D 1.1.3 Knowledge Processing Requirements Analysis

Coordinator: Pavel Shvaiko (UniTn)
with contributions from
Alain Léger, François Paulus, Laurent Rocuet (FT);
Lyndon Nixon, Malgorzata Mochol (FU Berlin);
Yiannis Kompatsiaris, Vasileios Papastathis, Stamatia Dasiopoulou (CERTH)

Abstract.
EU-IST Network of Excellence (NoE) IST-2004-507482 KWEB
Deliverable D1.1.3 (WP1.1)

This document reports on (i) a typology of knowledge-based processing tasks and (ii) high level components needed to fulfill prototypical application requirements.

Document Identifier:	KWEB/2004/D1.1.3/v0.9
Class Deliverable:	KWEB EU-IST-2004-507482
Version:	V0.9
Date:	January 14, 2005
State:	Draft
Distribution:	Public

Knowledge Web Consortium

This document is part of a research project funded by the IST Program of the Commission of the European Communities as project number IST-2004-507482.

University of Innsbruck (UIBK) – Coordinator
Institute of Computer Science,
Technikerstrasse 13
A-6020 Innsbruck
Austria
Contact person: Dieter Fensel
E-mail address: dieter.fensel@uibk.ac.at

École Polythechnique Fédérale de Lausanne (EPFL)
Computer Science Department
Swiss Federal Institute of Technology
IN (Ecublens), CH-1015 Lausanne.
Switzerland
Contact person: Boi Faltings
E-mail address: boi.faltings@epfl.ch

France Telecom (FT)
4 Rue du Clos Courtel
35512 Cesson Sévigné
France. PO Box 91226
Contact person : Alain Leger
E-mail address: alain.leger@rd.francetelecom.com

Freie Universität Berlin (FU Berlin)
Takustrasse, 9
14195 Berlin
Germany
Contact person: Robert Tolksdorf
E-mail address: tolk@inf.fu-berlin.de

Free University of Bozen-Bolzano (FUB)
Piazza Domenicani 3
39100 Bolzano
Italy
Contact person: Enrico Franconi
E-mail address: franconi@inf.unibz.it

Institut National de Recherche en Informatique et en Automatique (INRIA)
ZIRST - 655 avenue de l'Europe - Montbonnot
Saint Martin
38334 Saint-Ismier
France
Contact person: Jérôme Euzenat
E-mail address: Jerome.Euzenat@inrialpes.fr

Centre for Research and Technology Hellas / Informatics and Telematics Institute (ITI-CERTH)
1st km Thermi – Panorama road
57001 Thermi-Thessaloniki
Greece. Po Box 361
Contact person: Michael G. Strintzis
E-mail address: strintzi@iti.gr

Learning Lab Lower Saxony (L3S)
Expo Plaza 1
30539 Hannover
Germany
Contact person: Wolfgang Nejdl
E-mail address: nejdl@learninglab.de

National University of Ireland Galway (NUIG)
National University of Ireland
Science and Technology Building
University Road
Galway
Ireland
Contact person: Christoph Bussler
E-mail address: chris.bussler@deri.ie

The Open University (OU)
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA
United Kingdom.
Contact person: Enrico Motta
E-mail address: e.motta@open.ac.uk

Universidad Politécnica de Madrid (UPM)
Campus de Montegancedo sn
28660 Boadilla del Monte
Spain
Contact person: Asunción Gómez Pérez
E-mail address: asun@fi.upm.es

University of Karlsruhe (UKARL)
Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB
Universität Karlsruhe
D-76128 Karlsruhe
Germany

Contact person: Rudi Studer
E-mail address: studer@aifb.uni-karlsruhe.de

University of Liverpool (UniLiv)
Chadwick Building, Peach Street
L697ZF Liverpool
United Kingdom
Contact person: Michael Wooldridge
E-mail address: M.J.Wooldridge@csc.liv.ac.uk

University of Manchester (UoM)
Room 2.32. Kilburn Building, Department of Computer Science, University of Manchester, Oxford Road
Manchester, M13 9PL
United Kingdom
Contact person: Carole Goble
E-mail address: carole@cs.man.ac.uk

University of Sheffield (USFD)
Regent Court, 211 Portobello street
S14DP Sheffield
United Kingdom
Contact person: Hamish Cunningham
E-mail address: hamish@dcs.shef.ac.uk

University of Trento (UniTn)
Via Sommarive 14
38050 Trento
Italy
Contact person: Fausto Giunchiglia
E-mail address: fausto@dit.unitn.it

Vrije Universiteit Amsterdam (VUA)
De Boelelaan 1081a
1081HV. Amsterdam
The Netherlands
Contact person: Frank van Harmelen
E-mail address: Frank.van.Harmelen@cs.vu.nl

Vrije Universiteit Brussel (VUB)
Pleinlaan 2, Building G10
1050 Brussels
Belgium
Contact person: Robert Meersman
E-mail address: robert.meersman@vub.ac.be

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed writing parts of this document:

Centre for Research and Technology Hellas
France Telecom
Freie Universität Berlin
Universidad Politécnica de Madrid
University of Trento
Vrije Universiteit Brussel

