due to its approach combining Semantic Web technologies [31] and Tuple Space technologies [14]. Tripcom could replace or complement the role of the actual data hub in ODS architecture, making use of distributed spaces to share information from different enterprise sources and using semantic technologies to provide a common means for all these sources.

We can also think in Triple Space like the physical integration database which would substitute the virtual database of the federated model, because it performs the role of the data interchange level using the same data model (triples in the case of Triple Space) as depicted in Fig.2.2.

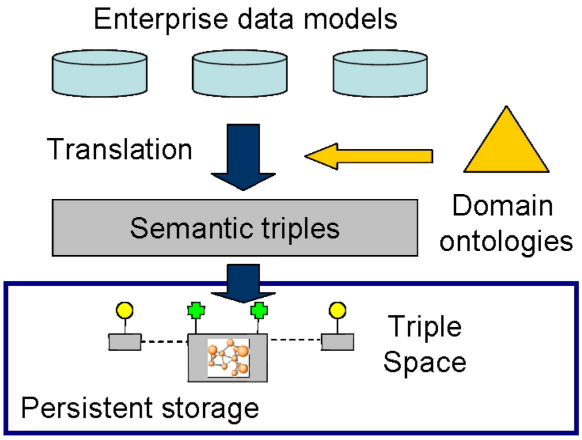

Figure 2.2: Role of the Triple Space inside an EII architecture

In this EII approach, data is collected from enterprise applications with their own data model and must be translated to a semantic model defined by the business domain ontologies from each enterprise. Once information is translated to a common data schema defined by this semantic model, the Triple Space acts as the data warehouse of the enterprise. Besides being a common data structure for all applications, the semantic meaning is also stored enhancing the information retrieval by solving the semantic integration problem presented in [3].

The Triple Space layer should take also care of the data hub functions, like security and trust, error checking and handling, data search and retrieval and optimization. All these functions will have special issues to be treated due to the semantic orientation of this approach and the special data model used as explained in [10]. Key points can be outlined here:

- The use of semantic triples as common data model for the whole enterprise enhances data management, improves maintenance of all applications which deal with this data and performs a semantic mediation task which enterprises keep on trying to solve.
- The process of data modeling needed to build a new enterprise application is governed by the ontologies which work in conjunction with the Triple Space, so it is simplified and standardized inside the enterprise.

- The use of a persistent storage space allows enterprise applications to communicate asynchronously, reducing bandwidth needed to perform data exchanges in a synchronous way.
- Asynchronous communication between enterprise applications allows them to communicate without needing to have data from other application ready from this application server, so improving fault tolerance.

However, some critical points to enterprises should be addressed in order to implement these kind of solution based on Triple Spaces in real systems:

- The effort needed to define an ontology as the semantic data schema for a an enterprise can be unaffordable for big companies. Even if reusing common business ontologies could be convenient for the company, there would be needed many company-specific concepts which are obviously used everyday within a company.
- Coexistence with current EAI solutions would be a critical factor for companies. This coexistence may derive in the creation of data wrappers or mediators to current solutions being used, thus making a lot of effort necessary to perform this integration.
- Performance in data definition, data translation, semantic mediation and data retrieval should be tested in order to offer an efficient solution.
- Security should be properly addressed by Triple Space security layer, since information stored in data warehouses are usually confidential.

The answers to previous points after Triple Space development will determine the applicability of Triple Spaces to data integration technologies in real systems.

2.4.2 Tripcom application to message-level EAI

Nowadays typical business integration scenarios are becoming very complex and dynamic. To satisfy these properties the SOA paradigm [34] is getting more and more important for EAI [18]. The so called event-driven architecture (EDA)[17] emphasizes the *realtime* aspect of a SOA. Its key characteristics are:

Broadcast Communication Participating systems broadcast events to any interested party. More than one party can listen to the event and process it.

Timeliness Systems publish events as they occur instead of storing them locally and waiting for the processing cycle.

Asynchrony The publishing system does not wait for the receiving system(s) to process the event(s).

Fine Grained Events Applications tend to publish individual events as opposed to a single aggregated event.

Ontology The overall system defines a nomenclature to classify events, typically in some form of hierarchy. Receiving systems can often express interest in individual events or categories of events.

Complex Events Processing The system understands and monitors the relationships between events, for example event aggregation (a pattern of events implies a higher-level event) or causality (one event is caused by another).

Comparing Tripcom and EDA one-by-one makes clear that Tripcom can be the perfect implementation backbone of an event driven architecture:

- The events have to be published in a reliable and persistent manner. This perfectly fits to Tripcom's "persistent publish and read" paradigm.
- In EDA (in contrast to a SOA) a client (whether requester or service) is responsible for updating subscriptions at the server - the server has nothing to do with the subscription besides of storing them.
- Furthermore a client has to be accountable for reading the event from the "source".
- The EDA backbone is responsible for continuously propagating relevant information to the subscribers. E.g. through notification.
- An EDA has to provide sophisticated model based subscriptions.

As all these aspects characterize Tripcom, it provides a well-suited foundation for modern EDA technology to implement message driven EAI.

2.4.3 Triple Space application to process-level EAI

Triple Space can be used both for communicating messages pertaining to the orchestration part of a service, and for messages pertaining to the choreography part. Such an architecture which utilizes a shared space for exchanging messages decouples message exchange in space and time.

Another benefit of using the Triple Space is that the entire communication history is potentially available at one central point, which would allow for rolling back or restarting processes in case of exceptions. By adding archival functionality to the space, the space can be used additionally to trace the communication history of a process.

3 Definition of inter and intra enterprises architectures using TripCom

In this chapter, the application of TripCom to inter- and intra-enterprise communication is explained. In Section 3.1, state of the art technologies are described and a set of criteria is provided to distinguish between these areas of communication based on their different focus. Section 2.4 shows the benefit provided by when TripCom is applied to these scenarios.

3.1 State of the art in intra-enterprise and inter-enterprise communication

Enterprise Application Integration (EAI) can be employed in two different scenarios: integration between applications (i) inside an organization or (ii) across organization boundaries. Both occurrences of communication and their potential uses in EAI are outlined in the following sections. Intra- and inter-enterprise integration architectures require functionality which fulfill the set of criteria outlined in the following subsections.

3.1.1 Intra-enterprise communication architectures

Intra-enterprise communication describes communication inside the same administrative domain, e.g. between applications of partners from different departments of the same enterprise. A typical use case for intra-organization EAI are service-oriented architectures.

Service-oriented architectures (SOA) build on services, which essentially are shared business functions with added interface descriptions. The service descriptions are published in a service directory which can be used by service requestors to discover the services (i.e. business functions) they need. Services can be integrated to form new applications by means of service composition (e.g. workflow technologies [21, 1]).

SOAs build on the concept of loose coupling, i.e. applications interact only via exchange of self-contained business requests in form of messages, while making as few assumptions as possible about the other communication partner. The messages which are exchanged between a service and a requesting client are defined according to the service description.

The middleware platform which is used to realize a SOA is an *Enterprise Service Bus (ESB)* [5].

3.1.2 Inter-enterprise communication architectures

The term *inter-enterprise communication* describes communication between applications from business partners from different administrative domains, which can be e.g. used to enable distributed business processes and B2B integration.

If a certain business process involves several business functions, which are already implemented in existing applications and exposed as shared business functions by another enterprise, distributed business processes can be used to coordinate the interaction between the applications involved (e.g. by using workflows [21]). This scenario is useful e.g. for outsourcing fragments of business processes. Generally business processess can be used for business-to-business (B2B) integration.

Inter-enterprise communication architectures extend intra-enterprise communication architectures with respect to e.g. standards, transformation, security and discovery. Both occurrences of communication rely heavily on messaging in form of message oriented middleware systems as the underlying *communication infrastructure*.

While in inter-organization communication the need for a *standardized message format* (e.g. EDIFACT [11], ebXML [25]) to capture the message payload is high, in intra-organization communication the focus often is placed on performance (message payload could e.g. be Java objects) rather than standardized message formats.

Even when standardized message formats are employed, *message transformation* might still be necessary to enable communication between certain communication partners. So message transformation capabilities are an important requirement for EAI in both inter- and intra-organization communication, and in fact an essential idea behind loose coupling.

Security in terms of encryption of message payload, guarantee of message integrity and verification of message sender important in both occurrences of communication. Establishment of trust relationships might be more important in communication across organization boundaries, when/where communication partners establish first-time communication channels.

In inter-organization communication, communication partners are often not known in advance. So mechanisms for *discovery* of communication partners are important here. Communication parterns can e.g. be annotated with semantic service descriptions and/or policies.

Furthermore communication infrastructure used both inside and across organizational domains are typically required to provide quality of services (QoS), such as *reliability* and *transactional behaviour*.

To conclude, the same criteria are relevant for the communication infrastructure used in both the intra- and inter-organization scenario, although each of them has a different focus. Since the partners involved in inter-organisation communication are typically from different administrative domains, agreement on a *standardized message format* is much more complicated than within organisational boundaries. Furthermore unsecure network links between geographically distributed sites demand a much stronger focus on secured and trusted communication channels as within an organisation (e.g. inside the same local network).

4 Use case scenario: Digital Asset Management (DAM)

4.1 Use case context

Digital content constitutes an important commercialization resource[1]. The Internet, and more specifically the Web, have enabled content providers to distribute content on a world-wide scale with very little cost, since content is distributed digitally. With the appearance of new wireless and mobile technologies, additional ways of consuming content have become available. For digital content, the full workflow can be carried out online, from offering content, browsing catalogs, delivery of goods, to final payment. This enables a vendor to get huge cost-savings by using the Internet (or mobile Internet) as a distribution channel for digital content.

4.1.1 DAM business domain

For telecommunication companies such as Telefónica [2], digital asset management and digital content distribution offers a large business market. Telecommunication providers can offer services based on digital content, primarily delivering content to end users by providing network communication infrastructure plus value-added services in the digital asset management domain. Telecommunication providers can leverage their existing bandwidth services, both in fixed and mobile networks, and offer distribution channels from providers of digital content to end users.

