TripCom

Triple Space Communication

FP6 – 027324

Deliverable

D8A.2

EAI Prototype Application

David de Francisco Marcos
Guillermo López Reyes
Germán Toro del Valle

October 23, 2008

Executive Summary

This deliverable presents the work performed to implement the Digital Asset Marketplace prototype presented in the D8A.1 deliverable. The prototype has focused on accomplishing the technical requirements stressed out by this deliverable, and will be completed with the scalability validation outlined by this deliverable and scheduled for the end of the project.

The Digital Content Marketplace implementation details are addressed in an accepted ISWC article which is attached, instead of re-writing the information in the deliverable. Additionally, the implementation of a second version of the use case ontology is delivered and documented as part of this deliverable. This ontology has not been used for the implementation of the prototype, but instead will be used in the scalability prototype which will be presented in the end of the project. This scalability prototype is outlined in this deliverable as well. Finally, some design documentation about the prototype implementation is attached as annexes of this deliverable.

The source code of the digital content marketplace implementation can be reached through the TripCom Sourceforge project in `http://sourceforge.net/projects/tripcom/`.

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.2	**Title**	EAI Prototype Application
Work Package	**Number**	8A	**Title**	Use Case 1 - Enterprise Application Integration

Date of Delivery	**Contractual**	M30	**Actual**	31-Sept-08
Status	version 0.10		final ☐	
Nature	prototype ☒ report ☐ dissemination ☐			
Dissemination Level	public ☒ consortium ☐			

Authors (Partner)	David de Francisco Marcos (TID), Guillermo López Reyes (TID), 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)	This deliverable covers the implementation of a Digital Content Marketplace based on Triple Spaces as the underlying communication and coordination infrastructure. The technical design of the prototype, including its integration into the TripCom infrastructure, is presented in this deliverable. In order to validate this prototype in terms of scalability, the outline of a validation plan is presented.
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**
2008-01-07	1	David de Francisco	TOC defined.
2008-16-07	2	David de Francisco, Germán Toro	Introduction and ISWC Paper integration made.
2008-24-07	3	Guillermo López	Ontology section filled.
2008-18-08	4	David de Francisco	Summary and abstract defined.
2008-20-09	5	David de Francisco	Evaluation section filled. Minor cleaning-up.
2008-30-09	6	David de Francisco	Integration of comments from the peer review (Cefriel) made.
2008-12-10	7	David de Francisco	Integration of comments from QA review.
2008-23-10	8	David de Francisco	Integration of Vassil's comments with his help.

Project Consortium Information

Acronym	Partner	Contact
Semantic Technology Institute Innsbruck http://www.sti-innsbruck.at	STI	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	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, Finland 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

CP	Content Provider
DAM	Digital Asset Management
EAI	Enterprise Application Integration
EDI	Electronic Data Interchange
ISWC	International Semantic Web Conference
OWL	Ontology Web Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SLA	Service Level Agreements
SOA	Service Oriented Architecture
SP	Service Provider
TS	Triple Space
UML	Unified Modeling Language
UN/EDIFACT	United Nations / Electronic Data Interchange for Administration, Commerce and Transport
WS	Web Service
WSML	Web Service Modeling Language

1 Introduction

Triple Space Computing [7] is a novel paradigm which aims at defining a scalable, co-ordinated and flexible communication by the combination of space based computing principles and semantic knowledge representation [14]. Such communication offers powerful integration means to enterprise application integration (EAI) scenarios [13]. This kind of integration scenarios arise in collaborative businesses, where heterogenous systems collaborate to offer complex services. That's the case of digital asset management (DAM), a very fragmented value chained business with strong integration needs.

Telefónica, as a telecommunication company, foresees great business opportunities in the digital content distribution. With this aim in mind, the work carried out in this TripCom work package has motivated the suitability of the Triple Space Computing paradigm, and more precisely the TripCom technology within the DAM domain [3]. The aim of this EAI Use Case development has been devoted to the implementation of a digital content marketplace, where all the actors involved in a DAM business scenario could be integrated in a flexible manner by using the TripCom infrastructure. Deliverable 8A.1 [4] presented a functional analysis of a DAM prototype, defining its functional requirements. The suitability of Triple Spaces in order to implement marketplace-like communication patterns has been discussed and motivated in this work [2].

In this deliverable we present the implementation of such marketplace infrastructure carried out so far. Section 2 describes the implementation of the digital content marketplace. The information from the section is contained in the paper [5] attached as part of the deliverable, and the source code can be reached through the TripCom Sourceforge project [1]. This marketplace will be evaluated with the framework presented in Section 3, which will be developed during the project's third year. This framework will make use of the refined ontology described in Section 4. After the conclusions in Section 5, this deliverable attaches two annexes: Annex A contains the design diagrams of the marketplace implementation. Annex B explains the mappings between the EDIFACT ontologies delivered as part of the WP7 work and the second version of the DAM ontology.

[1] See `http://sourceforge.net/projects/tripcom/`

2 Implementation of a Digital Content Marketplace Using Triplespaces

The focus of this deliverable is on the implementation of a digital content marketplace. The objective of the implementation is the creation of a prototype which can demonstrate the functional capabilities of the TripCom infrastructure. This implementation will be refined based on the results of our prototype evaluation, which will be undertaken as described in Section 3. This section has been complemented with an attached ISWC 2008 paper to be published [5]. This article covers the motivation of the implementation, the prototype architecture design, the implementation details and outlines the planned evaluation of this architecture, which will be further extended in Section 3.

The deliverable also contains the current version of the digital content marketplace implementation (both the code and the documentation). This code is publicly accessible as part of the TripCom Sourceforge project (See `http://sourceforge.net/projects/tripcom/`).

3 Evaluation of the Digital Content Marketplace

The objective of the prototype implementation provided by this deliverable is to demonstrate the suitability of a Triplespace-based implementation for a digital content marketplace. In this sense, the emphasis has been made in the integration of the TripCom infrastructure as well as in using its functionality to fulfil the technical requirements listed in [4]. This section outlines the evaluation process for the EAI use case, which will be conducted by the WP8A work during the last 6 months of the project. This process will aims at deriving some conclusions (both from the technical and application point of view), which can provide valuable exploitation conclusions, as well as to refine the use case prototype implementation. Validation tests will include three kind of tests described below: functional, scalability and concurrent access tests.

3.1 Objectives of the Validation

The Triple Space Computing paradigm sets the scalability as one of the crucial features offered by the Triple Spaces. As stated in [12], there is an obvious trade-off between scalability and functionality which has led to the definition of three different Triplespace APIs: Core, Extended and Further Extended. This trade-off requires an analysis that can validate the applicability of the TripCom infrastructure to solve real business situations, as well as their related technical challenges. A real digital content marketplace arises different kind of requirements:

Functional Requirements Related to the expected functionality to be covered by the marketplace implementation. A functional analysis was given in [4], and although it is a simplified view of a commercial implementation, it has enough technical rigor to be considered as a basic functionality for the proposed test bed. These requirements have been contrasted with the functionality offered by the TripCom infrastructure in [1] (refer to Section 2.1), and prioritized according to their functional criticalness.

Non-functional Requirements Non functional requirements have been also prioritized in [1] (refer to Section 2.2).

QoS Requirements A real implementation of the system prototyped in this deliverable would expect some quality of service requirements in terms of scalability and performance to be provided by the system. In this sense, we have defined a set of tentative scalability levels and some associated success factors (see Section 5 in [1]).

The next sections described the evaluation principles which will be followed in order to check the fulfillment of previous requirements.

3.2 Functional Evaluation

Functional evaluation is an ongoing process in which both unitary and integration tests are made. Functional evaluation checks the fulfillment of the functional analysis presented in [4]. Deliverable D8A.3 will provide a list of the functional tests performed, providing documentation about how to reproduce the tests and expected results. Such

functional tests are being tested with the Web application whose code is submitted as part of this deliverable.

3.3 Scalability Evaluation

The scalability tests will test the load and the performance of retrieving information by measuring the performance of the marketplace implementation. The marketplace implementation will use the rd without URL operations as the data size increases, simulating the fact that content providers publish their catalogues in the Web using Triple Spaces. With this aim in mind, a real data set will be simulated, and the RDF data will be stored in spaces. From the physical point of view, these spaces will be stored in distributed kernels. This decision aims at simulating the most typical business deployment, in which each content provider would have its own catalogue kernel, in order to have its data physically controlled. This is the way content catalogues are deployed in current Imagenio [1] implementation.

