



TripCom
Triple Space Communication

FP6 – 027324

Deliverable

D8A.3
EAI Validation and Recommendations

David de Francisco Marcos
Guillermo López Reyes
Germán Toro del Valle

April 8, 2009

EXECUTIVE SUMMARY

The EAI Validation and Recommendations deliverable aims at collecting the report on the latest implementations done for the digital asset marketplace prototype, as well as to present the ontology validation conducted. The functional evaluation of the use case was performed in D8A.2 [3], and the scalability evaluation and conclusions from the use case perspective are part of D8.5v2 [5]. The reader should therefore refer to these deliverables for a complete information about validation of the EAI prototype. Additionally, the deliverable title suggests recommendations to be drawn from this deliverable. Since most of these recommendations are related to the use of the TripCom architecture with respect to its application for a marketplace implementation, we included these recommendations as part of D6.5v2, in the architecture evaluation.

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.3	Title	EAI Validation and Recommendations
Work Package	Number	8A	Title	Use Case 1 - Enterprise Application Integration

Date of Delivery	Contractual	M36	Actual	31-March-09
Status	version 0.10		final	<input type="checkbox"/>
Nature	prototype <input type="checkbox"/> report <input checked="" type="checkbox"/> dissemination <input type="checkbox"/>			
Dissemination Level	public <input checked="" type="checkbox"/> consortium <input type="checkbox"/>			

Authors (Partner)	David de Francisco Marcos (TID), Guillermo López Reyes (TID) and Germán Toro del Valle (TID)		
Resp. Author	David de Francisco Marcos (TID)		E-mail davidfr@tid.es
	Partner	TID	Phone +34(983) 367587

Abstract (for dissemination)	<p>Triplespaces, as a semantic middleware implementation, provide a scalable and added value platform to build a marketplace on its top. This deliverable aims at presenting the validation activities carried out for the marketplace implementation, as well as to present a customer centered application that will benefit from the semantic publication of digital content services, in the form of recommendation and linked data advantages.</p>
Keywords	Enterprise Application Integration (EAI), implementation, prototype, Digital Asset Management (DAM), Space Based Computing, Triple Spaces, Marketplace Business Model, Enterprise Integration Patterns

Version Log			
Issue Date	Rev No.	Author	Change
2009-03-05	1	David de Francisco	TOC defined.
2009-03-12	2	David de Francisco	Intro and UML diagrams defined.
2009-03-16	3	Guillermo López	Analysis and ontology evaluation part defined.
2009-03-19	4	David de Francisco and Germán Toro	User Interface definition completed.
2009-03-29	5	David de Francisco and Guillermo López	Deliverable integration.
2009-03-31	6	Germán Toro	Deliverable review.
2009-04-08	7	Dario Cerizza and David de Francisco	Peer review done and comments integrated in the document.

PROJECT CONSORTIUM INFORMATION

Acronym	Partner	Contact
Semantic Technology Institute Innsbruck http://www.sti-innsbruck.at	STI  STI · INNSBRUCK	Prof. Dr. Dieter Fensel Semantic Technology Institute (STI) Innsbruck, Austria E-mail: dieter.fensel@sti-innsbruck.at
National University of Ireland, Galway http://www.deri.ie	NUIG  National University of Ireland, Galway Ollscoil na hÉireann, Galway	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  Universität Stuttgart	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  TECHNISCHE UNIVERSITÄT WIEN VIENNA UNIVERSITY OF TECHNOLOGY	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  Freie Universität Berlin	Prof. Dr.-Ing. Robert Tolksdorf AG Netzbaasierte Informationssysteme Berlin, Germany E-mail : tolk@inf.fu-berlin.de
Ontotext Lab, Sirma Group Corp. http://www.ontotext.com/	ONTO  Ontotext Knowledge and Language Engineering Lab of Sirma	Atanas Kiryakov, Vassil Momtchev, Ontotext Lab, Sirma Group Corp. Sofia, Bulgaria E-mail: vassil.momtchev@ontotext.com
Profium OY http://www.profium.com/	Profium  profium	Dr. Janne Saarela Profium OY Espoo, Finland E-mail: janne.saarela@profium.com
CEFRIEL SCRL. http://www.cefriel.it/	CEFRIEL  CEFRIEL FORGING INNOVATION KNOWLEDGE	Davide Cerri CEFRIEL SCRL. Milano, Italy E-mail: cerri@cefriel.it
Telefonica I+D http://www.tid.es/	TID  Telefonica TELEFÓNICA INVESTIGACIÓN Y DESARROLLO	Noelia Pérez Crespo Telefonica I+D Madrid, España E-mail: npc@tid.es

TABLE OF CONTENTS

1	INTRODUCTION	2
2	IMPLEMENTATION OF THE DAM CUSTOMER APPLICATION	3
3	DEFINITION OF THE VALIDATION TESTS	5
3.1	Ontology evaluation tests	5
3.1.1	Content Related Competency Questions	5
3.1.2	Auction Management Related Competency Questions	6
3.1.3	Business Related Competency Questions	6
3.1.4	Catalog and Evaluation Related Competency Questions	6
3.2	Further Uses of Semantics in the EAI Use Case: Reasoning	7
3.2.1	Recommendation	7
3.2.2	Matching Product Capabilities	7
4	ANALYSIS OF THE DAM CUSTOMER APPLICATION	9
4.1	Analysis of the Customer Application	9
4.2	User Interface Design	11
4.2.1	Definition of Gadgets	12
4.3	Composition of Views	14
5	CONCLUSIONS	17