New services must be designed and created in very short amounts of time to satisfy possible market needs as soon as possible. This paper examines a specific scenario in the Digital Asset Management area. The intent is to supplement a sports news site, which is offered to customers of DSL lines free of charge, with multimedia content (audio and video) as a premium, paid-for service. To be able to quickly put together such service offerings, companies require methods to partially or fully automate the service creation process.

4.1.2 Actors involved in a DAM scenario

Within the DAM business domain many actors interact collaboratively to engage in e-commerce with multimedia content. Following the eTOM e-business reference model [12] – the most widely accepted standard for business practices in the telecommunications industry – we can identify the actors relevant to the DAM business domain.

A *service provider* provides content services to subscribers by storing a catalogue of proffered media content which it offers through a service portal. All the billing processes, customer care, user management, and user security issues are performed by or subcontracted out by the service provider.

This service provider needs media content to provide these content services. A *content provider* is a supplier which owns media content and wants to trade with it.

In order to address the unauthorized copying issue, a service provider needs to use a Digital Rights Management (DRM) system [6]. A *DRM provider* provides security functions such as privacy, integrity, or authentication which are especially suited

[1] See figures in http://www.naa.org/technews/tn981112/editorial.html
[2] http://www.telefonica.com/home_eng.shtml

for multimedia content [32]. The DRM provider plays the role of a complementary provider in the eTOM business reference model.

The service provider needs to distribute the selected content in a secure way to the final clients that purchased it. The *content distributor* is an intermediary that not only provides the subscriber access to the media content, possibly using different communication networks; it might also sell value-added services to the content provider, providing storage and bandwidth functionalities.

4.1.3 Inherent features of DAM scenarios

Mobility, security, quality of service, rights management and interoperability are key features offered by a service provider supplier. For that reason it is necessary for all the business processes related to DAM environments to be easily adaptable and configurable so that new business possibilities can become available in a fast and effective way. The DAM challenge is to provide agile and seamless interaction between these actors, apart from bridging different proprietary multimedia environments – the so-called content ecosystems in DAM domains – in conjunction with DRM technologies [19].

Due to the inherent heterogeneity of actors, content, and services involved in a content service design, different kind of transactions are usually held between service provider and different suppliers. These transactions should be held in parallel to save time and be maintained in a transparent way, so that service providers need not know which vendor is able to provide the content and services they need. To manage this heterogeneity all the actors involved should agree on a common communication language.

Communication between these actors should be reliable and ensure confidentiality since business data from different enterprises is being dealt with. Messages in such communications would derive from the service definition and should be easily accessed by actors involved in the content service due to the frequent modifications made to content services.

All the aforementioned features are currently addressed by using EAI technologies in order to solve all integrations problems that are raised in these collaborative scenarios. Data sources must be shared between different actors – employing for example a federated database between them –, messages must be exchanged them – using point-to-point integration in some real implementations –, and business processes and workflows are also shared between those actors.

4.1.4 An auction business model

In the past, telecommunication operators provided services based on a fixed set of core partners. More and more, enterprises and operators offer complex services drawn from an increasing number of partners. Some of the complex services are created with only little prior agreement based on ad-hoc business relationships between actors.

Automatization of these ad-hoc business relationships is a major goal for telecommunication operators. The automatization has the aim of providing more flexible business processes, reducing the ”time to market” of a new service offering.

One emerging business model to deal with these ad-hoc business relationships is the so-called auction model. In this business model, a Service Provider wishes to contract services from a set of usually trusted suppliers, to assemble an end-to-end service covering a business need. The composed end-to-end service will be offered to the customers of the Service Provider, who will rate the final service offered, evaluating its quality.

The Service Provider can follow an auction model to contract required services from the set of suppliers. The Service Provider requests the services needed to assemble the end-to-end service and a temporal deadline until when the request has to be served. Then, all target suppliers will offer their services following a bidding model until deadline expires and Service Provider selects the best offer made.

This auction-based model is suitable for ad-hoc relationships, where time to market time is a strong requirement. The customer's opinion about the end-to-end service is also very relevant, and despite getting the needed services supplied by a set of suppliers, the set of suppliers might be changed with very little effort.

The auction negotiation assumes that each supplier has published the offered services. The offers will be stored in a registry; an auction will determine which suppliers can effectively fulfill the Service Provider's needs.

4.2 Domain of the scenario for use case

The EAI use case presented in this document will have to deal with different kind of data from actors involved in the collaborative business process scenario, the auctions being hold and the management of the services and dependencies handling. A simplified version of the data model can be seen in Figure 4.1. A detailed specification of this data analysis can be seen in Section 5.1 following the elicitation methodology defined in [9].

An *auction* will be a set of active *bids* related to a *content type* the *Service Provider* is searching for. This will be the negotiation mechanism employed to get desired *contents* from *Content Providers*, and will be formalized by *contracts*. These contracts will be used to assemble *services* which will be offered to *customers* (an explanation of what assembling services mean is later given in Section 5.2). Those customers will perform *service evaluations* to the services they are subscribed to, influencing the service offered.

These evaluations are an attribute of a service because they can rate both the contents being offered within that service and other functional and non-functional properties of the service (like downloading speed or security provided) from a generic perspective. Rating a service usually implies rating the contents being part of the service, but contents are rated in the context of a service because a service context usually influences a content rating (i.e: a good content with a bad streaming server usually derives in a bad rating to the content when the real problem is the service context).

From a conceptual point of view we can divide the data storages in categories in order to address the distribution of data within the Triple Space:

- The *actor's definition* data, where all information of service providers, content providers and users who interact in this collaborative scenario is stored.

- The *content catalogue* deals with information related to the contents, which will be a public subset of the internal definition which every content provider performs for its contents.

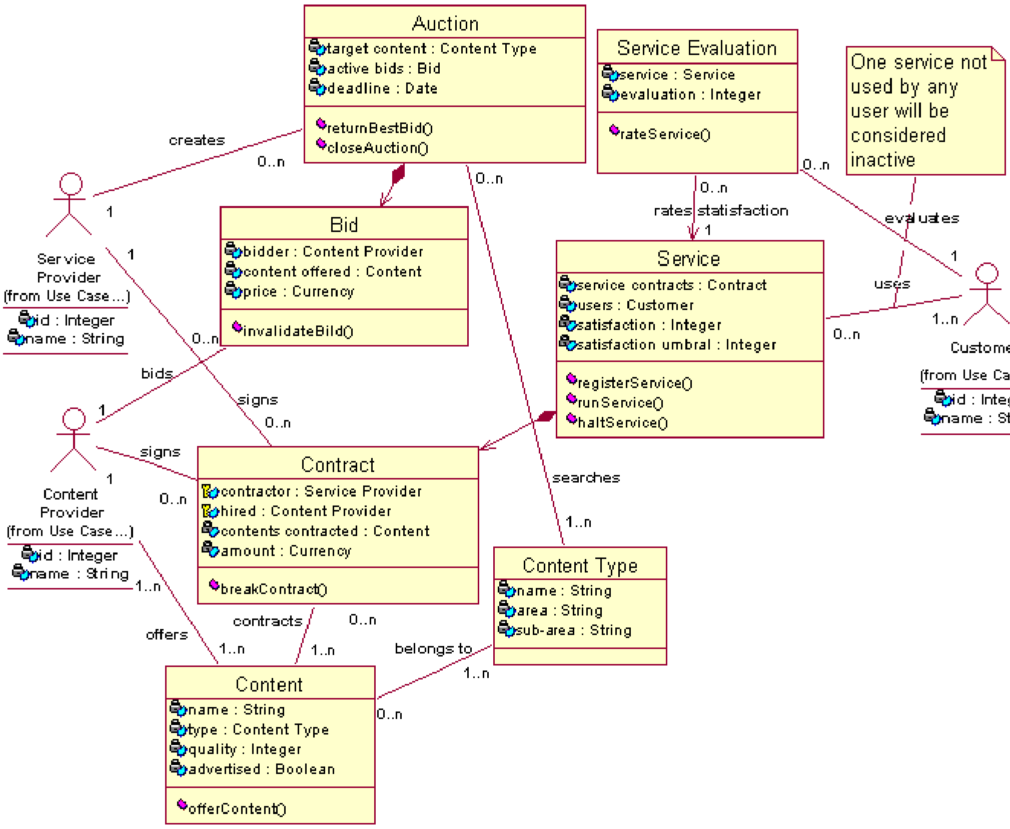

Figure 4.1: UML Class Diagram

- The *auction register*, which will store the temporal data that must be managed when an auction is created and has to be managed (with all active bids) until the temporal deadline specified reaches.
- The *service register*, where all services are stored once an auction process has finished and a service has assembled. All signed contracts being as well as service evaluations should be also part of this register.

4.3 Motivations for applying TripCom to the use case

The technology used to realize the DAM scenario presented in the previous sections needs to satisfy a series of core requirements:

- Due to heterogeneous message and data formats of the communication partners involved in the DAM scenario, mediation between those formats is an essential requirement. The integration should be performed according to a broker/bus-based approach so as to reduce the number of message and data transformations (see Section 2.2).
- It needs to be able to cope with a high number of loosely connected parties.
- The technology should support the non-destructive consumption of data. This means that the application data should be persistently stored so that it is accessible to other communication partners as well.

- It should support a reliable, fine grained publish-subscribe style communication mechanism, allowing partners to register for events based on their content rather than on a queue name
- It should include feasible instruments for the management of the access rights of the communicating business parties to the published information. This holds for the authentication mechanisms for both information providers and consumers.
- The integration infrastructure should ensure the security of the communication among the participants and with external applications

State-of-the-art EAI technology – mostly *Web services* on top of *message-oriented middleware* solutions as outlined in Section 2.2 – shows a number of shortcomings in providing a platform for integrating the parties involved in the use case scenario and fulfilling the aforementioned requirements. While current MOM products offer reliable communication based on the paradigm of publication and subscription, they lack scalability in terms of the number of interacting partners [2] and do not provide straightforward support for non-destructive consumption of messages. Most MOM systems also lack a means for real content-based data access (they rely on metadata). Furthermore, mediation between communication parties is limited to syntactic message and data transformation methods like XSLT, which primarily rely on simple document structure and text comparisons, rather than comparing the underlying concepts on a semantic level. For instance a field named "price" may or may not include VAT in the schemas of the data to be integrated, but a potential mismatch would not be identified at the syntactic level. These drawbacks are explicitly addressed by a Triple Space-based integration platform.