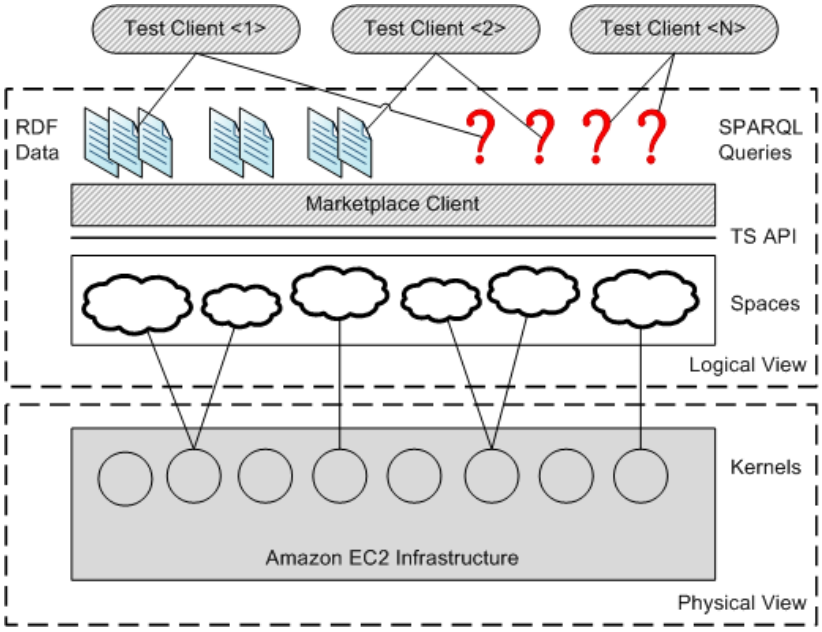

Figure 3.1: Architecture for the Scalability Test Infrastructure

As depicted in Figure 3.1, a number of scalability test clients will be run (top part of the figure). These clients will perform the data insertion and retrieval, using RDF data files and SPARQL queries respectively. Queries and data insertion orders are managed by the Marketplace Client, which stores and retrieves data from the spaces using the TripCom infrastructure. These spaces will be physically stored in a number of kernels deployed in the Amazon Elastic Cloud Computing infrastructure.

In a real marketplace implementation, content catalogues and the data from customers are likely to be hosted by each service provider and content provider individually, keeping the control of their own kernel and sharing the information in a controlled fashion by using the TripCom infrastructure. This behavior will be simulated in the use case deployment made in the Amazon infrastructure.

[1] Imagenio is the Telefónica TV over IP product. See `www.telefonica.es/tol/imagenio.html` (in Spanish).

3.4 Concurrent Access Validation

The most significant business logic implemented by the EAI use case is the business interaction in an auction pattern. This has been motivated as a suitable demonstration of the communication and co-ordination capabilities offered by the TripCom infrastructure. Such capabilities will be tested by a concurrent access validation process. Given that an auction is created by a service provider and that we have several auction participants competing in a given time-frame, it's a clear requirement for the auction participants that the marketplace implementation is capable of assuring a performance in terms of bid publishing and retrieving. The objective of this test is to measure the performance of these operations with an increasing number of concurrent participants in the auction environment (i.e. information stored in a single kernel, frequent information publication and retrieval).

Concurrent access validation will follow the approach of simulating the behavior of auction participants by using lightweight clients. Auctions will be conducted by running these clients in parallel, according to a simple business logic that will decide when they perform new bids and the conditions of these bids. Concurrent access validation could be considered as part of the functional validation, but the need of specific testing clients has made us to consider this evaluation independently.

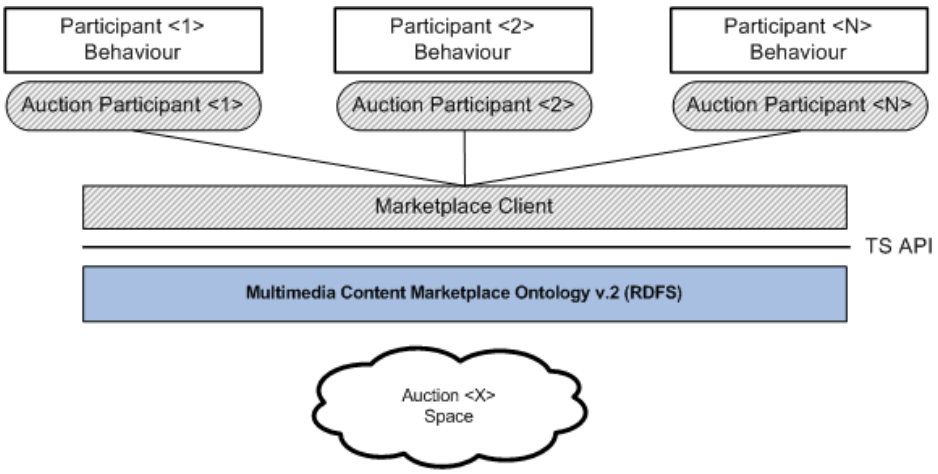

Figure 3.2: Architecture for the Concurrent Access Validation Infrastructure

The Figure 3.2 shows the architecture of the concurrent access validation infrastructure. In this architecture, an increasing number of auction participants publish and retrieve information by making use of the Marketplace Client, which exposes the TS API functionality. This information makes use of the second version of the use case ontology (named Multimedia Content Marketplace Ontology v.2 in the figure), documented in the Section 4. The information is stored in a single non-distributed space.

4 Digital Asset Management Ontology Refinement

This section presents the second version of the DAM ontology, encoded in RDFS [15]. This new version covers more terms related to the marketplace functionality and is aligned with the WP7 set of ontologies, thus integrating knowledge from the widely accepted UN EDIFACT standard [8]. As stated in [4], a triple space based model aims, among other things, at providing the means for the integration of the inherent heterogeneous formats and data used by the different actors who interact in a Digital Asset Management (DAM) marketplace. In order to decouple the actors in a schema-wise manner, the use of ontologies is considered, allowing the transformation between different representations of the same meaning. RDFS allows an easier integration of new business requirements, as well as the combination of several schemas in our marketplace architecture. Besides, it ensures a formal and reusable model, making easier to define further logic or interpretation, when mediation or inference is needed.

The use of semantics not only ensures the schema decoupling and the mediation between heterogeneous data schemas used by actors, but also grants the inference of information not explicitly stated, as well as the automatic classification of multimedia content. Furthermore, the system [5] should actively moderate business transactions by understanding the messages exchanged by actors, granting support for a smart information handling. Although Triple Space coordination capabilities provide themselves a decoupled implementation, the use of semantics is crucial to effectively address the heterogeneity of actors and the support of agile business transactions.

4.1 Ontology Structure

The goal of the EAI use case [5] is to define a prototype digital content marketplace implementation. The scenario is conceived as a realistic solution for a real business problem, while the functionality of the Triple Space infrastructure is tested. For the semantic description of the use case, an ontology has been developed and refined. The ontology design process has followed the Methontology methodology [10]. The tasks carried out have been building a glossary of terms, building taxonomies of concepts, building binary relation diagrams and defining attributes, and afterwards describing formal axioms and rules. The final step has been the definition of the instances.

In order to be able to deal with the wide range of the ontology, the ontology has been developed by dividing the scope of the ontology in sub-ontologies which are related among them. As a result, the scope of the ontology is divided into three separated sub-ontologies, which cover the different domains of the use case, providing a common vocabulary to all actors (see figure 4.1):

- Multimedia Content Ontology: This ontology defines a taxonomy of assets, allowing the automatic classification of individual assets and the retrieval of inferred (not explicitly stated) information.

- Business Transactions Ontology: This ontology imports concepts from an EDIFACT ontology [9], with the aim of solving the heterogeneity problems that arise when performing a business transaction, by agreeing the format of the messages sent.