LIST OF ABBREVIATIONS

API	Application Programming Interface
DAM	Digital Asset Management
EAI	Enterprise Application Integration
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
TS	Triple Space
UML	Unified Modeling Language
URI	Universal Resource Identifier
WP	Work Package
XML	eXtended Markup Language

1 INTRODUCTION

This deliverable aims at reporting the last developments made in the context of the EAI use case prototype: the multimedia asset marketplace. D8A.2 showed the prototype implementation focused on the marketplace perspective [2], that is, managing the interactions between the different partners collaborating in bundling digital content services to be offered to final customers. These interactions allowed different content providers to offer their digital contents following a marketplace like scheme, and service providers to purchase licenses to offer these contents as part of their services by calling auctions.

The missing part of the scenario described in [1] is the customer side of the prototype described. This deliverable defines the developments carried out to outcome the missing customer functionality of the system. Given that the marketplace logic has a different scope, it has been decided to implement the customer application (which enables customers to subscribe and enjoy the services offered by services providers with the help of our previous marketplace implementation) as a separate application. This decision has been motivated by two facts. First, to decouple both applications, so that as a future use different modules could be used together rather than these implemented in TripCom. Second, and more important, to integrate the user interface to the customer into a mash up platform, ezWeb ¹, which is widely used to present services to Telefónica.

Apart from the flexibility offered by mash up platforms in terms of user interaction and data interlinking, many Telefónica prototypes have been built using this platform lately. Adopting this user interface platform has thus allowed us to integrate our DAM prototype into joint exploitation initiatives, covered under the same user interface umbrella, and therefore being more familiar and acceptable to the business attendees to these exploitation demos.

¹ezWeb is a Spanish government funded research project. Information and code is available at <http://ezweb.morfeo-project.org/lng/en>

2 IMPLEMENTATION OF THE DAM CUSTOMER APPLICATION

This section outlines the implementation details of the DAM customer application. An architectural design is provided in Figure 2.1 for a better understanding of the work done. The figure depicts the components used and developed, and the physical deployment of the architecture. For the prototype purposes, the application has been deployed in two different servers, one for the user interface (mufflon.hi.inet), and another for the Triplespace and business logic deployment (lince.hi.inet). We will now describe the components from the lowest to the upper most.

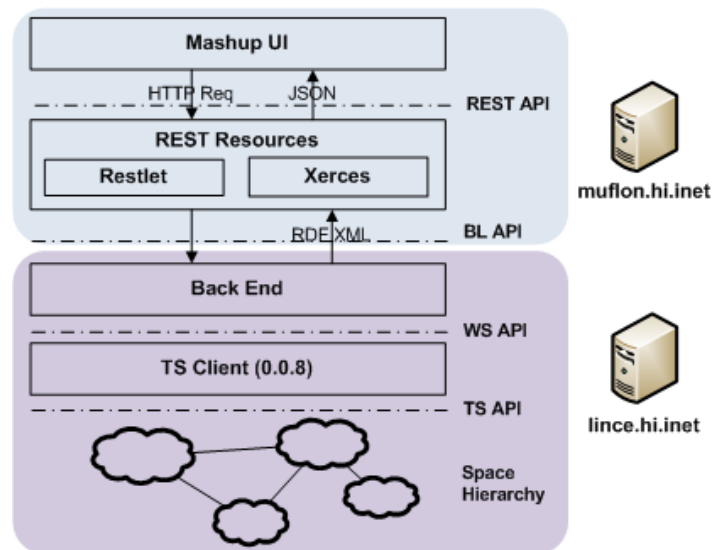


Figure 2.1: Architecture Design of the DAM Customer Client

The space hierarchy is deployed in a single kernel for prototype purposes. The space hierarchy can be seen in Figure 2.2. The called Marketplace space is the root space in the hierarchy. Direct sons of this space are the Customer, Evaluation and what has been called in a general way Service Catalog Spaces (but that doesn't mean that it includes only the service catalogs information). The Customer space gathers the information related to the clients who participate in the DAM marketplace process, from personal data to the subscriptions made to services. The Evaluations space was included in order to gather all the evaluations made to the different services, assets and media works. The generally called Service Provider space is divided in several sub-spaces which will gather the information of Service Catalogs, Services, Assets and all the information relating the media works from each Service Provider. As an example, in figure 2.2 three of them are shown.

The latest version of the TripCom kernel is deployed in the same machine. The spaces store the RDF triples according to the latest release of the DAM ontology developed ¹. An Apache Tomcat is also deployed to offer the TS Client SOAP API which is used by the business logic of the application.

The box labeled as "Back End" contains the business logic of the DAM customer application. This logic involves basically the information exchange between the cus-

¹Refer to <http://www.tripcom.org/ontologies/dam.php> to check the latest version of the DAM ontology.