The *data mediation* functionality of the Triple Spaces allows different suppliers and service providers to have a common understanding of both the business terms of the transactions being negotiated, and the products and services mentioned in the transactions. This is achieved using semantic technologies, in particular formal ontologies and automatic mediation services. We differentiate between two ontology levels. The first ontology level includes an ontology for electronic business transactions, based on EDIFACT standards. The ontology defines the format of the messages exchanged between the actors involved in the DAM use cases in a machine-understandable manner. A second level is concerned with an ontology that models the business domain, in this case DAM-specific contents and services. Using the semantic mediation capabilities of the Triple Space, a common data schema for both suppliers and service providers can be automatically generated, thus allowing communicating parties to preserve their local autonomy with respect to the schemes used to store and manage the exchanged information. The usage of semantic technologies in this context further enables actors to perform reasoning and semantic validation tasks within the business transaction process, which leads to more accurate results in discovering content and services than pure syntactic matching mechanisms.

A second core feature of Triple Space technology is its *reliable transport mechanism* for Web services [20], which is an essential requirement for the electronic business transactions in our scenario. This communication is dynamic, allowing ad-hoc relationships between business parties, which can join or exit a negotiation depending on the satisfaction of business rules, verified by the Triple Space using semantic mediation. This behavior is highly desired within a DAM area, in which the inherent dynamism of the business domain results in frequent availability and functionality changes of content and services.

From a technical perspective, the implementation of the use cases using Triple Spaces follows the *publish-subscribe* paradigm. Message consumers express their interest in messages by describing their content, rather than listening on certain topics. This is an important difference between this and related state-of-the-art approaches. Even though there are MOM systems available that support content based subscription, these subscriptions are based on syntactic content matching while Triple Space allows for subscribing to semantically defined data publication. In existing Enterprise Integration Architectures [16] based on MOM and/or Web services [33] messages are *pushed* to certain destinations – identified by e.g. queue or topic names. This means that an endpoint reference has to be explicitly specified as destination of a message and exchanged before the actual message exchange is carried on. By contrast message publication in Triple Spaces can be done without knowing the receiver of a message because data is simply published to a space not addressing a communication partner directly. This is necessary when dealing with DAM business processes because service providers may not know the suppliers of media content and services in advance. In the space-based approach, message receivers *pull* messages from the space by describing their content. This interaction paradigm greatly simplifies ad-hoc communication between previously unconnected parties because communicating partners do not need to share any *a priori* information about each other.

In the DAM context suppliers prefer to offer their content or services to potentially interested service providers. This requires a mechanism that enables *multiple, non-destructive consumption* of messages. In the Triple Space-based approach, this is achieved by persistently storing messages that represent a supplier's content and service descriptions in the space, thus enabling all interested parties to non-destructively retrieve them (read operation) in a way similar to broadcast communication. In turn, service providers need to be able to retrieve information about already existing content offers in order to provide the services they are designing. Order, time, and publisher of this information are irrelevant. Current MOM solutions typically deliver messages in FIFO or priority order. In terms of the DAM scenario this would require each service provider to locally store all content offers and keep them up to date. By contrast, space-based technology offers persistent storage of these messages, classifying the information of each message according to its content's semantics, not depending on its publication time or sender. Using such storage policies, the middleware is able to implement random access and advanced query mechanisms to retrieve relevant information for each service provider.

In summary, we can say that although existing EAI solutions could be used to implement the presented use case, employment of a Triple Space will simplify the realization of the use case and allow for a more intuitive use of the provided integration mechanisms.

5 Requirements analysis to realize DAM scenario

Before presenting the way TripCom is going to be applied to the scenario defined, we will describe the requirements extracted from DAM and auction scenario which TripCom will have to cope with.

These requirements have been divided in data, functional and non-functional requirements.

5.1 Data requirements and restrictions to model

In order to implement a prototype within the scenario presented in Section 4, we will simplify some domain information that would be present in an implementation for a real system. These simplifications are made because we consider they don't provide extra functionalities to test a Triple Space implementation.

- We will only deal with Service Providers and Content Providers as actors for our prototype. We won't consider DRM providers (thus skipping the legal issues management) and Content Distributors involved in offering services in our auction model.
- We will assume that content types are a small subset of the real amount of media type present nowadays in media market. We will use video files, audio files and images as the only media types available.
- We will restrict the set of visualization devices to be employed to some of the most relevant examples, and not considering all real features that categorize current devices.
- We will skip all communications details when assembling a service, assuming that a Service Provider manages the communication or it was defined before the content assemble takes place.

Domain model presented in Section 4.2, and restrictions presented in previous list derive some data requirements for the Triple Space data layers. These requirements are presented in Appendix A, following the templates described in [9].

5.2 Functional analysis of the use case

UML use cases within the realization of EAI use case are depicted in next figures. See Appendix B for detailed use case descriptions. This detailed description follows the elicitation methodology defined in [9].

These are the key functionalities that the Triple Space marketplace implementation should cover. The actors represented in the diagrams are the following:

Service Provider Is a company who wants to offer some multimedia content to final customers but lacks the contents in order to offer it. It will take care of the Digital Rights Management service (DRM), the content distribution and the billing processes.

Content Provider Performs a supplier role in this scenario. It hires multimedia contents to the Service Provider demanding them.

Customers Final users of the scenario. They subscribe to services offered by Service Providers and evaluate these services.

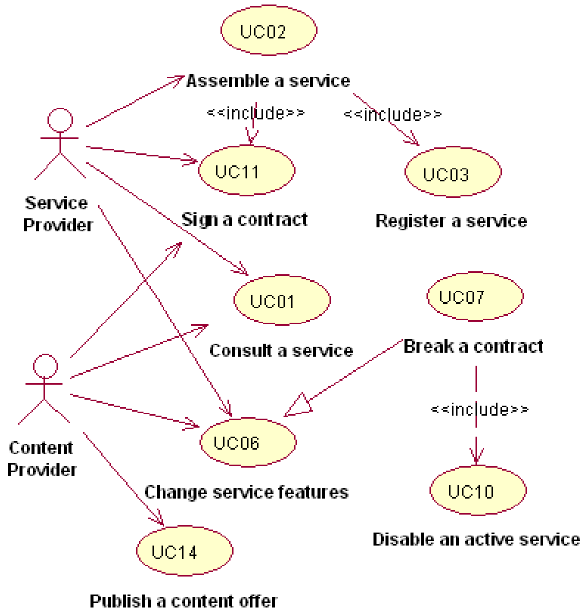

Figure 5.1: UML Use Cases Related to Service Life Cycle

End-to-end services follow a life cycle defined by the functionalities shown in Figure 5.1. An end-to-end service is *assembled* by a Service Provider, getting the content(s) needed from those *published* by Content Providers. Assembling here means to offer a service which provides several contents, which may be delivered by different Content Providers – for example a european football leagues highlight would be an end-to-end service assembled from football matches got from different TV operators –. These contents are provided by Content Providers through a business relationship formalized with a contract, which has to be *signed* by both Service Provider and Content Provider. Once an end-to-end service is assembled, it is *registered* to be managed.

Registered end-to-end services can be *consulted* by any actor in the scenario to get the public information of the end-to-end service or any contract information if the actor is part of the contract.

End-to-end services can also be *changed* by modifying some parameters with the agreement of all actors involved or by *breaking some contract* associated to the end-to-end service. These changes may demand to temporary disable the end-to-end service meanwhile it is re-designed.

Negotiation between Service Providers and Content Providers in order to get the needed contents to assemble an end-to-end service follows an auction model (see Section 4.1.4) as depicted in Figure 5.2. In this scenario a Service Provider will *call an auction process* in order to get desired contents. Each target Content Provider will be able to *perform bids* and *consult active bids* within the auction to adjust its offers. In order to ensure the reliability of bids, Content Providers must be *identified* within the system.

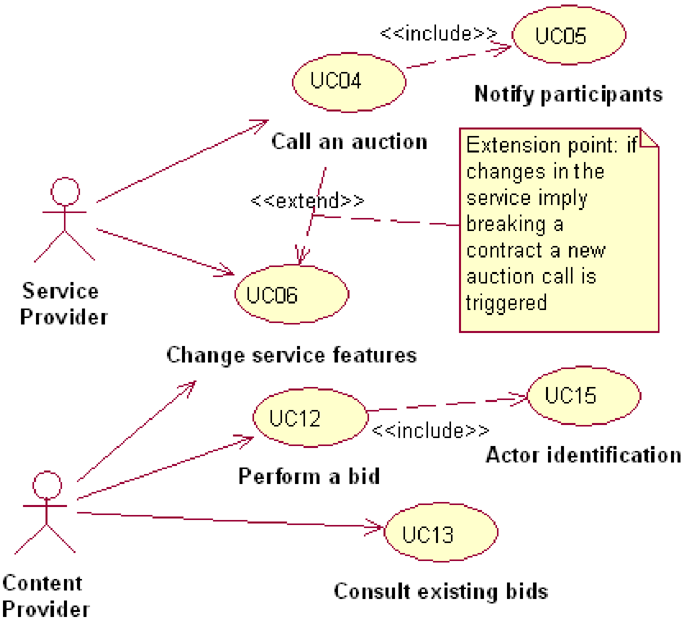

Figure 5.2: UML Use Cases Related to Auction Life Cycle

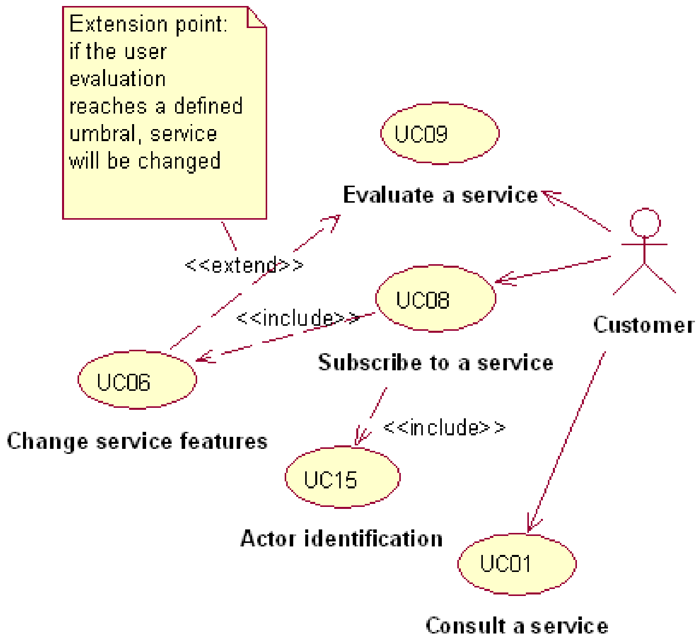

Figure 5.3: UML Use Cases Related to Customers Interaction

Customers interact in the scenario described by *subscribing to services* as shown in Figure 5.3. Subscribing to an end-to-end service means using the service, and gives them the chance of *evaluating this service*, influencing the Service Provider offer. Again, a customer must be *identified* within the system in order to perform such an evaluation.