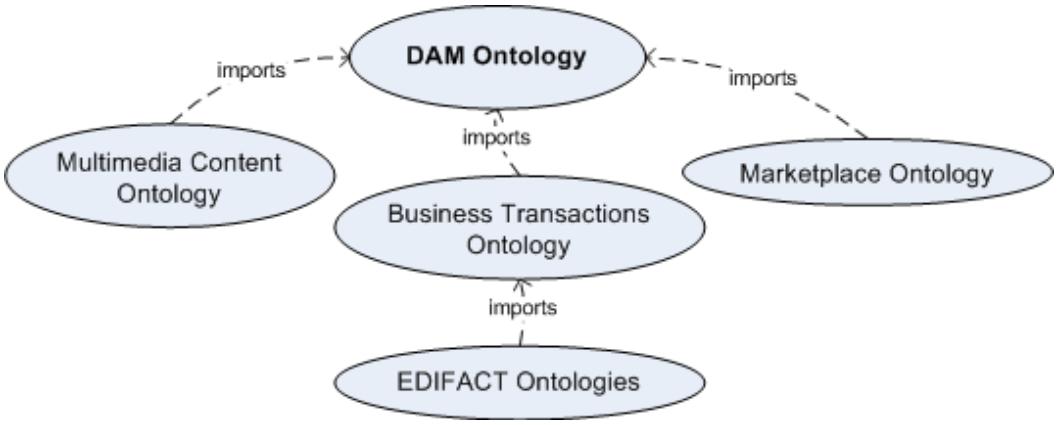

Figure 4.1: Ontology Structure of the Marketplace Implementation

- Marketplace Ontology: This ontology defines the knowledge related to the marketplace logic, such as actors and auctions related knowledge. Its objective is to facilitate the interoperability between actors and the semi-automatic handling of auctions (i.e: automatically starting an auction when a service is terminated by a content provider).

Since the ontology covers several aspects which are not DAM domain specific, many of the necessary concepts for this ontology have already been defined in other ontologies. There has been an important concern in reusing as many existing concepts as possible, as long as they match the needs of the design, reducing the developing time, and making possible the interaction with other ontologies. Given the business orientation of the use case, the ontology design has reused the work carried out by Work Package 7. The reutilization of EDIFACT related concepts has enabled easier ontology evolution and maintenance, since most of the knowledge formalized by WP 7 ontologies is widely used by industry. On the other hand, since the EAI use case final goal is to distribute digital multimedia content, there is a need to describe the content that will be delivered. It is necessary to indicate who the writer of a book is, which songs are sung by a group or singer, and which genre a film belongs to, for example. Building a whole new ontology from scratch to match these knowledge capture requirements would not be efficient, so extensive mappings to OpenCyc [1] terms have been made for non use case specific terms. Some explained examples of mappings to external ontologies are shown in WP7 D7.3 [6]. For a complete list of all mappings made by DAM ontology to external ontologies, see Appendix B.

4.2 Ontology Evaluation

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 [11]. 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

[1] OpenCyc is one of the largest general knowledge ontologies which is also used in the WP7 ontologies

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 [16] and executed against a test data set. Details about the ontology evaluation tests will be provided in D8A.3.

4.2.1 Content Related Competency Questions

1. Return me the offered films/books for a given subject, both simple and composed (i.e: drama, or action, thriller and crime).
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'.

4.2.2 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.

4.2.3 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.

4.2.4 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.

4.3 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 R-entailment rules, extending RDFS ontology with custom semantics, thus helping us to motivate the benefits of semantic enriched data. These R-entailment rules are shared in this deliverable, as future steps for the use case exploitation. The reasoning capabilities needed to use these rules are provided by the storage component of the TripCom infrastructure (in the TRREE reasoner), so no extra component is needed to achieve this functionality.

4.3.1 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 a .
 y daml:hasDirector a .
------------------------
x daml:isSimilarToFilm y .
```

This rule defines that two films 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. Hence, lowering time and cost to develop new features is a strong advantage to be further evaluated and used in our dissemination activities.

4.3.2 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. The logic could be implemented as part of the RDFS semantics or extended as R-entailment rules. In this section we discuss the different approaches of introducing business logic in the model. To express which are the content types playable in an iPhone (e.g., VCD, DVD, PDF, MPG), two technical alternatives exist. The first is to define the specific content types compatible with iPhone as subtypes of `iPhoneSuitableAsset`. The other option is to introduce a new R-entailment rule, which uses as constants the individual formats.

In the previous scenario that requires a hierarchical product categorization we recommend to implement the business logic as integration part of the ontology. The main advantage is the better flexibility and possibility to reuse the logic with other reasoners. Some times, however, there is a need to extend RDFS semantics to capture domain knowledge. For example, if we want to say that a film that features an actor who has been awarded in his practice is of type `FeaturesAwardedActor`, we can use a R-entailment rule like the following one (prefixes are omitted):

```
movie rdf:type Movie
actor rdf:type Actor
movie dam:hasActor actor
actor dam:awardedBy institution
```