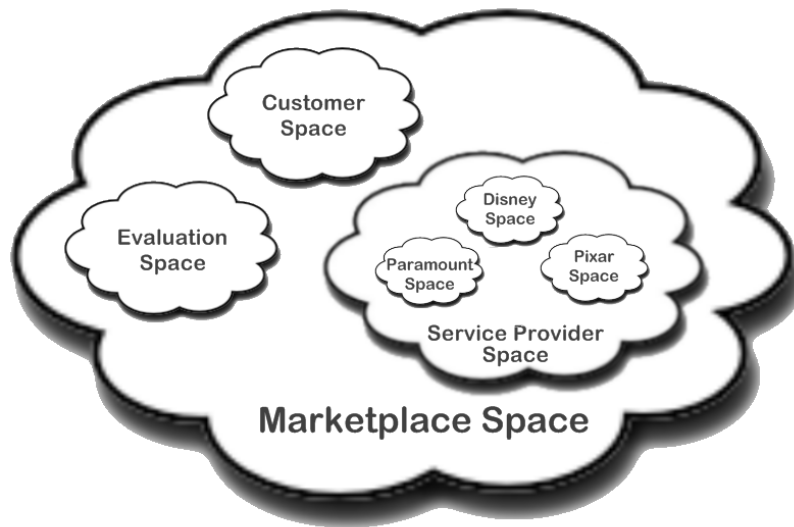


Figure 2.2: Space hierarchy of the DAM Customer Client

tomer and the Triplespace infrastructure, managing the information storage and query composition from the data provided by the user interface.

The aforementioned business logic is accessed from the REST resources, which has been developed using the Restlet framework and deployed in the user interface server. These resources expose, as URIs, the different kind of information that will be shown by the gadgets in the user interface (i.e. service lists, asset list, evaluation of services or assets, etc.). REST resources are accessed by the user interface every time a user command an action. REST resources invoke the business logic offered as a Java API by the backend component, providing information to query information which is stored in the Triplespace kernel, and receives this information as RDF XML. This XML is parsed using Xerces, and translated into JSON notation to be used by the gadgets in the user interface.

As we commented in the introduction, the user interaction is performed in a free mash-up platform, ezWeb (See <http://ezweb.morfeo-project.org/lng/en>), which enables the design of a customizable and interoperable user interface. The information is shown as interconnected gadgets, which make use of the JSON information provided by the REST resources and make HTTP requests to these according to the user interactions.

3 DEFINITION OF THE VALIDATION TESTS

The validation of the EAI use case has been performed in three steps:

1. The functional validation of the prototype, which was performed in the context of the deliverable D8A.2 [3].
2. The scalability evaluation, which has been integrated in D6.5v2 [5]. Scalability tests, results and analysis of these results are covered in the aforementioned deliverable, as part of the architecture WP.
3. The ontology validation. The second version of the DAM ontology was presented in D8A.2 [3] as well, and is evaluated in this section.

The scope of the ontology was defined and tested by the means of a series of competency questions in a concrete scenario, and these should be correctly answered when querying the semantic information modeled by this ontology. To populate the ontology, real films information was used, instantiating large quantities of data. However, in order to accomplish the validation tests, a small sub-joint of digital assets was used, in order to have a controlled number of triples in the tests.

3.1 Ontology evaluation tests

The identification of the requirements of the ontology has derived from the particular problems that arise in particular cases (a motivating scenario), and expressed as competency questions [4]. The competency questions have been used not only to define the requirements of the ontology, but also as a method to evaluate the design of the ontology. The results returned by the ontology when asked this competency questions have been envisioned to indicate the quality of the design, and possible malfunctioning or errors. In this sense, the following competency questions have been translated to SPARQL [6] and executed against a test data set. However, due to the concrete range of the developed demo, some aspects of the ontology were not instantiated and, therefore, not all the competency questions verified.

3.1.1 Content Related Competency Questions

1. Return me the offered films with genre comedy and adventure.
2. Return me the available films directed-produced / books written-edited by X, who are awarded with an oscar / nobel.
3. Return me available films in which X but not Y acted.
4. Return me any offer package containing this film X.
5. Return me a list of reggae songs awarded with a grammy.
6. Return me the song whose title I don't remember, but that is singed by Juanes, with the collaboration of Nelly Furtado.
7. Return me the songs of the solo career of the singer of The Police (the songs which do not belong to The Police).

8. Return me the groups or singers who played the official sound track of the film X.
9. Return me the action serials released in 2006 in English.
10. Return me titles of films that could be similar to *The Godfather*, which I liked so much’.

3.1.2 Auction Management Related Competency Questions

11. Give me a list of all active auctions negotiating for this film or kind of film.
12. List me all content providers participating in this auction who have submitted a valid bid last hour.
13. Return me all Content Providers I should notify their bid were beaten (all CPs who made a valid bid which is not the winning bid).
14. Notify me all contents taking part in a bid which have been offered by a bid. Imagine I publish a template searching for crime films and CPs offer me several films. There will be N auction threads each of them with a winning bid for each content that matches it is a crime film and SP will choose later.
15. Return me the final results of the auction (content offered by bids and the winning bid for each).
16. Return me the remaining time for the auction to conclude.
17. Is this content X being offered in the bid I am participating in?
18. Return me the current winning bid for this content.