5.3 Non-functional requirements

The use case described makes use of the coordination and semantic capabilities provided by Tripcom [24] in order to implement a marketplace business model.

This use case stores information in Triple Space distributed nodes making use of persistent storage and distribution capabilities explained in [30]. This information will be queried according to some query patterns defined in [27] for the concrete queries needed for this model.

The use case assumes the application of a Triple Space node in a commercial setting. Thus, TripCom must provide scalable storage and query functionality to provide for a large number of concurrent requests. To cater for scalability, the architecture of TripCom may be distributed.

TripCom must provide functionality to store in a scalable way

- massive amounts of content catalogue information (in the order of 10^7 content descriptions)
- a large number of service descriptions (in the order of 10^3 service descriptions)
- a moderate number of mappings between schema concepts and instances (in the order of 10^2 mappings)

TS has to provide an API for web services for periodical (e.g. once a minute) checks as to whether the underlying dataset has changed. Such a change can happen if the Service Provider puts a request for content into the system. Since content providers are interested in selling their content, the content provider sales service should be able to monitor requests for content to adjust their pricing strategy.

To lower the load on the TS, the TS implementation may provide functionality that allows for notification messages sent to relevant parties once a message enters the system.

Query response times are crucial for adoption in commercial environments. Messages which do not require schema mediation should be returned in sub-second response times. Queries involving complex matching operations should be completed in the order of minutes. However, depending on the number of complex matching rules in the system, a TripleSpace node may take longer to match request to content descriptions.

To be able to stay operational and responsive over time even in the presence of huge numbers of messages exchanged, there may be functionality to purge content to external storage.

In order to handle the dependencies modeled in the services being designed, a Web service approach will be used. Discovery, mediation, orchestration and choreography capabilities will be needed as pointed out in [28].

The system analyzed in this deliverable assumes that all actors accessing the system are trusted actors, and that a reputation manager will evaluate this issue. Also the identification of the actors when performing some concrete operations like bidding or accessing a contract information assumes that Tripcom security layer will handle these issues as explained in [4].

6 Realization of the DAM use case

6.1 Functionality to be implemented within the use case

In order to implement the aforementioned DAM use case, we will have to apply the auction model to DAM domain defined. This means that auction negotiations and dependencies handling will have to be implemented.

6.1.1 Application of an auction model to a DAM scenario

In order to implement an auction business model in a DAM scenario using Triple Spaces, the aforementioned register would be stored (possibly in a distributed way) in a Triple Space implementation. As shown in Figure 6.1, a business flow takes place between the service provider and its content suppliers. These business flows comprises next steps:

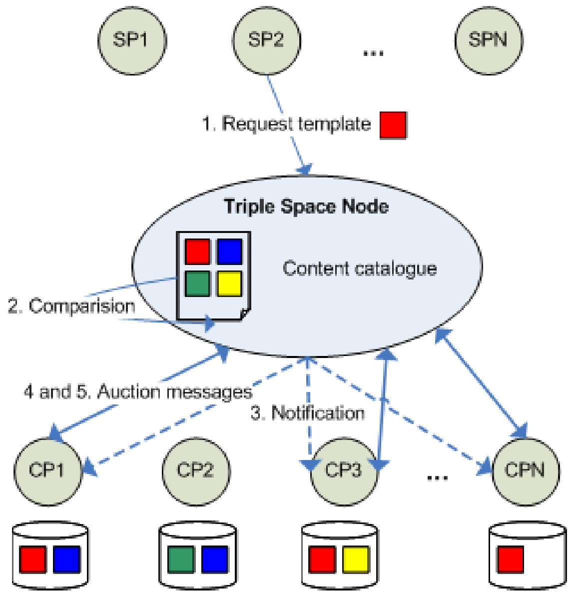

Figure 6.1: Auction Negotiation Implemented with Triple Space

1. **Template publication:** A Service Provider demanding some kind of content (a red one in previous example). In this request, the service provider also specifies the time until any content provider can send offers to this request.
2. **Content providers finding:** Since all service offerings are persistently stored in the Triple Space, the system checks which content provider can provide the content the service provider is demanding. For our example CP1, CP3 and CPN can provide a red type content.
3. **Supplier notification:** The Triple Space notifies to the corresponding content providers both the request and temporal deadline.
4. **Asynchronous offers:** Content Providers can make offers asynchronously, sending messages to the Triple Space. These offers will be internally stored until the process ends reaching the deadline.

5. **Asynchronous offer checking:** All Content Providers involved in the auction can also asynchronously check published offers. These offers will be ranked by the Triple Space, making the consultant aware of the current best offer so that it can improve it.

6.1.2 Handling dependencies of these ad-hoc business relationships

Dependencies within an end-to-end service means the business relationships shared between different actors. These business relationships make a Service Provider to depend on services from suppliers to offer an end-to-end service. As shown in the example depicted in Figure 6.1, and assuming CP3 provided the best offer during the auction negotiation, a business relationship between SP2 and CP3 has been arranged. This relationship consists on CP3 providing a red-type service so that SP2 can offer an end-to-end service to its customers. This relationship will be stored inside the Triple Space when the end-to-end service is assembled.

In ad-hoc business relationships, these interdependencies are very dynamic, because of i)new customer needs or customer evaluation about the end-to-end service being offered by the Service Provider and ii)possible change of conditions in the services supplied to the Service Provider. Both situations derive in a final service redesign in order to satisfy customers expectations.

We are going to focus in the dependencies originated by the customer interaction, since dependencies originated by suppliers follow the same steps that we are going to present. Handling these interdependencies with a Triple Space implementation consists on the steps depicted in Figure 6.2:

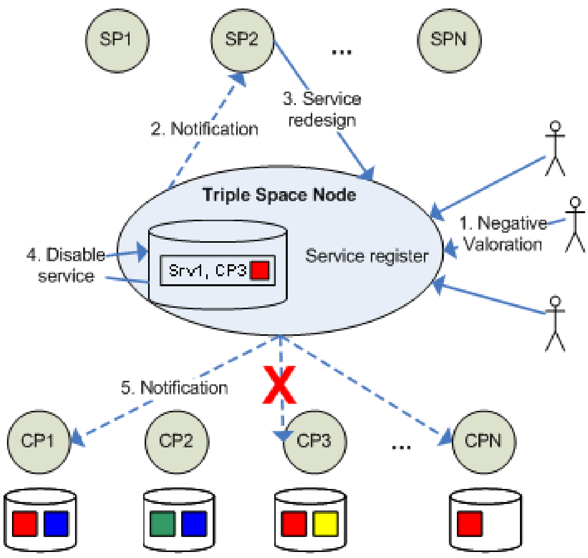

Figure 6.2: Business Interdependencies Handling Implemented with Triple Space

1. **User evaluation:** Users provide feedback of the end-to-end service they are receiving from the Service Provider asynchronously. This feedback is collected by the Triple Space.

2. **Notification to the service provider:** Users feedback is sent to the service provider periodically so that it can check the quality levels defined. These quality levels are part of the business rules each Service Provider implement in a custom manner.

3. **Service redesign request:** When users satisfaction reaches an umbral defined in business rules aforementioned, the end-to-end service must be redesigned in order to satisfy customer needs in a proper way. This request is made automatically to the Triple Space.

4. **Event resolution:** The Triple Space automatically performs the action to be taken to start the final service re-design. Examples of these actions can be pausing the end-to-end service offering, notifying users, notifying content providers or finding inter-final services dependencies and handling them.

5. **Suppliers new auction notification:** All Content Providers candidates to substitute the content being replaced are notified, starting the auction process explained in previous section. Obviously, the content provider offering the replaced service is not notified again.

6.2 Proposed architecture

In order to implement the aforementioned use case using Triple Spaces, a use case architecture has been defined (see Figure 6.3). This architecture defines the core functional components needed to implement the marketplace model presented, and serves as the entry point for the actors (Service Providers, Content Providers and customers) to the system. Besides, it communicates with the Triple Space nodes, managing the information stored.

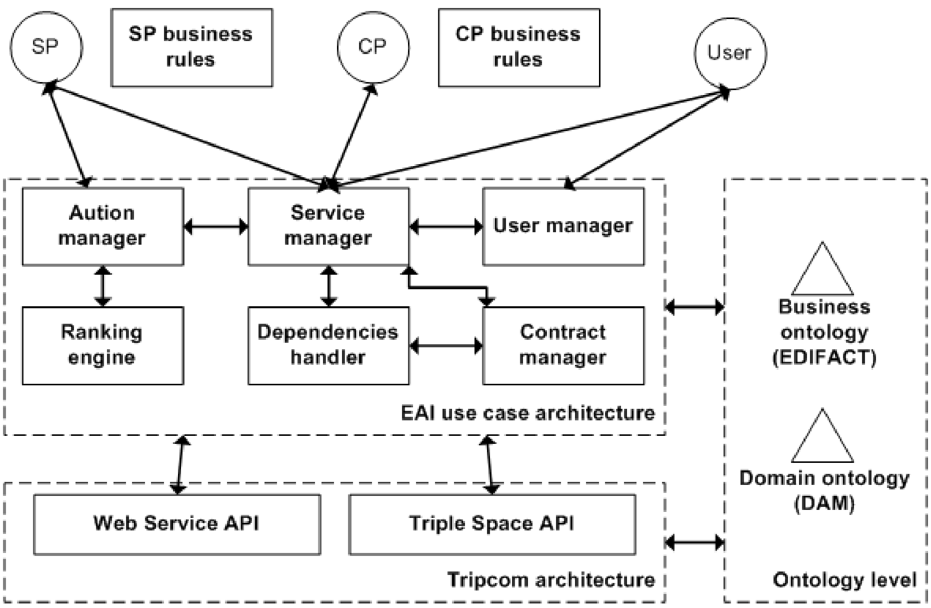

Figure 6.3: Use Case Architecture

The actors have been represented with circles, and both Service Providers and Content Providers have been depicted with a *business rules* box, which models all business logic relevant for the actor within the use case. Example of these business rules can be quantifying how interesting is a content for a Service Provider in order to rank possible offers or what can be the bid limit for a Content Provider.