```
------------
movie rdf:type FeaturesAwardedActor
```

As remarked before, both options (to define business logic using RDFS semantics or R-entailment rules), are supported by the TRREE reasoner integrated in the storage component of the Tripcom infrastructure.

5 Conclusions

This deliverable has presented the EAI use case prototype implementation carried out by the work package 8A. This implementation has been rooted on the technical analysis performed during the first year of the project, and has produced a Web application which addresses most of the requirements stressed out during that period. Details about the implementation plan, the architecture followed and the relevance of the prototype are addressed in the article attached as part of this deliverable. The deliverable presents the ontology refinement process which has produced a second use case ontology release, which integrates some knowledge from the EDIFACT ontologies being developed within the project.

The remaining work of the work package is the validation of the prototype. The validation approach has been discussed, and the functionality, scalability and concurrent validation processes described. It will part of the remaining effort of the work package to conduct this validation plan and to document and analyze the results. This validation process will derive in valuable exploitation conclusions which will be presented as well.

References

[1] Dario Cerizza, doug foxvog, David de Francisco, Daniel Martin, and Hans Moritsch. Tripcom evaluation plan. Internal document, TripCom, May 2008.

[2] David de Francisco, Javier Elicegui, Daniel Martin, Martin Murth, and Daniel Wutke. Using triple spaces to implement a marketplace pattern. In *Proceedings of the first workshop on Space Based Computing as Semantic Middleware for Enterprise Application Integration : SBC 2007, in conjunction with ESTC 2007. Vienna, Austria, May 31, 2007*, pages 1–8. Online, May 2007.

[3] 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.

[4] David de Francisco Marcos, Daniel Martin, Thorsten Scheibler, Daniel Wutke, Andreas Harth, Martin Murth, and Elena Paslaru Bontas Simperl. Tripcom requirements analysis and architecture profile for eai applications. Deliverable, TripCom, May 2007.

[5] David de Francisco Marcos, Lyndon Nixon, and Germán Toro del Valle. Towards a multimedia content marketplace implementation based on triplespaces. 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.

[6] David de Francisco Marcos, Guillermo López Reyes, and doug foxvog. Relationships among ontology modules within edifact and with external ontologies. Deliverable, TripCom, October 2008.

[7] Dieter Fensel. Triple-space computing: Semantic web services based on persistent publication of information. In Finn Arve Aagesen, Chutiporn Anutariya, and Vilas Wuwongse, editors, *Proc. of the IFIP Int'l Conf. on Intelligence in Communication Systems*, volume 3283 of *Lecture Notes in Computer Science*, pages 43–53. Springer-Verlag, November 2004.

[8] United Nations Economic Commission for Europe. United nations directories for electronic data interchange for administration, commerce and transport. http://www.unece.org. 2005.

[9] D. Foxvog and C. Bussler. Ontologizing edi semantics. In Springer Verlag, editor, *Advances in Conceptual Modeling - Theory and Practice.*, volume 4231/2006 of *Lecture Notes in Computer Science.*, pages 301–311, November 2006.

[10] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering*. Advanced Information and Knowledge Processing. Springer Verlag, November 2003.

[11] 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.

[12] Reto Krummenacher and Elena Simperl et al. Towards a scalable triple space. Deliverable, TripCom, April 2008.

[13] David S. Linthicum. *Enterprise application integration*. Addison-Wesley Longman Ltd., Essex, UK, UK, 2000.

[14] Johannes Riemer, Francisco Martín-Recuerda, et al. Triple space computing: Adding semantics to space-based computing. In *The Semantic Web - ASWC 2006, First Asian Semantic Web Conference, Beijing, China, September 3-7, 2006, Proceedings*, volume 4185 of *Lecture Notes in Computer Science*, pages 300–306. Springer Verlag, September 2006.

[15] W3C. Rdf vocabulary description language 1.0: Rdf schema. http://www.w3.org/TR/rdf-schema/, February 2004.

[16] W3C. Sparql query language for rdf. http://www.w3.org/TR/rdf-sparql-query/, January 2008.

A Appendix: Design Documentation

In this technical annex the interaction of the most common scenarios we will handle in the use case application of WP8A are documented. Interactions are depicted using UML Sequence diagrams. Half arrows represent asynchronous interactions; meanwhile normal arrows represent synchronous interactions. We will use the following abbreviations, already listed in the abbreviation list at the beginning of the deliverable:

CP Content Provider.

SP Service Provider.

Space hierarchy diagrams will contain a number of marketplace actors, denoted as rectangles. Ellipses denote spaces that will be used by these actors.

A.1 Content Catalogue Management

In this scenario (see Figure A.1), several CP want to publish their content offer in a common Content Catalogue Space, allowing any actor involved in the use case to query any information regarding their content. In order to ensure that competitors can't modify your content offers, each CP creates a subspace in the Content Catalogue Space (root space), granting read access to everyone but restricting data insertion and/or deletion to creator only (see Figure A.2). SP will be able to use the template matching mechanism in order to query any content before any offer related to this content is published in the catalogue, blocking its read until this kind of content offer is published.

A.1.1 UML Sequence Diagram:

Figure A.1 shows the UML sequence diagram of the catalogue management.

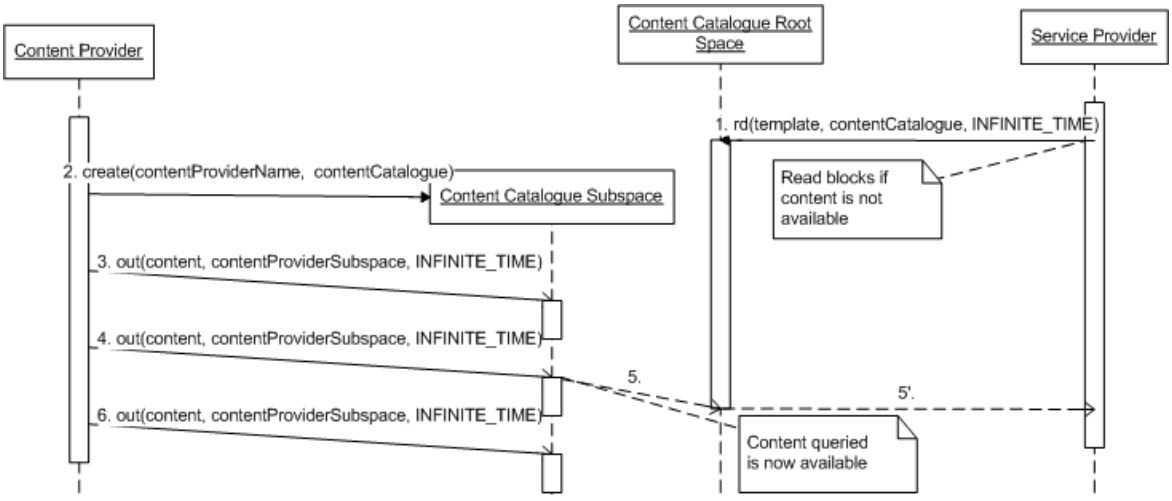