3.1.3 Business Related Competency Questions

19. Return me the postal address of a client / partner.
20. Return me the owner of the distribution rights of a product.
21. Tell me who the signers of the contract who agrees the use of a determined product are.
22. Tell me the bank and bank account number associated to a credit card.
23. Return me the clients who have purchased the rights to distribute a concrete service.
24. Tell me which is the currency for a Product Offer contract.

3.1.4 Catalog and Evaluation Related Competency Questions

25. Tell me which are the catalogs published by a concrete Service Provider.
26. Give me all the service offers of a concrete Service Provider.
27. Return me the evaluation result of a content.

3.2 Further Uses of Semantics in the EAI Use Case: Reasoning

The use of semantics in the EAI use case has provided a clear added-value to the development of the prototype, but defining a very expressive information model for the use case has never been the main concern of the WP8A work. However, during most of the industrial dissemination activities carried out with Telefónica's business staff, there seems to be an increasing concern about the benefits of reasoning in the design of knowledge based applications. This fact has motivated the definition of some rules within the prototype in order to motivate the benefits of semantic enriched data. These rules are shared in this deliverable using a TRREE and Jena reasoner syntax respectively, as future steps for the use case exploitation. The definition of these rules does not mean that the reasoning capabilities needed to be executed will be granted by the TripCom infrastructure. The demonstration of reasoning capabilities could be made with the integration of a reasoner in the Back-End component of the EAI use case, in case the TripCom infrastructure doesn't support the reasoning capabilities required by these rules.

3.2.1 Recommendation

Next code defines a rule with the aim of recommending similar contents for a Service Provider:

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

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

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

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

3.2.2 Matching Product Capabilities

One added value that has proven to be very interesting for application designers is the definition of business logic in a decoupled manner using rules. Next example shows an example of a definition of playable contents in an iPhone, using Jena syntax for defining rules. In this example we assume that an iPhone can play any video with VCD format but no more resolution than 300x300, or any video with DVD format:

```
//VCD assets with a resolution of 300x300 or less can be played in the iPhone
[iPhone1: (?asset rdf:type dam:VideoAsset), (?asset dam:videoFormat
"VCD"), (?asset dam:horizontalResolution ?horizontal), (?asset
dam:verticalResolution ?vertical), le(?horizontal, 300),
le(?vertical, 300), -> (?asset rdf:type sp:IPhoneSuitableAsset )]
```

```
//DVD format assets can be played in the iPhone, no matter the resolution
[iPhone2: (?asset rdf:type dam:VideoAsset), (?asset dam:videoFormat
"DVD") -> (?asset rdf:type sp:IPhoneSuitableAsset )]
```

Next SPARQL query (prefixes are omitted) could retrieve contents suitable to be played in an iPhone, according to our previous rule:

```
SELECT DISTINCT ?provider WHERE { asset rdf:type
sp:IPhoneSuitableAsset . ?asset dam:isListedInContentCatalog
?catalog . ?provider dam:publishesContentCatalog ?catalog .}
```

The definition of which contents are playable in an iPhone ensures that if next generation of iPhone can play different formats, or current format changes, the re-definition of previous rules would keep our system updated and running without the need of a business logic or data base redesign.

4 ANALYSIS OF THE DAM CUSTOMER APPLICATION

During the first year of the EAI use case, a digital content marketplace application was defined. This application was composed by two different main scenarios. First, the marketplace interaction between several content providers, which wanted to offer the media assets to service providers so that digital content services could be bundled. Second, the use of such services by final customers. In both cases, the telco operator (i.e. Telefónica), has been seen as a mediator in the business process value chain. While during the first scenario was the marketplace provider, in the offering process of digital media to customers the operator could play an added value business services provider (e.g: billing process, customer management, QoS and rating, etc.).

The use case prototype development activities have been devoted to the latter scenario during the final year of the project, and thus this report provides detailed information about the implementations carried out during this period. It is important to remark that it is out of the scope of this implementation task to provide a fully operational and commercial version of a digital content service Web portal, but rather to demonstrate how a TripCom based architecture can be fully integrated in a complete prototype that covers all the aspects of a commercial application, and present this prototype as part of exploitation activities to be continued after the project's end.

This section outlines the basic requirements taken into account to implement the customer side of our prototype, as well as a brief analysis of the implemented application. Given that this development is customer oriented, a more thorough user interface design has been performed, and its results are presented as well.

4.1 Analysis of the Customer Application

The Customer Application **MUST** be able to allow an anonymous client to navigate throughout all the service providers and its services catalogs, services and assets offered. The clients also **MUST** be able to be shown the contents of a determined container, i.e. the assets contained in a service or the services contained in a service catalog or owned by a service provider. They **MUST** be able to know which is the media work digitalized by a concrete asset, and the information relating it. They **MAY** have the possibility of being offered an URL with information relating the media work, or even with a trailer of the media work offered. They **SHOULD** have the option to know which are the services which include a determined asset, and the evaluations to media works, assets and services given by other users. The client **SHOULD** be able to use the application to search for desired media works which are available, and **MAY** have the chance of doing a search by type of digitalization. The application **MAY** also offer a set of recommended items based on the selections made by the user, by the means of the semantic background offered by TripCom.