Next we will describe the components of the architecture depicted in the architecture image. These components, which contains the logic of the use case, will be developed during the implementation tasks of WP8A next year.

The *Auction manager* takes care of handling auction calls, notifying the target Content Providers, managing the bidding process and closing the auction. Once the auction is closed the *Ranking engine* will return the Service Provider the results ranked according to the Service Provider business rules.

The *Service manager* is responsible for maintaining the service register, allowing the service registration, service querying and service deleting. These services are defined by contracts, which are managed by the *Contract manager*. Is a service needs to be redesign, this service will be temporary down and this has to be managed in a secure manner. This is performed by the *Dependencies handler*, which manages all relationships within an active service.

Finally customers subscribes to services and provide feedback to the services they are using through the *User manager*. They can also unsubscribe to services through this module.

This use case architecture makes use of a two level ontology. A business ontology based on EDIFACT standard [13] models all the negotiation relevant concepts. These concepts are used within auctions and the bidding processes, as well as the signature of contracts, when the business relationships are defined. On the other hand the domain ontology takes care of all DAM relevant concepts which are used when dealing with content services, content hierarchy or actors of the domain. Examples of information to be covered in this ontology are content metadata, actors metadata, digital rights management (DRM) metadata if needed, etc.

The actors of the use case communicates with the Triple Space via both the Web services API and the Triple Space API. The first one will be used when complex operations are executed (like halting a service):

- *Dependencies handler* will make use of this API when interacting with actors in order to halt a service.
- *User manager* will make use of this API when having to notify or dialogue with customers.
- *Contract manager* will make use of this API when handling a contract break process.
- *Auction manager* will make use of this API when setting the auction, notifying target suppliers, validating bids and closing the auction.

On the other hand, the Triple Space API will be used for direct data operations, like querying a service or registering a new content:

- *Services manager* will make use of this API when registering or retrieving services or contents data.
- *User manager* will make use of this API when registering or retrieving user data.

- *Contract manager* will make use of this API when registering and retrieving contract information.
- *Auction manager* will make use of this API when registering and retrieving bid information.

More information about the Triple Space architecture can be seen in [24].

6.3 Behavioral view of the proposed architecture

In the following section, a number of important scenarios are identified based on the DAM business scenario and the auction business model presented above. These are illustrated with the help of Enterprise Integration Patterns (EIP) [18], which present abstract steps from a messaging perspective that have to be performed to achieve the goal of a scenario. On the abstraction layer of EIP, nothing is said about the real implementation. This question will be answered in further work. Each scenario description is a set of interactions between an external component and the space. However, a single EIP does not necessarily represent an interaction between the space and an external entity.

This section is divided into five subsections. In the first subsection we present the main scenario of the use case (see 6.3.1). Afterwards we describe other scenarios which are needed within the main scenario or present some different aspects of the use case. The last scenario is about service deregistration. To distinguish between the terminology coming from the EIPs and the terminology used in the DAM scenario, all EIP will be written in italic letters (e.g. *Polling Consumer*).

6.3.1 Scenario 1: "Service creation"

The goal of this scenario is to create a new service. When the scenario has been successfully completed, a Service Provider (SP) can provide a so called "Service" to end users.

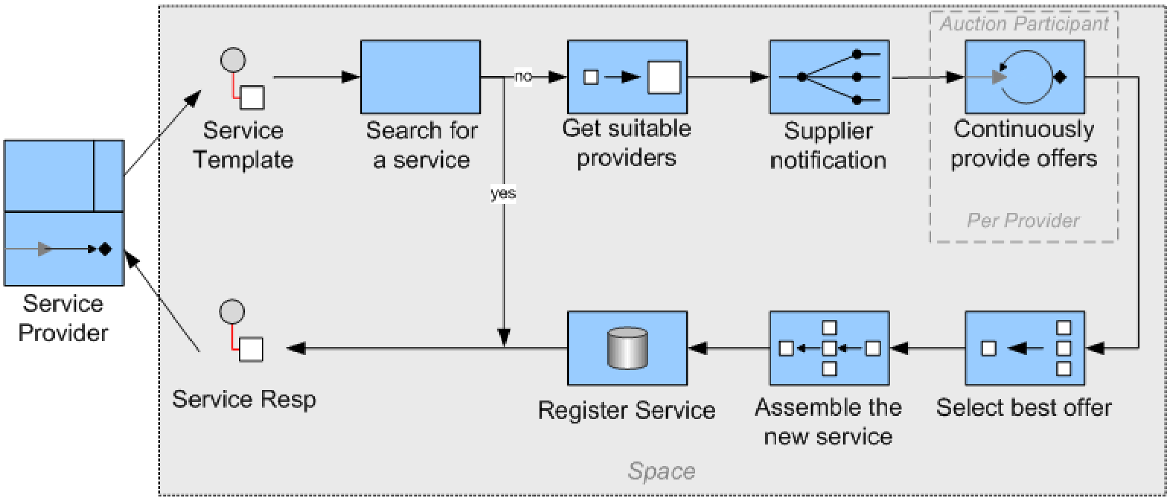

Figure 6.4: Service creation with EIP

Sequence

1. Search for a service (see 6.3.2)
   (a) If service is available $\Longrightarrow$ ready
   (b) No such service $\Longrightarrow$ go on
2. Search for suitable providers (see 6.3.3)
3. Notify all suitable providers
4. Provide offers asynchronously and check other offers (see 6.3.4)
5. Select the best offer
6. Assemble the new service
7. Register the new service

Description

A SP sends a template to the space to "inform" the space that it is looking for a (new) service. Therefore we first have to search if such a service exists, i.e. if such a service is registered (see 6.3.2). If no registered service could be found, the space searches all possible providers for such a service. A message has to be augmented with the appropriate information. Thereafter all suppliers will be notified that they can participate in an auction of a service request. The next step is the main part of this scenario. Now an auction takes place where all interested suppliers can place offers for the service (see 6.3.4).

For this behavior there is a need for a new pattern. We take the former known pattern *Polling Consumer* as a base and extend it with the ability of not only consuming but also sending update messages to the auction. We illustrate that with a new pattern called *Auction Participant*. When the auction is finished the best service will be selected. Afterwards the service provider and the content provider formalize the contract, then the service is assembled. At the end we have to register the service - in our case illustrated by a *Message store* where information is persistently stored. The result of the overall service creation (service response message) will be delivered to the SP. Because the SP is an *Event-driven Consumer* the SP gets an event that the creation of the service is finished. Now the SP can fetch the message from the space.

6.3.2 Scenario 2: "Search for a service"

This scenario shows how the space determines if a requested service is already registered.

Description

A Service Provider sends a template to the space. Within the Space an answer is generated which includes all possible services (may be zero). This can be seen as an enrichment of the request message (*Content Enricher*). Afterwards the SP receives the answer to his query. The SP is a so called *Event Driven Consumer*, that means he will get an event as soon as the response on his request is available.

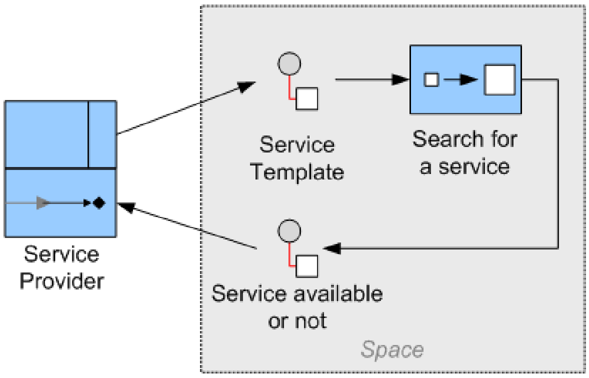

Figure 6.5: Search for a service with EIP

6.3.3 Scenario 3: "Get possible providers"

Description

A template request is submitted to the space. All content providers check whether they can provide such a service or not. Therefore the providers possibly have to provide service details. Each provider answers to this request. The answers are aggregated (*Aggregator* Pattern)so that all suitable providers are included in the answer.

6.3.4 Scenario 4: "Auction participitation"

This is a quite complex scenario, its detailed description can be found in chapter 4.1.4.

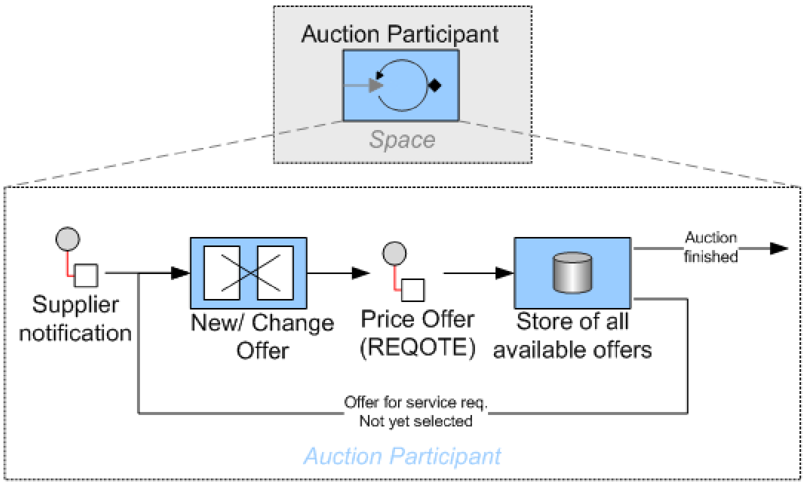

Figure 6.6: Auction participitation with EIP

Sequence

1. Supplier is notified that a new auction takes place
2. If so, he creates a new price offer
3. If the auction is not finished he can check the other offers

4. If he wants to, he can change its price offer and send it to the "auction"

5. If the auction is finished the participant cannot change its offer anymore

Description

A Content Provider participates in an auction. Because of that he is a so called *Auction Participant*. An *Auction Participant* gets informed when a new service is requested. Subsequently he puts a price offer into the *Message Store* associated with the auction. Afterwards he can check the *Message Store* by polling for other offers. If he wants to update its own offer he can change his previous offer and put it into the "Auction *Message Store*". When the auction is finished he cannot access the *Message Store* anymore.