Figure A.1: Content Catalogue Management UML Sequence Diagram

A.1.2 Actors Involved:

Content Provider Provides or updates content offers which will be added to the catalogue.

Service Provider In this context a Service provider is an actor who is searching for some concrete contents to compose a content service.

Content Catalogue Root space where all content subspaces are placed.

Content Catalogue Subspace A subspace for the content offers made by a concrete CP.

A.1.3 Interaction Steps:

1. A Service provider looks for a content which might not be published in the space already. In this case, read operation blocks until an available content match the Template provided.
2. A Subspace is created within the Content Catalogue root space, which was created by the application. This Subspace allows only the creator CP to write and modify triples inside and everyone to read from it.
3. CP publishes a new content offer in the subspace it has just created.
4. CP modifies the content offer it created in step 3.
5. Modified content matches the template provided with the read operation performed by the SP in step 1 and is returned to SP thorough a notification.
6. Content offer is marked as not available by CP, thus modifying the status value in the content definition (updating the information). We will follow the policy of no tuple deletion, but tuple invalidation, in order to i) store the history of the catalogue and ii) allow a dynamic content management in which contents can be easily re-offered.

A.1.4 Space Hierarchy:

Figure A.2 shows the space hierarchy diagram of the catalogue management.

A.2 Auction Management

This scenario (see Figure A.3) describes the typical behaviour of an auction lifecycle in which a Service Provider is looking for content in order to provide a content service, and starts an auction to get a provider for this content. Auction Participants will join the auction and perform binding bids which might be validated by the auction creator. In order to implement this business logic, the space hierarchy depicted in Figure A.4 is used.

A.2.1 UML Sequence Diagram:

Figure A.3 shows the UML sequence diagram of the auction management.

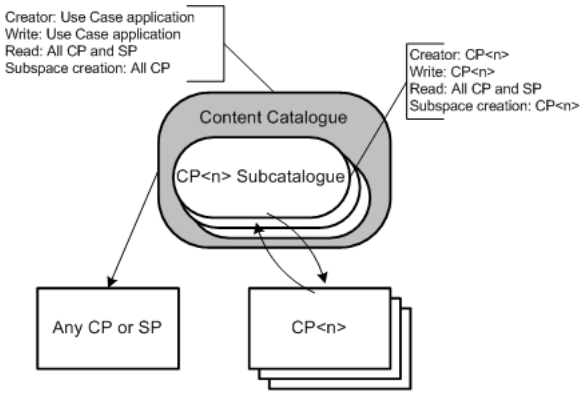

Figure A.2: Content Catalogue Management Space Hierarchy

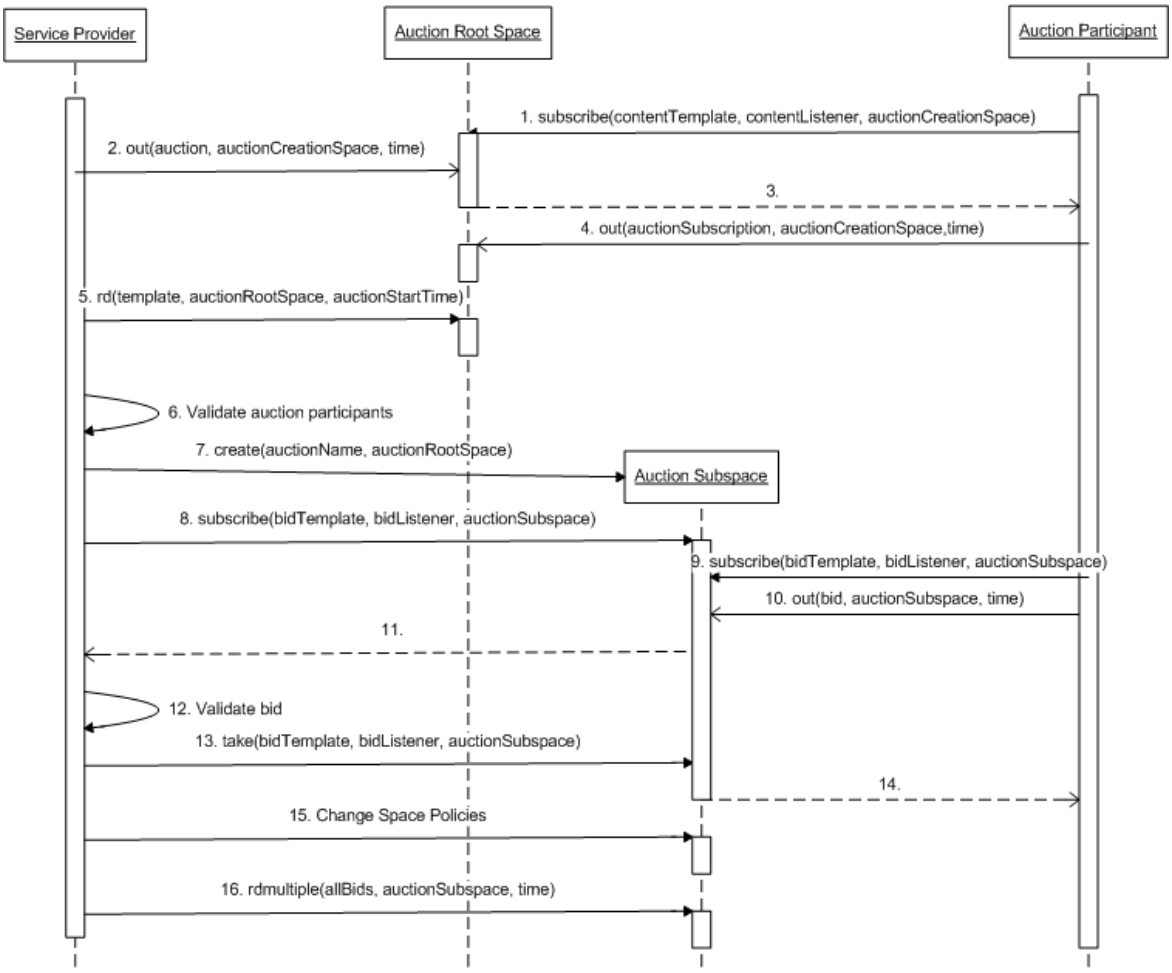

Figure A.3: Auction Management UML Sequence Diagram

A.2.2 Actors Involved:

Service Provider A SP who is searching for target content and wants to find a CP who can deliver it by using an auction.

Auction Creation Space A root space where auctions are published, as well as subspaces for each action created.

Auction Space Each auction is held in a separate space. These spaces are created by the SP creating the auction. Here all bids will be written by participants and creating SP has the right to delete those bids which are no valid.

Auction Participant Any of the CP invited to the auction.

A.2.3 Interaction Steps:

1. A CP are subscribed to the Auction Creation Space in order to get notified if any content targeted in any auction can be provided by them.
2. A SP publishes a new auction arrangement in the Auction Creation Space, in order to get a content it needs to provide a service to its clients.
3. CPs subscribed to the space whose templates are matched are notified of the new auction.
4. CPs interested in the auction can subscribe to it by publishing a request in the Auction Creation Space.
5. SP gets the auction subscription requests.
6. SP validates all subscriptions received from auction participants (i.e: checking internal black lists).
7. SP creates an Auction Space allowing all validated CPs to write bids (no bid can be modified or deleted), as well as everyone to consult existing bids in the auction.
8. Subscription to auction management messages to get notified to things like winning bid change or auction end.
9. The same step for auction participants.
10. An auction participant writes a new bid.
11. SP gets notified about this bid.
12. Bid is validated by the SP following its own validation logic.
13. If the bid is rejected, the graph containing the information of this bid is deleted from the Auction Space by the SP.
14. In that case, CP which submitted the bid is notified.
15. Once the auction timeout reaches (which is controlled by the Service Provider), policies of the Auction Space are changed avoiding new bids to be published.
16. SP reads all the bids published in the space to be evaluated by the SP.

A.2.4 Space Hierarchy:

Figure A.4 shows the space hierarchy diagram of the auction management.

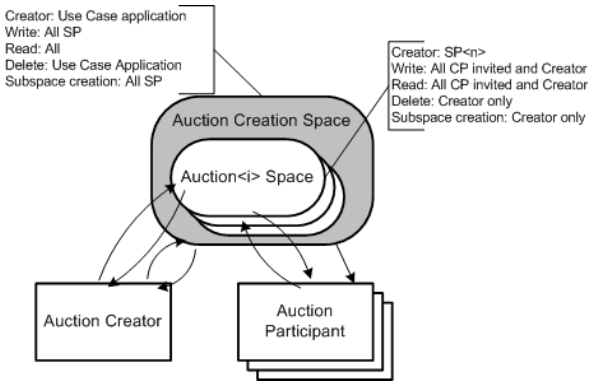

Figure A.4: Auction Management Space Hierarchy

A.3 Contracts Management

This scenario (see Figure A.5) describes a contract management flow, which takes in place when an auction is finished and a contract is formalized according to one bid chosen as the winning one by the SP. SP will present the contract to CP which signs it, and both of them are subscribed to any change of the document. During a contract management phase, no information can be removed from the space, avoiding this any unilateral modification of the contract. If an agent modifies a contract, other agents involved can easily be notified about the change and compare to previous information, thus minimizing the effect of any malicious behavior by contract signers. In order to implement this business logic, the space hierarchy depicted in Figure A.6 is used.

A.3.1 UML Sequence Diagram:

Figure A.5 shows the UML sequence diagram of the contract management.

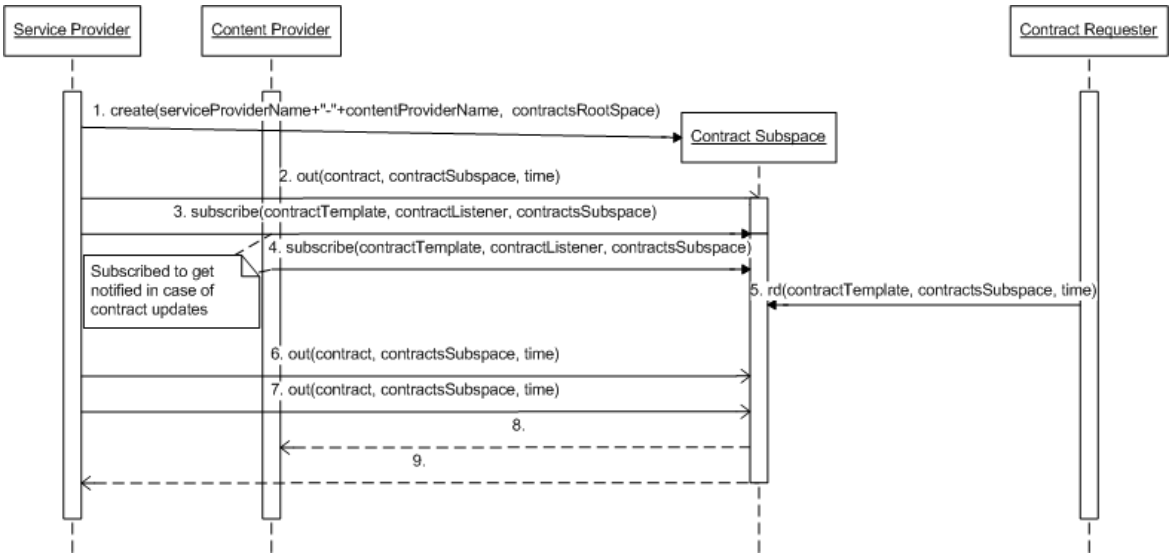

Figure A.5: Contracts Management UML Sequence Diagram

A.3.2 Actors Involved:

Service Provider The SP who has created an auction and has decided the best bid after the auction has finished.

Content Provider The auction winner according to SP's decision.

Contract Subspace A subspace where all contracts between this SP and CP are stored. A new subspace will be needed per each pair of actors having a business relationship.