The application **MUST** offer a way for anonymous customers to register and then, if desired, subscribe to a determined service, and also to cancel any subscription previously made. When registered and logged in, the client **MUST** be able to know which are the services he is subscribed to, and **SHOULD** be able to complete an evaluation about his subscribed services, assets or media works. The application **MAY** have an online player to allow customers watching the film directly on the web browser, but, anyway, when subscribed to a service, the client **MUST** be able to watch the hired products, whether on the web browser or at a local one after downloading it.

In order to support all these high-level operations, specially the navigating and exploring one, the final application MUST be able to retrieve all the information about service providers, catalogs, services or assets at once, or a sub-joint of it depending on the information desired (e.g. getting all available services vs getting all the services offered by a concrete service provider).

The use cases diagram which represents the customer application workings is shown in figure 4.1. For an anonymous user, it shows that a non-logged-in user would be able to navigate and explore all the information offered by the service providers, or even searching for a concrete piece of information by introducing some searching parameters. When this happens, a semantic search on the TripCom kernel will be launched to return the desired results. In case the user wants to subscribe to a service, he would have to be logged-in to be identified. When being a first-time user, it will be necessary to fill a registration form first. Obviously, any registered user will be able to log-in to access his subscribed services.

When logged-in, a user will be capable of navigating and exploring all his subscribed services (and its assets and media works). He will be offered the possibility of evaluating all his subscribed services, assets or media works, and also watching the selected asset, whether online or locally downloading it first.

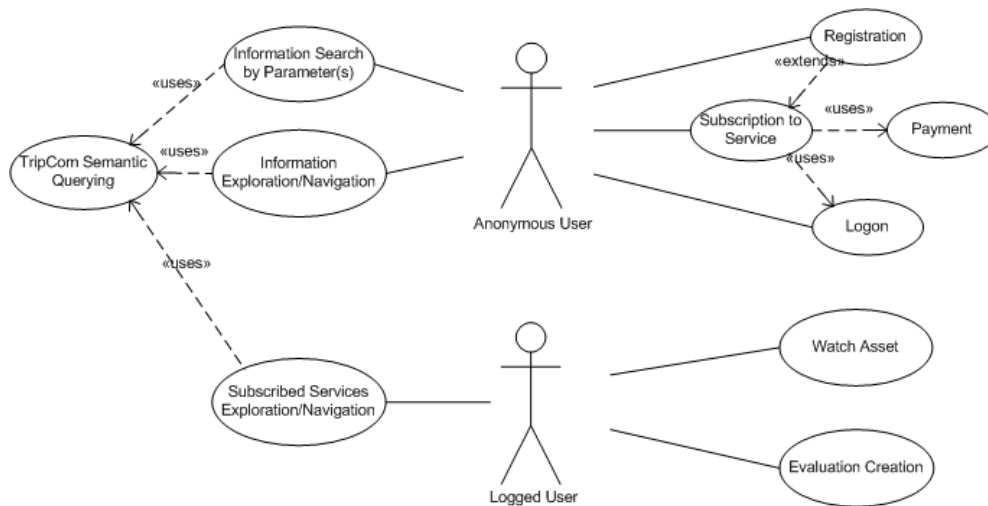


Figure 4.1: Use Cases Diagram

The navigation/exploration issue for a non-registered customer is depicted in the sequence diagram shown in figure 4.2. When a client connects, the `onload()` operation retrieves all available information (service providers, catalogs, services and assets offered) for the client being able to navigate them. When clicking on a concrete service provider, all the service catalogs offered by it will be displayed and, recursively, all the services and assets contained in those catalogs. When clicking on a concrete service catalog, all the services included in it will be shown, as well as the assets contained in those services. Similarly, clicking on a concrete service will display only those assets gathered by this service. Finally, when a concrete asset is selected, information relating the media work digitalized by the asset is displayed, as well as the services which contain that asset, to allow the user to subscribe to one of them if desired. Any time a concrete item is selected, a series of evaluations (to services, to assets or to media works, depending on the item selected) made by other users will be displayed, as well

as semantically inferred recommendations of other items similar to the chosen ones, and therefore may like the user.