6.3.5 Scenario 5: "Service deregistration"

In certain cases it is possible that a service has to be deregistered (see UC10 in Annex B). This scenario shows the required steps to process such an deregistration.

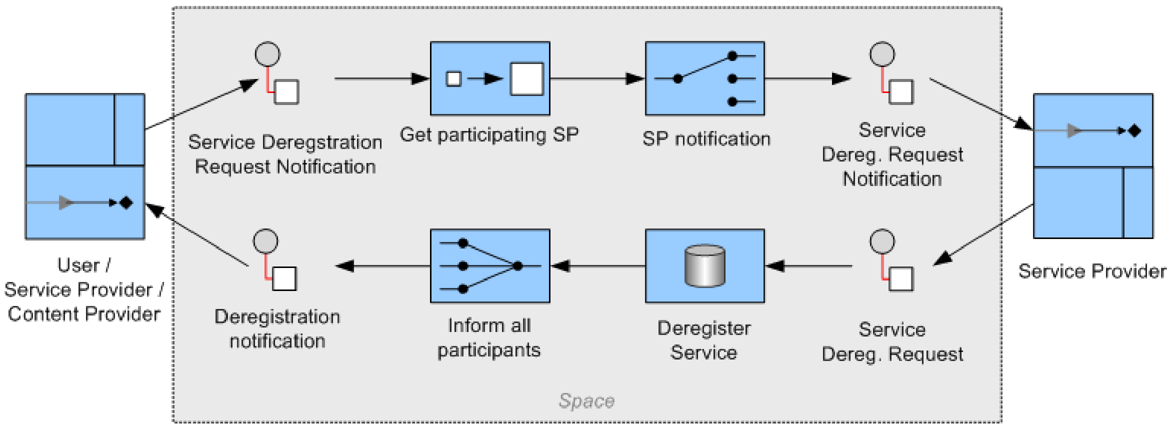

Figure 6.7: Service deregistration with EIP

Sequence

1. A user, SP or Content Provider (CP) requests that a service should no longer be available

2. The participating SP is determined

3. The SP is being informed about that notification

4. The SP sends a message about the deregistration of a service to the space

5. The service is deregistered

6. All participants of this service are informed about the deregistration

Description

After a user, a SP or a CP detects that a service doesn't deliver the desired content or the requests for content have changed, the according participant notifies the supplier of the service that it should be deregistered. Afterwards the SP which owns the service is determined and there after notified. Subsequently the SP sends a Service deregistration request to the space. The rervice will then be deregistered. Finally, all participating parties will be informed with a deregistration notification.

7 Conclusions

In this deliverable we have presented some general EAI scenarios where Tripcom can be applied providing a set of technically backed benefits that are worth the effort to explore.

One of this possible scenarios, the implementation of a marketplace business model (applied to a DAM scenario for our example), can be a valuable test scenario to extract some useful requirements for the Triple Space architecture implementation. A technical analysis has been performed to i)define the objectives pursued with the implementation of this model, ii)give an overview of the planned implementation for Tripcom WP8A prototype and iii)provide a set of indicators to finally test the results obtained after the implementation has been performed.

The functional analysis made, and the research of Triple Space based technologies conclude that the implementation of this use case can be a good testing point for Tripcom implementation and the opportunities of Tripcom exploitation.

References

[1] Business Process Execution Language for Web Services, version 1.1, July 2002. Available at: http://www.ibm.com/developerworks/library/ws-bpel.

[2] M. Astley, J. Auerbach, S. Bhola, G. Buttner, M. Kaplan, K. Miller, R. Saccone Jr, R. Strom, D.C. Sturman, M.J. Ward, et al. Achieving Scalability and Throughput in a Publish/Subscribe System. *IBM Research Report RC23103 (W0402-026)*, February 2004.

[3] David Burdett. Avoiding EDI's Mistakes With Web Services. *EAI Journal*, pages 8–11, December 2002.

[4] Dario Cerizza, Davide Cerri, Alessandro Ghioni, Geri Joskowicz, Jacek Kopecky, Daniel Martin, Henar Mu noz, Lyndon Nixon, Noelia Pérez, Thorsten Scheibler, and Daniel Wutke. Security and trust requirement analysis and state-of-the-art. Technical report, Tripcom, 2006.

[5] Dave Chappell. *Enterprise Service Bus*. OReilly, 2004.

[6] Coral Consortium. Coral consortium whitepaper. `http://www.coral-interop.org/main/news/Coral.whitepaper.pdf`, February 2006.

[7] Umeshwar Dayal, Meichun Hsu, and Rivka Ladin. Organizing long-running activities with triggers and transactions. In *SIGMOD '90: Proceedings of the 1990 ACM SIGMOD international conference on Management of data*, pages 204–214, New York, NY, USA, 1990. ACM Press.

[8] Alain Dumas. Select the best approach for your data integration initiatives. *Sunopsis White paper*, 2002.

[9] A. Durán, B. Bernárdez, A. Ruiz, and M. Toro. A requirements elicitation approach based in templates and patterns. In *WER'99 Proceedings, Buenos Aires*, pages 17–29, 1999.

[10] Dieter Fensel. Triple-space computing: Semantic web services based on persistent publication of information. In *In: Proc. of IFIP International Conf. on Intelligence in Communication Systems (Bangkok, Thailand)*, November 2004.

[11] United Nations Economic Commission for Europe. United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport. http://www.unece.org, 2005.

[12] Tele Management Forum. Enhanced Telecom Operations Map (eTOM): the business process framework. Release 5.0, 2005.

[13] Doug Foxvog, Juan Pablo Palacios, and Elena Paslaru Bontas Simperl. Analysis of edifact and other standards. Technical report, Tripcom, 2006.

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

[15] L.M. Haas, E.T. Lin, and M.A. Roth. Data integration through database federation. *IBM Systems Journal*, 41(4):578–596, 2002.

[16] G. Hohpe and B. Woolf. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Professional, 2003.

[17] Gregor Hohpe. Programming without a call stack event-driven architectures. http://www.eaipatterns.com/docs/EDA.pdf.

[18] Gregor Hohpe and Bobby Woolf. *Enterprise Integration Patterns. Designing, Building and Deploying Messaging*. Addison-Wesley Professional, 2003.

[19] G. Larose. DAM and interoperable DRM: Maintaining agility in the world of evolving content ecosystems. *Journal of Digital Asset Management*, 2:17–25, January 2006.

[20] F. Leymann. Space-based Computing and Semantics: A Web Service Purist's Point-Of-View. Technical Report Fakultätsbericht Nr. 2006/05, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, March 2006.

[21] F. Leymann and D. Roller. *Production Workflow*. Prentice Hall PTR, 2002.

[22] Frank Leymann. Web services: Distributed applications without limits. *Proceedings of Database Systems For Business, Technology and Web BTW 2003*, 2003.

[23] D. Linthicum. *Enterprise Application Integration*. Addison-Wesley Professional, 1999.

[24] Martin Murth, Gerson Joskowicz, eva Kühn, Dario Cerizza, Davide Cerri, David de Francisco, Alessandro Ghioni, Daniel Martin, Lyndon Nixon, Nuria Sanchez, Brahmananda Sapkota, Omair Shafiq, and Daniel Wutke. Triple space reference architecture. Technical report, Tripcom, 2007.

[25] OASIS. ebxml technical architecture specification v1.0.4. http://www.ebxml.org/specs/ebTA.pdf, February 2001.

[26] T. Puschmann and R. Alt. Enterprise application integration - the case of the robert bosch group. *hicss*, 09:9047, 2001.

[27] Janne Saarela, Tommi Koivula, Lyndon Nixon, and Axel Polleres. State of the art and triplespace-specific requirements of semantic query languages. Technical report, Tripcom, 2007.

[28] Brahmananda Sapkota, Doug Foxvog, Daniel Wutke, Daniel Martin, Martin Murth, Omair Shafiq, Andrea Turati, Emanuele Della Valle, Nuria Sanchez, and Jacek Kopecky. Architectural integration of triple spaces with web service infrastructures. Technical report, Tripcom, 2007.

[29] Amit P. Sheth, Dimitrios Georgakopoulos, Stef Joosten, Marek Rusinkiewicz, Walt Scacchi, Jack C. Wileden, and Alexander L. Wolf. Report from the NSF workshop on workflow and process automation in information systems. *SIGMOD Record*, 25(4):55–67, 1996.

[30] Elena Paslaru Bontas Simperl, Lyndon J. B. Nixon, Reto Krummenacher, Vassil Momtchev, and Henar Mu noz. Representing rdf semantics in tuples. Technical report, Tripcom, 2006.

[31] T.Berners-Lee, J. Hendler, and O. Lassila. The semanticweb. *Scientific American*, 5:34–43, 2001.

[32] A. Uhl and A. Pommer. *Image And Video Encryption - From Digital Rights Management To Secured Personal Communication*. Springer, 2005.

[33] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson. *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*. Prentice Hall PTR, 2005.

[34] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D.F. Ferguson. *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2005.

[35] Montcheuil Y. and Dupuet C. Third generation etl: Delivering the best performance. *DRM Review*, 2000.

A Data requirements specification

IR01 ***Information about content services***

Associated Requirements :

- IR02.
- IR04.
- IR05.
- IR06.
- IR07.
- IR08.

Purpose : To handle efficiently the service redesign through the definition of all contracts and dependencies within a service.