Contract Requester Since bids are binding, contracts will remain public to be consulted by everyone or at least all service and content providers.

A.3.3 Interaction Steps:

1. Space crated to store all contracts between this Service Provider and Content Provider. We will have one space per pair of business partners we might find.
2. The content of the contract is written by the service provider.
3. SP is subscribed to the space to get notified if any change in the contract is published in the space.
4. Same subscription of step 3 for the CP.
5. Any external agent might request any contract information since bids were public and therefore contract information is already known.
6. Contract modification by SP (i.e: bad quality of the contents reported).
7. Contract modification by CP (i.e: content unavailability, see Section 1.5).
8. Notification to any change to CP.
9. Notification to any change to SP.

A.3.4 Space Hierarchy:

Figure A.6 shows the space hierarchy diagram of the contract management.

A.4 Service Management

This scenario (See Figure A.7) describes the service management flow in which a SP adds new content services to its service offering, allowing interested users to subscribe to services and rate them, providing feedback to SP. In order to implement this business logic, the space hierarchy depicted in Figure A.8 is used.

A.4.1 UML Sequence Diagram:

Figure A.7 shows the UML sequence diagram of the service management.

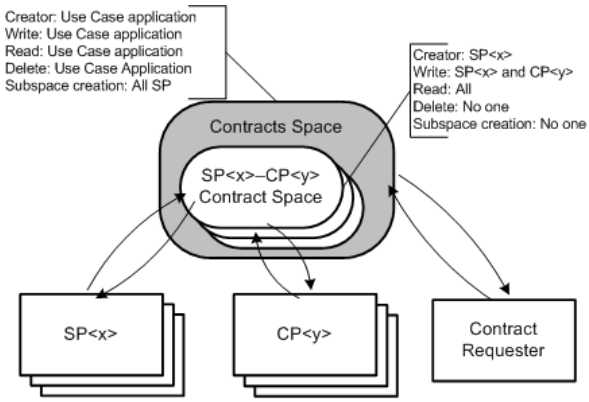

Figure A.6: Contracts Management Space Hierarchy

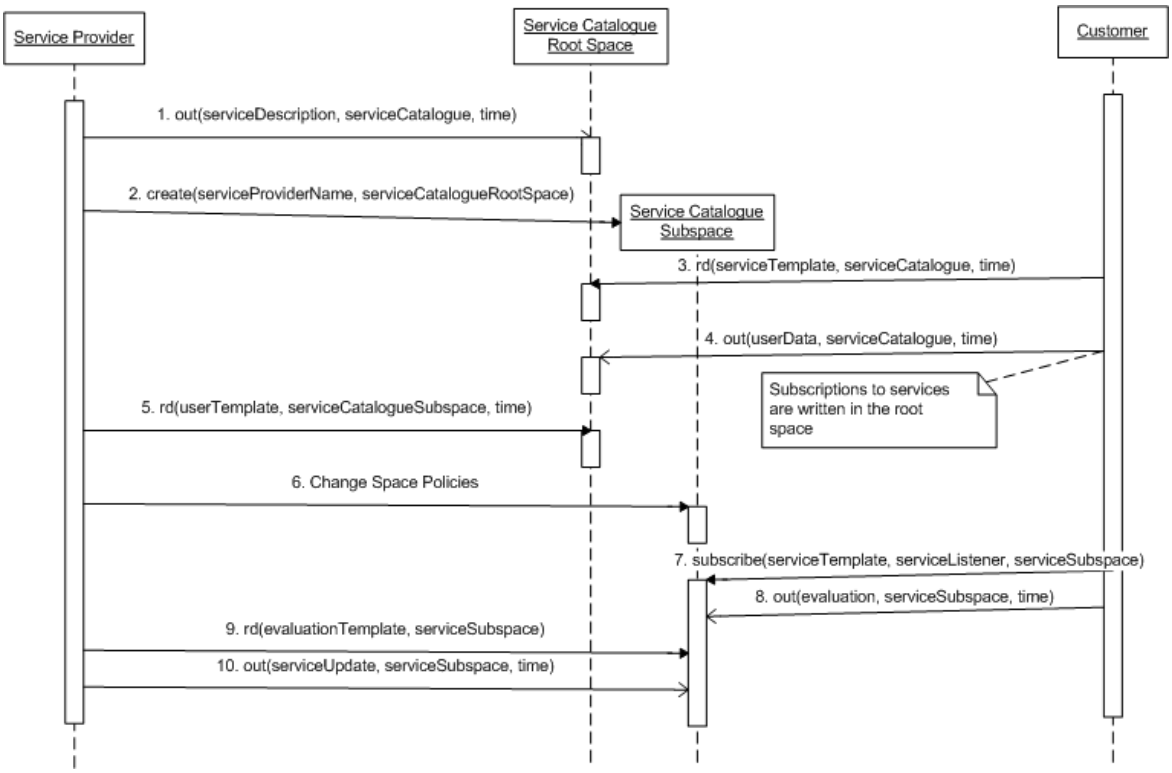

Figure A.7: Services Management UML Sequence Diagram

A.4.2 Actors Involved:

Service Provider The SP in charge of bundling the service once contracts are successfully stored.

Service Catalogue Root space where all SP subspaces storing available services are placed. It also stores user subscription and unsubscription to services.

Service Space Each SP creates a subspace to store user feedback regarding its services. SP will allow access to the space to subscribed users and deny it when users are unsubscribed.

Customer Final customers.

A.4.3 Interaction Steps:

1. A SP publishes a new Service description in the Service Catalogue, offering it to final customers.
2. The SP creates a new space for storing the user evaluation of the service, as well as enabling any communication needed between the SP and customers of the service.
3. Any user read from the Service Catalogue services that can fulfil user requirements. Note that this read might block if no service related to customer requirements is already available.
4. User registers to the service by publishing the information required by the SP in the Service Catalogue.
5. SP updates the Service Space policies, allowing the new subscribed customer to write evaluations of the service.
6. The customer subscribes to the Service Space to get notified of any updated information regarding the service he has subscribed to.
7. The customer writes an evaluation of the service to the Service Space.
8. SP reads any evaluations it is interested in from the Service Space.
9. SP updates the Service information by publishing the information in the Service Space (i.e: service unavailability).
10. Customers subscribed to the Service are notified of the change of the service.

A.4.4 Space Hierarchy:

Figure A.8 shows the UML sequence diagram of the catalogue management.

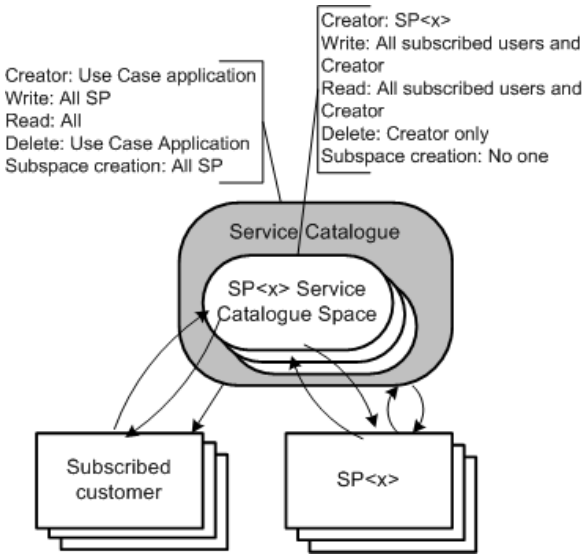

Figure A.8: Services Management Space Hierarchy

A.5 Content Unavailability

This scenario (See Figure A.9) reflects an example of a contract break due to a content unavailability. As the diagram depicts, this derives in side effects, since the service is unavailable and another CP has to be found.

A.5.1 UML Sequence Diagram:

Figure A.9 shows the UML sequence diagram of the content unavailability scenario.

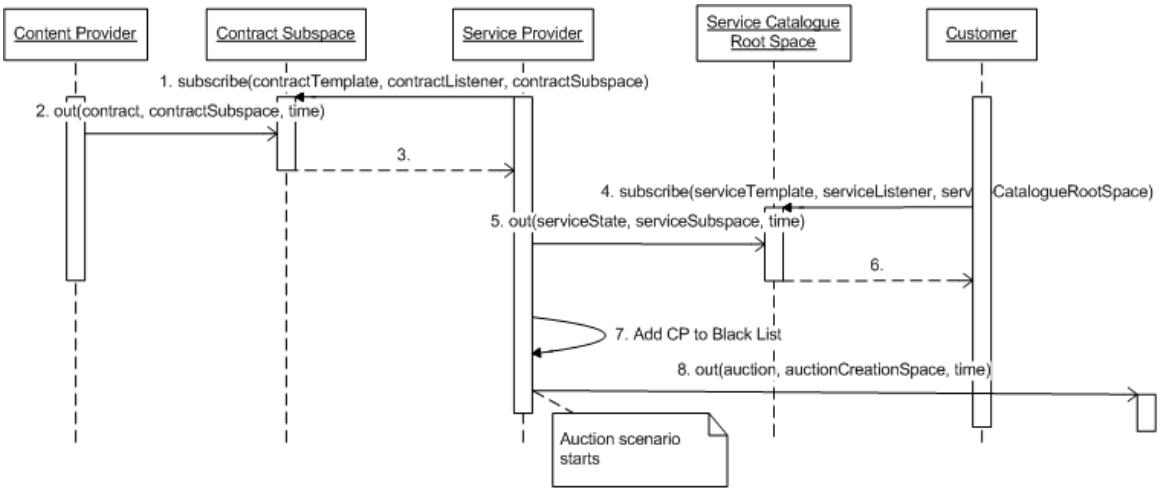

Figure A.9: Content Unavailability UML Sequence Diagram

A.5.2 Actors Involved:

Content Provider A CP participating in one or many contracts, who has some content unavailable at this moment.

Service Catalogue Root space where all SP subspaces storing available services are placed. It also stores user subscription and un-subscription to services.

Service Provider The SP offering a DAM service.

Service Space Each SP creates a subspace to store user feedback regarding its services. SP will allow access to the space to subscribed users and deny it when users are unsubscribed.

Customer Final customer of a DAM Service.

Auction Creation Space A root space where auctions are published, as well as subspaces for each action created.

A.5.3 Interaction Steps:

1. SP subscribes in order to get notified about any change in the contract status.
2. CP modifies a contract notifying that some content is no longer available.

3. SP receives the notification from the TS, since was previously subscribed to any change in the contract.
4. The customer subscribes to get notified about any change in the contract status.
5. SP modifies the Service description, making it unavailable.
6. Customers are notified.
7. SP adds CP to its black list, providing feedback to the auction process (this logic depends on each SP).
8. SP writes a new auction arrangement to the Auction Creation Space in order to substitute the CP who has stop providing this content.

B Appendix: DAM ontology list of mappings to external ontologies

The table below shows the complete list of mappings made to external ontologies for the DAM ontology. The first column shows the name of the imported term. The second one indicates the type (concept, relation or instance) and the ontology it is mapped to, and the third column indicates what kind of relationship exists between the imported term and the DAM ontology, following the legend below:

1. subConceptOf DAM Concept
2. superConceptOf DAM Concept