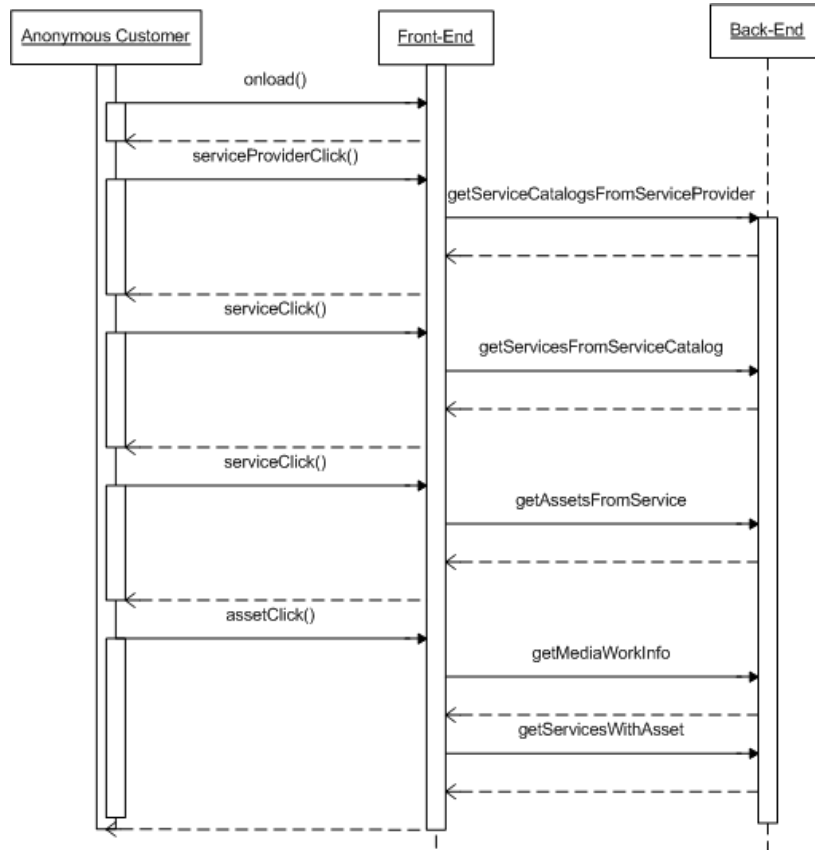


Figure 4.2: Sequence Diagram for the information navigation issue for an anonymous user

A sequence diagram representing the subscription of a registered (not-first-time) user to a service is depicted in figure 4.3. When clicking subscribe, the login menu is deployed. A first-time user would have to register, but an already registered one will only have to login. When doing so, the customer information will be retrieved from TripCom, and the payment form shown to the customer. When clicking again to subscribe, the payment will be effective, and the subscriptions updated.

In figure 4.4 a sequence diagram representing the information exploration by a logged customer is shown. The `onload()` operation will retrieve and show all the services (and its assets) which the customer is subscribed to, and, as it happened in the anonymous user case, when clicking on a concrete service only the assets contained in that service will be visible, and when selecting a concrete asset the digitalized media work information will be presented. Additionally, the asset will be played if desired.

4.2 User Interface Design

As it was previously motivated, the customer prototype application has a stronger focus on the user interface and user satisfiability. This goal has been achieved by the employment of a mashup platform which enables a dynamic configuration of the user interface, as well as the definition of dependencies between the gadgets which present

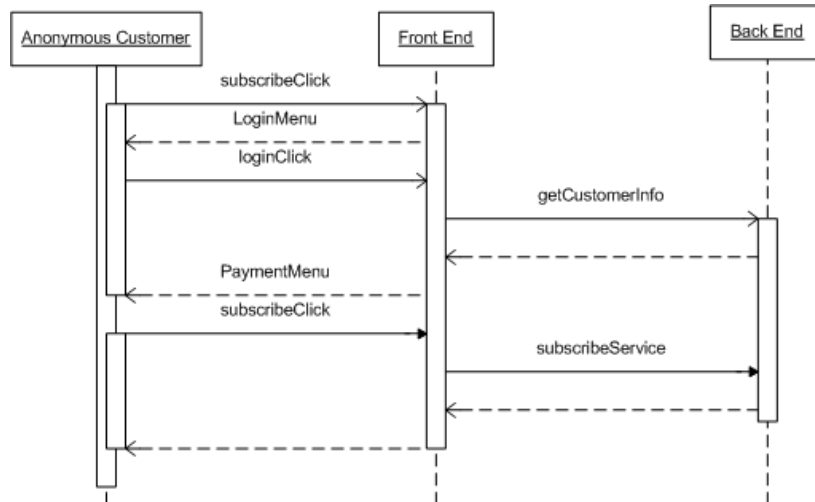


Figure 4.3: Sequence Diagram for the customer subscription to a service

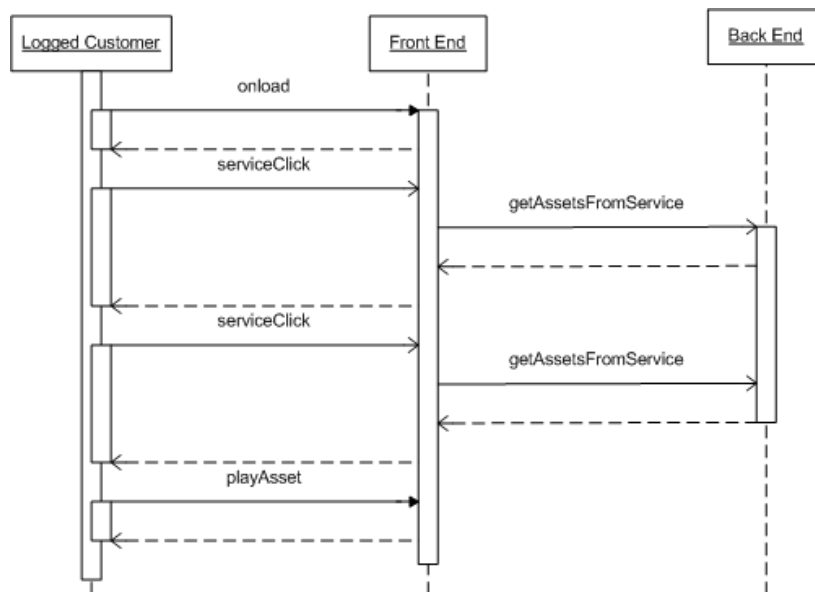


Figure 4.4: Sequence Diagram for the logged customer services navigation

the information to the application user. This section presents the user interface design, presenting both the gadget definition and the compositions used to provide the two aforementioned views (i.e. anonymous user and logged user).