Description : The Triple Space shall store the information corresponding to existing content services created within the system context. More precisely:

Specific Data :

- Identifier of the content service.
- Identifier of the contract which defines the business relationship which finally offer the service.
- The list of the users using this service.
- The satisfaction level that is derived from users evaluation of the service.
- The umbral satisfaction level that would trigger a re-design process because the user perception of the service turns dangerously negative.

Time interval : Present.

Importance : High.

Urgency High.

Comments :

IR02 ***Information about available contents***

Associated Requirements :

- IR03.
- IR05.

Purpose : To show service providers the available contents offered by content providers which want to trade with these contents.

Description : The Triple Space shall store the information corresponding to contents being offered by created within the system context. More precisely:

Specific Data :

- Title or name of the content.
- The content type.
- The quality of the content in a continuous description translated from discrete factors through the business logic implemented.
- The classification of content with or without advertising.

Time interval : Present.

Importance : High.

Urgency : High.

Comments :

IR03 ***Information about the content types***

Associated Requirements :

- IR:02.

Purpose : Have a classification model for contents depending on their purpose and media type.

Description : The Triple Space shall store the information corresponding to possible content types for the contents being offered. More precisely:

Specific Data :

- Name of the content type, like "Spanish league soccer videos"
- Media type: video, audio or image.
- Media area. For previous example could be sports¿soccer.
- Sub-area. A more precise classification parameter. In previous example would be "Spanish league".

Time interval : Present.

Importance : High.

Urgency : Medium.

Comments :

IR04 ***Information about service providers***

Associated Requirements :

- IR01.
- IR07.
- IR09.
- IR10.

Purpose : To have the relevant information about a service provider so that services designed could be exploited by using this information in a commercial system.

Description : The Triple Space shall store the information corresponding to service providers offering services within the system context. More precisely:

Specific Data :

- Identification of this service provider.
- The name.

Time interval : Present.

Importance : High.

Urgency : High.

Comments :

IR05 ***Information about customers***

Associated Requirements :

- IR01.
- IR08.

Purpose : To have the relevant information about a customer who is using some of the content services assembled within the system.

Description : The Triple Space shall store the information corresponding to using the services offered by the system. More precisely:

Specific Data :

- Identification of the user.
- The name.

Time interval : Present.

Importance : High.

Urgency : High.

Comments :

IR06 ***Information about content providers***

Associated Requirements :

- IR01.
- IR02.
- IR07.
- IR09.
- IR10.

Purpose : To have the relevant information about a content provider so that service providers can contact it to request contents they are interested in offering.

Description : The Triple Space shall store the information corresponding to content providers offering contents within the system context. More precisely:

Specific Data :

- Identification of the content provider.
- Name.

Time interval : Present.

Importance : High.

Urgency : High.

Comments :

IR07 ***Information about contracts***

Associated Requirements :

- IR02.
- IR04.
- IR06.

Purpose : To know all useful information from each product or service contracted by a service provider in order to assemble the content service it plans to offer to its final customers.

Description : The Triple Space shall store the information corresponding to contracts agreed by different DAM actors within the system context. More precisely:

Specific Data :

- Contractor ID (Service Provider).
- Supplier(s) ID(s) (Content provider(s)).
- Content(s) being contracted.
- Amount of money being paid for this contract form contractor to supplier.

Time interval : Present.

Importance : High.

Urgency : High.

Comments :

IR08 ***Information about user evaluations of a service***

Associated Requirements :

- IR01.
- IR05.

Purpose : To provide a mechanism to the users so that they can evaluate the services they are using.

Description : The Triple Space shall store the information corresponding to the evaluation of services performed by customers who are using them. More precisely:

Specific Data :

- The identification of the service being evaluated.
- The evaluation of the service in a measurable unit.

Time interval : Present and past.

Importance : Medium.

Comments :

IR UC09 ***Information about an auction***

Associated Requirements :

- IR03.
- IR04.
- IR06.
- IR10.

Purpose : To perform an auction negotiation process in which a Service Provider asks for a certain type of content and all possible suppliers for this content compete for making the best offer for this Service Provider.

Description : The Triple Space shall store the information corresponding to an auction process. More precisely:

Specific Data :

- The content type the Service Provider is looking for.
- The set of active bids being considered in the auction process.
- The temporal deadline when the auction process expires and no bids are admitted.

Time interval : Present and past for valid bids.

Importance : High.

Urgency : High.

Comments :

IR10 ***Information about a bid***

Associated Requirements :

- IR02.
- IR06.
- IR09.

Purpose : To allow content providers to perform offers that can satisfy the Service Provider objectives in a competitive manner.

Description : The Triple Space shall store the information corresponding to all active bids within an active auction process. More precisely:

Specific Data :

- Identification of the content provider performing the bid.
- The concrete content being offered by this content provider.
- The amount of money the content provider requires for this content being included in the bid.

Time interval : Present (within an active auction).

Importance : High.

Urgency : High.

Comments :

B Detailed functional analysis description

FR UC01 ***Consult a service***

Associated Use Cases :

Description : This use case model three possible actor interactions:

1. The Service Provider responsible for offering a content service wants to retrieve some information for the service (both public or restricted information since it is the responsible of the service). Any other Service Provider might want to consult services as well, but will only have access to public information.
2. Any content provider wants to check some public information regarding any service (in order to perform a market needs report for example) or some restricted information regarding their contracts subscribed in a service they are contributing to offer.
3. Any customer want to consult any available service in order to register to that service.

Precondition :

- There are some services stored in the Triple Space
- The agent searching for this information is registered in the system and thus is a trusted agent.

Sequence :

Step	Action
step 1	A Service Provider or a Content Provider publish in the Triple Space the pattern of the services it is searching for.
step 2	TS locate the node where this kind of services are stored.
step 3	TS gets all tuples which can fit this pattern.
step 4	TS sends the information to requester.

Postcondition : Data stored in TS remains unchanged and so consistent after the service search.

Exceptions :

Step	Action
step 4	If requester has not enough permissions to retrieve any contract information regarding the service being requested, it will be provided only the public information.

Importance : High

Comments : This use case will involve query interactions between actors and the service registry.

RF UC02 ***Assemble a service***

Associated Use Cases :

- UC03: Register service.

Description : A Service Provider creates a new service which will be managed by the Triple Space.

Precondition
- A contract has been defined between a Service Provider and one or many Content Providers.
- All requirements published by the Service Provider when the auction was created have been fulfilled by any Content Providers.

Sequence :

Step	Action
step 1	The Service Provider gets all contracts which are part of the service being assembled.
step 2	The Service Provider adds the preliminary list of customers subscribe to this new service or an empty list if there weren't defined preliminary customers.
step 3	The Service Provider defines the satisfaction evaluation method or algorithm.
step 4	The Service Provider specify the umbral value to fix the automatic redesign of the service.

Postcondition : If the service was successfully created, it will be registered in the service register.

Exceptions : Exceptions defined in the business logic if any.

Importance : High.

Comments : This use case takes place under the business logic definition in the Service Provider.

RF UC03 ***Register Service***

Associated Use Cases :

- UC02: Assemble a service.

Description : Covers the registration of a new assembled service in the Triple Space. This way, the service will be available for customers to subscribe.

Precondition :

- The agent searching for this information is registered in the system and thus is a trusted agent.
- A new content service has been successfully created.

Sequence :

Step	Action
step 1	Service Provider sends information which describes the content service which has recently created.
step 2	The Triple Space ensures the description of the service has all information about the service created.
step 3	The service is stored in the service registry with status "Ready" and the Triple Space notifies the Service Provider both the successful creation and state assigned .

Postcondition :

- The service is included in the Triple Space and it is possible to subscribe to it.

Exceptions :

Step	Action
step 2	If any of the contract references or feature definition is missing in the service description, the Triple Space will notify the Service Provider and ask for missing information. Nothing will be stored in the service register.

Importance : High.

Comments : No comments.

RF UC04 ***Call and auction***

Associated Use Cases :

- UC05: Notify participants
- UC06: Change service features

Description : A Service Provider starts an auction process in order to get the desired content types it is demanding to offer a service.

Precondition :

- The agent searching for this information is registered in the system and thus is a trusted agent.
- There is no active auction with the same parameters for the same Service Provider.

Sequence :

Step	Action
step 1	The Service Provider sends a message to the TS expressing its wish to start an auction process and specifying both the desired content type(s) and temporal deadline.
step 2	The TS creates a new virtual space for this auction given that preconditions are fulfilled.
step 3	*UC05: Notify participants.*
step 4	After temporal deadline expires a ranked list of bids is returned to the caller Service Provider.
step 5	The TS removes the virtual space corresponding to the finished auction.

Postcondition :

- If the auction process was successful, the caller Service Providers gets a ranked list of bids performed to fulfill its content needs.

Exceptions :

Step	Action
step 1	If not all the information needed to star an auction process is provided, the TS notifies to the Service Provider and asks for missing information.
step 3	Those related with UC05.
step 4	If no active bids are available at this moment a special message to the Service Provider will be delivered, giving the chance to make auction longer or finish the auction with no results.

Importance : High.

Comments : Central use case for the auction negotiation model.

RF UC05 ***Notify participants***

Associated Use Cases :

- UC04: Call an auction

Description : Notifies the potential supporters for a Service Provider need that an auction process is going to start, enrolling them in the auction process.

Precondition :

- The desired content types are defined.

Sequence :

Step	Action
step 1	A Triple Space queries all suitable Content Providers to offer the content types requested by a Service Provider when calling and auction.
step 2	The TS checks if some of these potential suppliers are restricted to the because users were previously unsatisfied with contents offered.
step 3	The TS notifies to all suitable Content Providers who are not restricted in this auction being called.
Step 4	The TS enrolls these Content Providers to the auction being created so that the access to the auction is granted for them.

Postcondition :

- All potential suppliers for a Service Provider request are notified and their access to the auction granted.

Exceptions :

Step	Action
Step 3	if notification could not be given, it should be tried again later meanwhile the auction is being created anyway.

Importance : High.

Comments : Intermediate step to start an auction process.

RF UC6 ***Change service features***

Associated Use Cases :

- UC04: Call an auction
- UC07: Break a contract
- UC08: Subscribe to a service
- UC09: Evaluate a service

Description : Describes a modification of a service registered in the Triple Space.