3. relatedTo DAM Concept
4. Indirect through other imported concepts
5. Relates DAM concepts to OpenCyc Concepts
6. instanceOf df/DateFormat
7. filler of meta-class dam/PaymentMethod
8. instanceOf DAM/VideoGenre
9. instanceOf DAM/AudioGenre
10. instanceOf DAM/DocumentGenre

Imported term	Type	Link
Actor	OpenCyc Concept	1, 3
Address-LocationDesignator	OpenCyc Concept	4
Award	OpenCyc Concept	2, 3
Band-MusicGroup	OpenCyc Concept	1
BankAccount	OpenCyc Concept	1
Book	OpenCyc Concept	2, 3
Business	OpenCyc Concept	2
Catalog	OpenCyc Concept	1
Choir	OpenCyc Concept	4
City	OpenCyc Concept	1, 3
Composer	OpenCyc Concept	3
Country	OpenCyc Concept	3
CreditCard	OpenCyc Concept	1, 3
Currency	OpenCyc Concept	1
Date	OpenCyc Concept	1, 2

Imported term	Type	Link
Editor	OpenCyc Concept	1, 2
FinancialInstitution	OpenCyc Concept	1
MagazineSeries	OpenCyc Concept	4
MediaProfessional	OpenCyc Concept	1, 3
MediaSeriesProduct	OpenCyc Concept	1
Movie	OpenCyc Concept	1
MusicalInstrument	OpenCyc Concept	1
MusicalInstrumentPlayer	OpenCyc Concept	1, 3
MusicalPerformer	OpenCyc Concept	3
Musician	OpenCyc Concept	1
MusicPerformanceAgent	OpenCyc Concept	4
MusicPerformanceOrganization	OpenCyc Concept	2
NaturalLanguage	OpenCyc Concept	1, 3
Orchestra	OpenCyc Concept	2, 3
Organization	OpenCyc Concept	1
Person	OpenCyc Concept	1, 3
Producer	OpenCyc Concept	4
Product	OpenCyc Concept	1, 3
Quartet	OpenCyc Concept	3
Singer	OpenCyc Concept	3
SocialBeing	OpenCyc Concept	1, 3
Song	OpenCyc Concept	1
State	OpenCyc Concept	1
Survey	OpenCyc Concept	1
TVNewsShow	OpenCyc Concept	1, 3
TVShow	OpenCyc Concept	1
TVShowHost	OpenCyc Concept	3
VocalEnsemble	OpenCyc Concept	3
Writer	OpenCyc Concept	1
accountNumber	OpenCyc Relation	5
addressOfLocation	OpenCyc Relation	4
bankAdministrator	OpenCyc Relation	4
cityOfAddress	OpenCyc Relation	4
clients	OpenCyc Relation	4
countryOfAddress	OpenCyc Relation	4
expirationDate	OpenCyc Relation	4

Imported term	Type	Link
hasMembers	OpenCyc Relation	5
BankTransfer	BankOperation Concept	7
PaymentByCreditCard	BankOperation Concept	4, 7
PaymentByGiro	BankOperation Concept	7
PaymentByBankDraft	BankOperation Concept	7
DateFormat	DateFormat Concept	4
CCYYMMB_Format	DateFormat Instance	6
CCYYMMDD2_Format	DateFormat Instance	6
CCYYMMDDHHMM2_Format	DateFormat Instance	6
CCYYMMDDHHMMSS2_Format	DateFormat Instance	6
CCYYMMDDHHMMSSZZZ_Format	DateFormat Instance	6
CCYYMMDDHHMMSSZZZZZ_Format	DateFormat Instance	6
CCYYMMDDHHMMSS_Format	DateFormat Instance	6
CCYYMMDDHHMMZZZZZ_Format	DateFormat Instance	6
CCYYMMDDHHMMZZZ_Format	DateFormat Instance	6
CCYYMMDDHHMM_CCYYMMDD_Format	DateFormat Instance	6
CCYYMMDDHHMM_Format	DateFormat Instance	6
CCYYMMDD_CCYYMMDD_Format	DateFormat Instance	6
CCYYMMDD_Format	DateFormat Instance	6
CCYYMMM_MMM_Format	DateFormat Instance	6
CCYYMMW_Format	DateFormat Instance	6
CCYYMM_CCYYMM_Format	DateFormat Instance	6
CCYYMM_Format	DateFormat Instance	6
CCYYQ_Format	DateFormat Instance	6
CCYYWW_Format	DateFormat Instance	6
CCYY_CCYY_Format	DateFormat Instance	6
CCYY_Format	DateFormat Instance	6
CC_Format	DateFormat Instance	6
DDD_Format	DateFormat Instance	6
DDMMCCYYHHMMSS_Format	DateFormat Instance	6
DDMMCCYYHHMM_Format	DateFormat Instance	6
DDMMCCYY_Format	DateFormat Instance	6
DDMMYYHHMM_Format	DateFormat Instance	6
DDMMYY_Format	DateFormat Instance	6
DD_DD_Format	DateFormat Instance	6
DD_Format	DateFormat Instance	6

Imported term	Type	Link
FourMonthsDuration	DateFormat Instance	6
HHMMHHMM_Format	DateFormat Instance	6
HHMM_Format	DateFormat Instance	6
HHMMSS_Format	DateFormat Instance	6
HHMM_HHMM_Format	DateFormat Instance	6
MMCCYY_Format	DateFormat Instance	6
HalfMonthsDuration	DateFormat Instance	6
MMDDCCYY2_Format	DateFormat Instance	6
MMDDCCYY_Format	DateFormat Instance	6
MMDDYY_Format	DateFormat Instance	6
MMDD_Format	DateFormat Instance	6
MMDD_MMDD_Format	DateFormat Instance	6
MMMCCYY_Format	DateFormat Instance	6
MMYY_Format	DateFormat Instance	6
MM_Format	DateFormat Instance	6
SemestersDuration	DateFormat Instance	6
TenDayPeriodsDuration	DateFormat Instance	6
UnstructuredTime_Format	DateFormat Instance	6
YDDD_Format	DateFormat Instance	6
YYDDD_Format	DateFormat Instance	6
YYMMDD_Format	DateFormat Instance	6
YYMMDD_MMDD_Format	DateFormat Instance	6
YYMMDD_YYMMDD_Format	DateFormat Instance	6
YYMMMDD_Format	DateFormat Instance	6
YYMM_Format	DateFormat Instance	6
YYMM_YYMM_Format	DateFormat Instance	6
YY_Format	DateFormat Instance	6
YY_YY_Format	DateFormat Instance	6
DaysDuration	DateFormat Instance	6
HoursDuration	DateFormat Instance	6
MinutesDuration	DateFormat Instance	6
MonthsDuration	DateFormat Instance	6
QuartersDuration	DateFormat Instance	6
SecondsDuration	DateFormat Instance	6
WeeksDuration	DateFormat Instance	6
YearsDuration	DateFormat Instance	6

Imported term	Type	Link
ActionMovie	OpenCyc Instance	8
AdventureMovie	OpenCyc Instance	8
AIGoneWrongMovie	OpenCyc Instance	8
BasketballMovie	OpenCyc Instance	8
BiblicalEpicMovie	OpenCyc Instance	8
ChildrensMovie	OpenCyc Instance	8
ComedyMovie	OpenCyc Instance	8
CrimeMovie	OpenCyc Instance	8
DarkComedyMovie	OpenCyc Instance	8
DocumentaryFilm	OpenCyc Instance	8
DramaticMovie	OpenCyc Instance	8
EducationalMovie	OpenCyc Instance	8
EroticDramaMovie	OpenCyc Instance	8
ExperimentalMovie	OpenCyc Instance	8
FamilyMovie	OpenCyc Instance	8
FantasyMovie	OpenCyc Instance	8
FilmNoirMovie	OpenCyc Instance	8
GangsterMovie	OpenCyc Instance	8
HistoricalNarrativeMovie	OpenCyc Instance	8
HorrorMovie	OpenCyc Instance	8
InstructionalMovie	OpenCyc Instance	8
MusicalMovie	OpenCyc Instance	8
MysteryMovie	OpenCyc Instance	8
ParodyMovie	OpenCyc Instance	8
PornographicMovie	OpenCyc Instance	8
RomanceMovie	OpenCyc Instance	8
RomanticComedyMovie	OpenCyc Instance	8
SatireMovie	OpenCyc Instance	8
ScienceFictionMovie	OpenCyc Instance	8
SpaghettiWesternMovie	OpenCyc Instance	8
SportsMovie	OpenCyc Instance	8
SuspenseMovie	OpenCyc Instance	8
TeenMovie	OpenCyc Instance	8
ThrillerMovie WarMovie	OpenCyc Instance	8
WesternMovie	OpenCyc Instance	8