4.2.1 Definition of Gadgets

This section defines the following information for each gadget developed for the DAM customer application:

- Gadget name
- Brief gadget description
- List of user interactions which trigger information updates and where (gadget and fields).

- Gadget image

Service Sources Gadget

This gadget lists the available service providers and its service catalogs in a directory display.

- Clicking on a service provider name updates the service list gadget (service names) and asset gadget (asset names).
- Clicking on a service catalog name updates the service list gadget (service names) and asset gadget (asset names).



Figure 4.5: Design of the Service Sources Gadget

Service List Gadget

This gadget lists the available services offered by all or a selected service / service catalog.

- Clicking on a service name updates the asset gadget (asset names) and the show evaluation information.

Search Gadget

This gadget allows the user to perform an asset search based on several film attributes, like the film genre or actors involved.

- Performing a search updates the asset gadget (asset names).

Media Work Detail Gadget

This gadget lists the film information associated to the selected asset (films are the only media work used for the prototype).

- Clicking on the visualization button the film can be visualized in a visor gadget we don't cover here as an external gadget already built in the mash up platform.

Show Evaluation Gadget

This gadget lists the evaluation information of the service or asset selected (i.e: rating and comments).

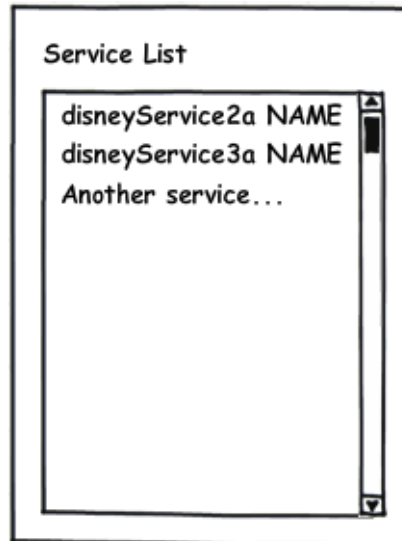


Figure 4.6: Design of the Service List Gadget

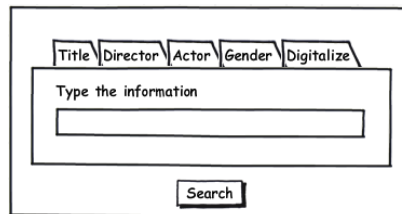


Figure 4.7: Design of the Search Gadget

- No interactions.

Create Evaluation Gadget

This gadget retrieves the evaluation information of the service or asset selected (i.e: rating and comments).

- Clicking on the save evaluation button stores the retrieved information in the Triple Space.

Assets Gadget

This gadget lists the assets offered in by the service provider / service catalog or service selected (or all if none of these are selected), as well as the asset details if one concrete asset is selected.

- If an asset name is selected, the details are filled (in the asset gadget). Additionally, the digitalized film details are displayed in the Mediawork Details Gadget are filled.

4.3 Composition of Views

The DAM customer application is designed to have two basic scenarios: one for the use of an anonymous user, and another for a logged user. While an anonymous user will

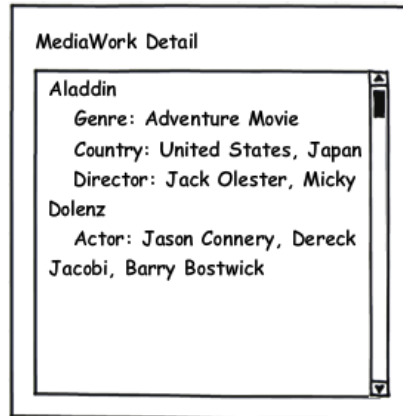


Figure 4.8: Design of the Media Work Details Gadget

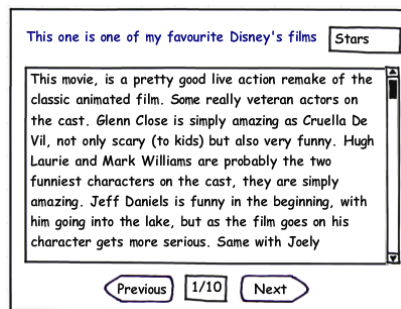


Figure 4.9: Design of the Evaluation Show Gadget

navigate through the public information (e.g: services, assets, film information and evaluations), the logged user will see its subscribed services, will be able to make evaluations and visualize some contents. Figure 4.12 shows how an anonymous composition could be made, showing service sources, service and asset lists, film details, evaluations and the search gadget. The flexibility of Web 2.0 interfaces enable a personal re-configuration of this user interface easily.