Precondition :

- The service being modified was previously registered in the service register.
- There is a modification wish either by the Service Provider, by any of its suppliers or as a consequence of a customer action (new users subscribed or negative evaluation of the service).

Sequence :

Step	Action
step 1	Either the Service Provider, a Content Provider or a user interaction want to change some of the parameters of a service it is involved, and send a message to the TS notifying this issue.
step 2	Security layer checks that requester is involved in the service offering and has enough permissions to request the service notification in the case the request come from a Service Provider or a Content Provider.
step 3a	If a new user is being added it modifies the service specification by adding a new user to the list.
step 3b	If the Service Provider or Content Provider offering the service wants to change some service parameters which don't suppose a contract break, a confirmation for the other part in the signing contract is asked and UC11 takes place again, replacing the older contract by the new one.

Postcondition
- Service changed will have all definition information after the service modifications.

Exceptions :

Step	Action
step 2	If requester does not have the necessary permissions to request a change of the service, a notification will be delivered and no request will be sent by the Triple Space. Also request won't be stored in the Triple Space.
step 3b	If notification can't be delivered, the Triple Space will try to send it again and the request will keep stored meanwhile.
step 3b	Has exceptions derived from UC11 use case.

Importance : High.

Comments : This use case comprises voluntary changes meanwhile UC07 comprises changes which derive in service redesign.

RF UC07 ***Break a contract***

Associated Use Cases :

- UC06: Change service features
- UC10: Halt an active service

Description : Describes the breaking process of a signed contract, due to the contractor wish, the supplier wish or an event come from a negative evaluation by users.

Precondition :

- The contract being broken is stored in the TS and defines some of the active services.

Sequence :

Step	Action
step 1	If this request come from any contract part, the TS validates that this part has enough permissions to break the contract.
step 2	Contract is deleted from the TS and all parts involved in the broken contract notified by the TS.
step 3	UC10 takes place.

Postcondition : Broken contract is no longer stored in the TS.

Exceptions :

Step	Action
Step 1	if requester is not part of the contract use case is interrupted and requester notified.
Step 2	if notification could not be delivered, the contract is kept inside the TS as active and notification is sent again until it reaches and contract can be deleted.
Step 3	those from UC09.

Importance : High.

Comments : Used to break ad-hoc business relationships in a secure manner.

RF UC08 ***Subscribe to a service***

Associated Use Cases :

- UC06: Change service features
- UC15: Actor identification

Description : Describes the subscription of a customer to any active services being offered from the service register.

Precondition :

- The user has access to the system and is a trusted user.
- The service to which the user is subscribing is stored in the TS and active.

Sequence :

Step	Action
step 1	UC15 takes place.
step 2	The user selects and active service from the service register and notifies the TS his intention to subscribe to this service.
step 3	The TS notifies the responsible Service Provider about this new subscription.
step 4	UC06 takes place.

Postcondition : Service definition is modified according to the new subscription received and Service Provider is notified about the modification.

Exceptions :

Step	Action
step 1	those from UC15.
step 3	if notification could not be given, it should be tried again later and the subscription managed anyway.
step 4	those from UC06.

Importance : High.

Comments : This functionality is used to attach users to services so that they can use and evaluate them.

RF UC09 ***Evaluate a service***

Associated Use Cases :

- UC06: Change service features

Description : Models the feedback provision from users to services being used by them.

Precondition :

- The service being evaluated is used by the customer.
- Customer is subscribed to this service.

Sequence :

Step	Action
step 1	A customer send an evaluation report to the TS.
step 2	The TS merges the evaluation from this user with previous evaluations from other used already registered in the service definition.
step 3	UC06 takes place.
step 4	If new satisfaction level reaches the umbral value defined by the Service Provider within the service definition, use case UC07 takes place.

Postcondition :

- The user feedback is merged with previous feedback and stored in the TS.

Exceptions :

Step	Action
step 3	those from UC06.
step 4	those from UC07 in case it takes place.

Importance : Medium.

Comments : The definition of the aforementioned umbral value is made by the Service Provider in its business rules.

RF UC10 ***Disable an active service***

Associated Use Cases :

- UC07: Break a contract

Description : Once a contract is broken the service has to be redesign. This use case models the needed interaction to disable a service which is no longer valid in order to start the redesign process.

Precondition :

- The service has to be disabled.
- The service selected to be disabled is an active service.

Sequence :

Step	Action
step 1	The TS detects what are the content(s) needed in order to restore the service function.
step 2	Users of the service are notified that this service will be temporary down and give them the estimated recovery time.
step 3	The TS marks the service inactive and calls the appropriate web services to stop in a secured way the service.

Postcondition :

- The service is disabled.
- Users are aware of the situation.

Exceptions :

Step	Action
step 2	If notification fails it will be tried again but service disabling process will continue anyway.

Importance : High.

Comments : Has to connect with automatic web service invocation to disable the service by orchestrating dependencies.

RF UC11 *Sign a contract*

Associated Use Cases :

- UC02: Assemble a service

Description : Covers the creation of a business relationship between a Service Provider and one Content Provider to hire a content.

Precondition :

- Service Provider has selected one of the bids made by the Content Provider as the best one in a recently past auction.

Sequence :

Step	Action
step 1	The TS notifies the Content Provider that the offer it sent to the active auction was selected, indicating the offer and the auction data, and request the contact data and legal issues from this Content Provider.
step 2	The Content Provider sends the requested data for this TS request and the TS validate the identity and the information.
step 3	The TS asks for the same data to the Service Provider and merges in the contract.
step 4	The Service Provider send the information and both the identity and the information is validated.
step 5	The TS merges this information and stores it attached to the service definition.

Postcondition :

- A contract is defined and it's attached to the corresponding service.

Exceptions :

Step	Action
step 1	If notification fails TS will try again and wait until Content Provider is aware of the contract formalization.
step 2	If data is not validated or even provided, the formalization of the contract is canceled, next offer is considered and Content Provider is not considered for further auctions.
step 3	If notification fails TS will try again and wait until Service Provider is aware of the contract formalization.
step 2	If any of the contract references or feature definition is missing in the service description, the Triple Space will notify the Content Provider and cancel the signature.

Importance : High.

Comments : No comments.

RF UC12 ***Perform a bid***

Associated Use Cases :

- UC15: Actor identification

Description : Describes the functionality of performing a bid inside an active auction from a Content Provider perspective.

Precondition :

- The Content Provider is part of the auction.
- The auction is still active.

Sequence :

Step	Action
step 1	The Content Provider send a bid to the Triple Space.
step 2	The Triple Space checks the auction is active and this Content Provider is registered in the auction.
step 3	UC15 is executed.
step 4	If UC15 returned a correct validation, the TS adds the bid information to the virtual space the auction is placed in.

Postcondition

- If the bid is performed correctly it is add to the bids pull.

Exceptions :

Step	Action
step 2	If the auction is no longer active or the Content Provider is not registered in the auction a message is sent to the Content Provider and the bid is not take into account.
step 3	Those from UC15.

Importance : High.

Comments : Critical use case in the auction negotiation process.

RF UC13 ***Consult existing bids***

Associated Use Cases :

Description : Describes how a Content Provider registered to an active auction process can consult existing bids in order to perform a better bid to the auction.

Precondition :

- The requester Content Provider is registered in the auction and is a trusted agent by the system.

Sequence :

Step	Action
step 1	The Content Provider send a query to the Triple Space.
step 2	The Triple Space checks the auction is active and this Content Provider is registered in the auction.
step 3	UC15 is executed.
step 4	The TS returns the bids requested by the Content Provider.

Postcondition :

- Content Provider gets the desired information if it has access to the information.

Exceptions :

Step	Action
step 2	If the auction is no longer active or the Content Provider is not registered in the auction a message is sent to the Content Provider and the bid is not take into account.
step 3	Those from UC15.

Importance : Medium.

Comments : This mechanism provides the Content Providers the chance to see active bids and how they could improve the best offer performed under their own point of view (which may be different for the Service Provider since it depends on the business rules).

RF UC14 ***Publish a content offer***

Associated Use Cases :

- UC18: Actor identification
- UC16: Provide content catalogue info

Description : A Content Provider make some content offers or modify previous content offers registered in the content catalogue. The information being published in the content catalogue is the public information can be given from the contents the Content Provider wants to trade with. Private information about the content will remain kept in the content catalogue systems and will only be provided if asked by any interested Service Provider.

Condition :

- The Content Provider has access to the system and is a trusted agent.
- The information provided by the content provider where some content parameter is modified assumes that content being modified was previously registered in the content catalogue of the TS.

Sequence :

Step	Action
step 1	Content Provider sends a message to the TS specifying a set of content offers which can be new one or modification of existing ones (even deletion which will be treated as a special modification in this context).
step 2	Security layer checks that CP has permissions for modifying contents referred in previous message.
step 3	TS adds new registers to the content catalogue if new offers has been sent.
step 4	TS modifies and/or deletes offers changed in the catalogue sent by the CP

Postcondition .

- New offers described become available to Service Providers.
- Content offers defined to be modified by the CP get their new values updated.

Exceptions :

Step	Action
step 2	If the CP has not permissions to modify referenced contents, it will received a notification from the TS and content parameters will remain unchanged.

Importance : High.

Comments : Content offers only contain public information from CP.

RF UC15 ***Actor identification***

Associated Use Cases :

- Potentially all

Description : Describes the functionality of an actor identification required by the system or other actor involved in a communication.

Precondition :

- Either the system or an actor has requested an actor to be identified.

Sequence :

Step	Action
step 1	The actor is requested to provide an identification by the TS by sending a notification.
step 2	Actor to be identified sends the identification information to the TS.
step 3a	If the TS itself needed this information to decide on an access policy, it evaluates the identification information and decides, notifying the actor if its rejecting and allowing communication otherwise.
step 3b	If this identification was requested by another actor, the TS will forward the information to the requester actor

Postcondition :

- If actor being identified is not allowed to obtain data or maintain the communication, it will be notified by the system.

Exceptions :

Step	Action
step 3a	If refusal notification could not be delivered, the TS must keep on trying to perform this notification, and the communication won't be allowed anyway.

Importance : High.

Comments : Important from the security architecture perspective.