AnimatedMovie	OpenCyc Instance	8

Imported term	Type	Link
AnimeMovie	OpenCyc Instance	8
ClassicMovie	OpenCyc Instance	8
CultMovie	OpenCyc Instance	8
ForeignMovie	OpenCyc Instance	8
IndependentMovie	OpenCyc Instance	8
NewReleaseMovie	OpenCyc Instance	8
PropositionalConceptualWork-AdultGenre	OpenCyc Instance	10
PropositionalConceptualWork-AdventureGenre	OpenCyc Instance	10
PropositionalConceptualWork-BasketballGenre	OpenCyc Instance	10
PropositionalConceptualWork-BiblicalEpicGenre	OpenCyc Instance	10
PropositionalConceptualWork-BiographyGenre	OpenCyc Instance	10
PropositionalConceptualWork-ChildrensNarrativeGenre	OpenCyc Instance	10
PropositionalConceptualWork-ClassicGenre	OpenCyc Instance	10
PropositionalConceptualWork-ComedyGenre	OpenCyc Instance	10
PropositionalConceptualWork-CrimeGenre	OpenCyc Instance	10
PropositionalConceptualWork-CultGenre	OpenCyc Instance	10
PropositionalConceptualWork-DarkComedyGenre	OpenCyc Instance	10
PropositionalConceptualWork-DramaGenre	OpenCyc Instance	10
PropositionalConceptualWork-EducationalGenre	OpenCyc Instance	10
PropositionalConceptualWork-EroticDramaGenre	OpenCyc Instance	10
PropositionalConceptualWork-ExperimentalGenre	OpenCyc Instance	10
PropositionalConceptualWork-FamilyGenre	OpenCyc Instance	10
PropositionalConceptualWork-FantasyGenre	OpenCyc Instance	10
PropositionalConceptualWork-GangsterGenre	OpenCyc Instance	10
PropositionalConceptualWork-HistoricalFictionGenre	OpenCyc Instance	10
PropositionalConceptualWork-HistoricalNarrativeGenre	OpenCyc Instance	10
PropositionalConceptualWork-HorrorGenre	OpenCyc Instance	10
PropositionalConceptualWork-InstructionalGenre	OpenCyc Instance	10
PropositionalConceptualWork-LiteraryCriticismGenre	OpenCyc Instance	10
PropositionalConceptualWork-MemoirGenre	OpenCyc Instance	10
PropositionalConceptualWork-MusicalGenre	OpenCyc Instance	10
PropositionalConceptualWork-MysteryGenre	OpenCyc Instance	10
PropositionalConceptualWork-ParodyGenre	OpenCyc Instance	10
PropositionalConceptualWork-RomanceGenre	OpenCyc Instance	10
PropositionalConceptualWork-RomanticComedyGenre	OpenCyc Instance	10
PropositionalConceptualWork-SatireGenre	OpenCyc Instance	10

Imported term	Type	Link
PropositionalConceptualWork-ScienceFictionGenre	OpenCyc Instance	10
PropositionalConceptualWork-SportsGenre	OpenCyc Instance	10
PropositionalConceptualWork-SuspenseGenre	OpenCyc Instance	10
PropositionalConceptualWork-TeenGenre	OpenCyc Instance	10
PropositionalConceptualWork-ThrillerGenre	OpenCyc Instance	10
PropositionalConceptualWork-TragedyGenre	OpenCyc Instance	10
PropositionalConceptualWork-WarGenre	OpenCyc Instance	10
PropositionalConceptualWork-WesternGenre	OpenCyc Instance	10
MusicalComposition-1990sPop	OpenCyc Instance	9
MusicalComposition-AlternativeCountry	OpenCyc Instance	9
MusicalComposition-Ambient	OpenCyc Instance	9
MusicalComposition-Bluegrass	OpenCyc Instance	9
MusicalComposition-BluesRock	OpenCyc Instance	9
MusicalComposition-Celtic	OpenCyc Instance	9
MusicalComposition-Christian	OpenCyc Instance	9
MusicalComposition-Country	OpenCyc Instance	9
MusicalComposition-CountryWestern	OpenCyc Instance	9
MusicalComposition-Disco	OpenCyc Instance	9
MusicalComposition-Electronica	OpenCyc Instance	9
MusicalComposition-Folk	OpenCyc Instance	9
MusicalComposition-Funk	OpenCyc Instance	9
MusicalComposition-GangstaRap	OpenCyc Instance	9
MusicalComposition-Gospel	OpenCyc Instance	9
MusicalComposition-GrungeRock	OpenCyc Instance	9
MusicalComposition-HeavyMetal	OpenCyc Instance	9
MusicalComposition-House	OpenCyc Instance	9
MusicalComposition-Jazz	OpenCyc Instance	9
MusicalComposition-Klezmer	OpenCyc Instance	9
MusicalComposition-NewAge	OpenCyc Instance	9
MusicalComposition-Opera	OpenCyc Instance	9
MusicalComposition-PostPunk	OpenCyc Instance	9
MusicalComposition-ProgressiveRock	OpenCyc Instance	9
MusicalComposition-Punk	OpenCyc Instance	9
MusicalComposition-Reggae	OpenCyc Instance	9
MusicalComposition-RhythmAndBlues	OpenCyc Instance	9
MusicalComposition-RockAndRoll	OpenCyc Instance	9

Imported term	Type	Link
MusicalComposition-Romantic	OpenCyc Instance	9
MusicalComposition-Ska	OpenCyc Instance	9
MusicalComposition-Soul	OpenCyc Instance	9
MusicalComposition-SurfRock	OpenCyc Instance	9
MusicalComposition-Techno	OpenCyc Instance	9
MusicalComposition-Trance	OpenCyc Instance	9
MusicalComposition-1980sPop	OpenCyc Instance	9
MusicalComposition-ACappella	OpenCyc Instance	9
MusicalComposition-AlternativeRock	OpenCyc Instance	9
MusicalComposition-Baroque	OpenCyc Instance	9
MusicalComposition-Blues	OpenCyc Instance	9
MusicalComposition-BritishFolkRock	OpenCyc Instance	9
MusicalComposition-Choral	OpenCyc Instance	9
MusicalComposition-Classical	OpenCyc Instance	9
MusicalComposition-CountryRock	OpenCyc Instance	9
MusicalComposition-Cyberpunk	OpenCyc Instance	9
MusicalComposition-Electronic	OpenCyc Instance	9
MusicalComposition-Experimental	OpenCyc Instance	9
MusicalComposition-FolkRock	OpenCyc Instance	9
MusicalComposition-Fusion	OpenCyc Instance	9
MusicalComposition-GlamRock	OpenCyc Instance	9
MusicalComposition-Gothic	OpenCyc Instance	9
MusicalComposition-HardRock	OpenCyc Instance	9
MusicalComposition-HipHop	OpenCyc Instance	9
MusicalComposition-Industrial	OpenCyc Instance	9
MusicalComposition-JazzRock	OpenCyc Instance	9
MusicalComposition-Lounge	OpenCyc Instance	9
MusicalComposition-NewWave	OpenCyc Instance	9
MusicalComposition-PopRock	OpenCyc Instance	9
MusicalComposition-Progressive	OpenCyc Instance	9
MusicalComposition-Psychedelic	OpenCyc Instance	9
MusicalComposition-Rap	OpenCyc Instance	9
MusicalComposition-Religious	OpenCyc Instance	9
MusicalComposition-Rock	OpenCyc Instance	9
MusicalComposition-Rockabilly	OpenCyc Instance	9
MusicalComposition-Salsa	OpenCyc Instance	9

Imported term	Type	Link
MusicalComposition-Song	OpenCyc Instance	9
MusicalComposition-SouthernRock	OpenCyc Instance	9
MusicalComposition-Swing	OpenCyc Instance	9
MusicalComposition-TraditionalPop	OpenCyc Instance	9
MusicalComposition-WorldFusion	OpenCyc Instance	9