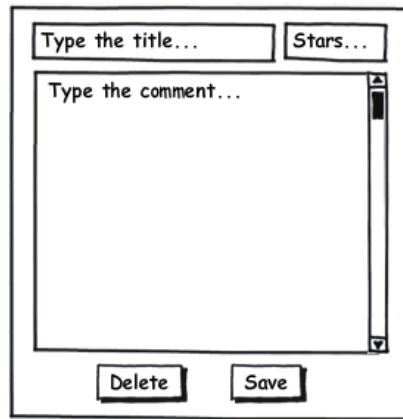


Figure 4.10: Design of the Creation Evaluation Gadget

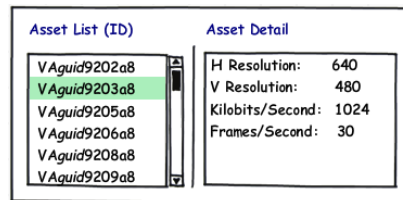


Figure 4.11: Design of the Assets Gadget

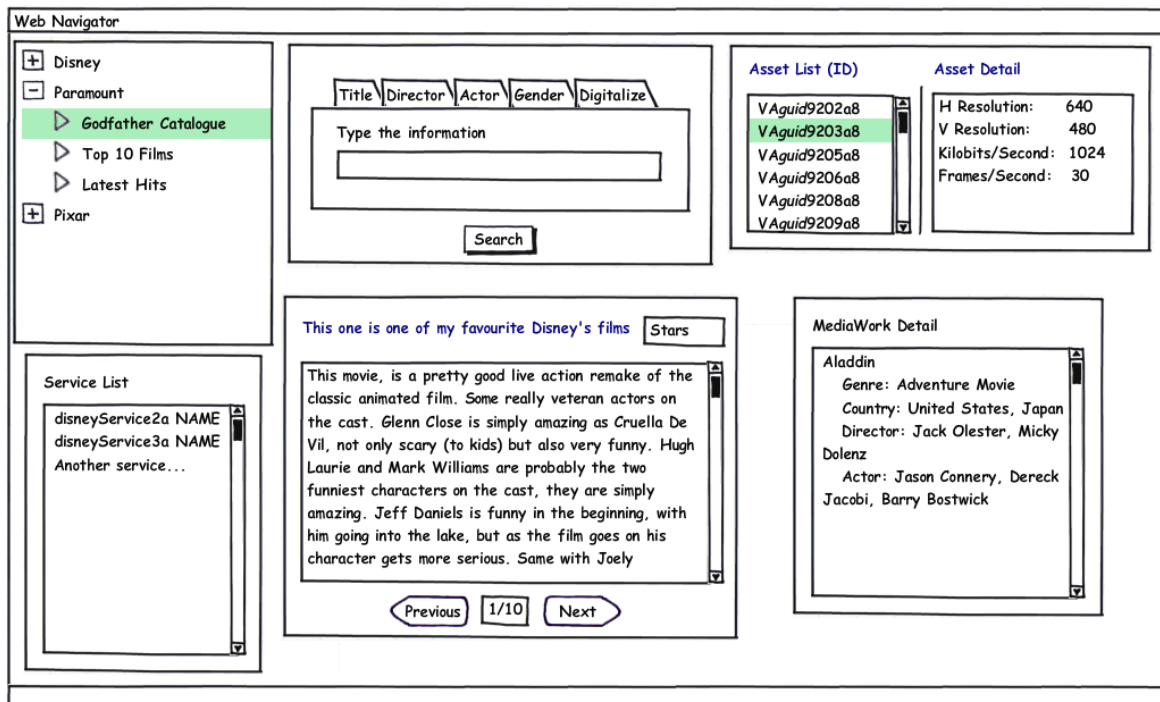


Figure 4.12: Example of an Anonymous Mashup Composition

5 CONCLUSIONS

The EAI use case defined as the very beginning of the TripCom project consisted on two main scenarios. The first one, the marketplace interaction between content and service providers was covered during the second year of the project. The second one, the use and evaluation of content services is covered in this deliverable, which outlines the design and implementation details of this customer oriented scenario.

Both scenarios brought about relevant technical requirements, and some of them (i.e. those which were found more relevant for a commercial development of the prototype) have been tested in the evaluation plan described. Tests have been described, their results presented, and special emphasis has been made in deriving technical and business conclusions that will be undoubtable valuable for further exploitation plans of the work carried out in the context of this work package.

REFERENCES

- [1] David de Francisco, Noelia Pérez, Doug Foxvog, Andreas Harth, Daniel Martin, Daniel Wutke, Martin Murth, and Elena Paslaru Bontas Simperl. Towards a digital content services design based on triple space. In Witold Abramowicz, editor, *Business Information Systems*, volume 4439/2007 of *Lecture Notes in Computer Science*, pages 163–179. Springer-Verlag, April 2007.
- [2] David de Francisco Marcos, Lyndon Nixon, and Germán Toro del Valle. Towards a multimedia content marketplace implementation based on triplesaces. In A. Sheth et al., editor, *Proceedings of the 7th International Semantic Web Conference (ISWC'08)*, number 5318 in *Lecture Notes in Computer Science*, pages 875–888. Springer Verlag Berlin Heidelberg, October 2008.
- [3] David de Francisco Marcos, Guillermo López Reyes, and Germán Toro del Valle. Eai prototype application. Deliverable, TripCom, October 2008.
- [4] M. Grüninger and M.S. Fox. Methodology for the design and evaluation of ontologies. In Skuce D, editor, *IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing*, pages 6.1–6.10, 1995.
- [5] Reto Krummenacher, Elena Simperl, Daniel Martin, Michael La te, Christian Schreiber, Alessio Carenini, and David de Francisco. Towards a scalable triple space v2.0. Deliverable, TripCom, March 2007.
- [6] W3C. Sparql query language for rdf. <http://www.w3.org/TR/rdf-sparql-query/>, January 2008.