Changes

Version	Date	Author	Changes
0.1	10-09-2004	Pavel Shvaiko	Creation. Index of the document.
0.2	23-09-2004	Pavel Shvaiko	Knowledge processing tasks analysis and corresponding components descriptions from the *Recruitment* use case (FU Berlin) are added.
0.3	29-09-2004	Pavel Shvaiko	Structure of the deliverable is modified by suggestions from Alain Léger and Laurent Rocuet.
0.4	25-10-2004	Pavel Shvaiko	Knowledge processing tasks analysis and components descriptions from the *B2C market place for tourism* use case (FT) and *Multimedia content analysis and annotation* use case (CERTH) are added.
0.5	03-11-2004	Pavel Shvaiko	Restructuring the deliverable. Dressing.
0.6	19-11-2004	Pavel Shvaiko	Knowledge processing tasks analysis and components descriptions from the *E-Photo album automation services on a portal* use case (FT) is added. Updates and refinements of the *Recruitment*, *Multimedia content analysis and annotation*, and *B2C marketplace for tourism* use cases are performed.
0.7	6-12-2004	Pavel Shvaiko	Refining all the use cases. Adding executive summary and conclusions.
0.8	14-12-2004	Pavel Shvaiko	Finalizing remarks.
0.9	14-01-2005	Pavel Shvaiko	Modifications by suggestions from the reviewer' comments list.

Executive Summary

One of the main goals of the Knowledge Web Network of Excellence is in the transfer of ontology-based technologies from academia institutions to strategic industries. As an immediate step towards achieving this goal, the Work Package 1.1 of the Knowledge Web project contains the following tasks:

- Forming an industrial board of companies interested and active in the application of Semantic Web technologies;
- Specifying business needs scenarios;
- Identifying problems in industry that can be successfully treated with the Semantic Web technologies;
- Identifying the knowledge components and processing mechanisms that Semantic Web applications will need;
- Showing value of ontology-based applications in key business areas.

This deliverable reports on the results of the task of "Identifying the knowledge components and processing mechanisms that Semantic Web applications will need". This task analyses typical knowledge processing needs within the information systems of organizations. It aims (i) at creating a typology of knowledge processing tasks and (ii) corresponding high level components implementing those tasks.

The deliverable shows a methodology for identifying knowledge processing tasks and corresponding components by examples of some business use cases. In particular, we have analyzed in detail four use cases introduced in the deliverable D1.1.2 and discussed our vision of knowledge processing tasks with companies providing those use cases. It is also worth noticing that due to the fact of the same delivery date for the deliverable D1.1.3 and D1.1.2, not all of the intended use cases of D1.1.2 have been analyzed in D1.1.3.

For each use case under consideration we identify knowledge processing tasks it requires. We structure them as primary and secondary tasks according to their influence on the architecture of a system. Based on the primary and secondary knowledge processing tasks we first build a typology of knowledge processing tasks and corresponding high level components for *each* use case, and then for *all* the use cases together as a final (general) typology. Also, whenever possible, we indicate state of the art solutions and relevant activities being held in the Knowledge Web research workpackages, thus, showing applicability of the knowledge-based technology.

The process of building a typology of knowledge processing tasks has shown that most of the knowledge processing tasks identified repeat with some variations/specificity from use case to use case. This observation suggests that the constructed typology is stable, i.e., it contains the core knowledge processing tasks stipulated by the current industry needs.

Contents

1. Introduction 1

1.1. Standard specification methodology.. 1

2. Knowledge processing tasks analysis 3

2.1. Use Case 1. Recruitment by FU Berlin 3
2.1.1 Use case summary... 3
2.1.2 Service requirements... 3
2.1.3 Analysis 4
2.1.4 Design 10
2.1.4.1 The use case knowledge processing tasks 10
2.1.4.2 The use case typology of knowledge processing tasks 12
2.1.4.3 The use case library of high level components... 12
2.2 Use Case 2. Multimedia content analysis and annotation by CERTH 15
2.2.1 Use case summary... 15
2.2.2 Service requirements... 15
2.2.3 Analysis 16
2.2.4 Design 19
2.2.4.1 The use case knowledge processing tasks 19
2.2.4.2 The use case typology of knowledge processing tasks 21
2.2.4.3 The use case library of high level components... 22
2.3 Use Case 3. B2C marketplace for tourism by FT 24
2.3.1 Use case summary... 24
2.3.2 Service requirements... 24
2.3.3 Analysis 25
2.3.4 Design 31
2.3.4.1 The use case knowledge processing tasks 31
2.3.4.2 The use case typology of knowledge processing tasks 32
2.3.4.3 The use case library of high level components... 33
2.4 Use Case 4. E-Photo album automation services on a portal by FT. 36
2.4.1 Use case summary... 36
2.4.2 Service requirements... 36
2.4.3 Analysis 37
2.4.4 Design 42
2.4.4.1 The use case knowledge processing tasks 42
2.4.4.2 The use case typology of knowledge processing tasks 43
2.4.4.3 The use case library of high level components... 44

3. Typology of knowledge processing tasks and a library of high level components 45

4. Conclusions 48

6. References 48

Appendix 1. Dependencies with other deliverables 51

1. Introduction

This document focuses on the analysis of the knowledge processing tasks and corresponding high level components required by existing or foreseen systems. In particular, as input to the analysis where taken four use cases presented in the deliverable D.1.1.2:

Use Case 1. Recruitment by *FU Berlin*,
Use Case 2. Multimedia content analysis and annotation by *CERTH*,
Use Case 3. B2C marketplace for tourism by *FT*,
Use Case 4. E-Photo album automation services on a portal by *FT*.

1.1. Standard specification methodology

The standard specification methodology used for the identification of knowledge processing tasks and corresponding high level components is based on Rational Unified Process (RUP) [35,18]. It requires an intensive use of Unified Modeling Language (UML) [2].

The standard system specification methodology with respect to the Knowledge Web, WP1.1 activities is presented in Figure 1. Let us discuss it in some detail.

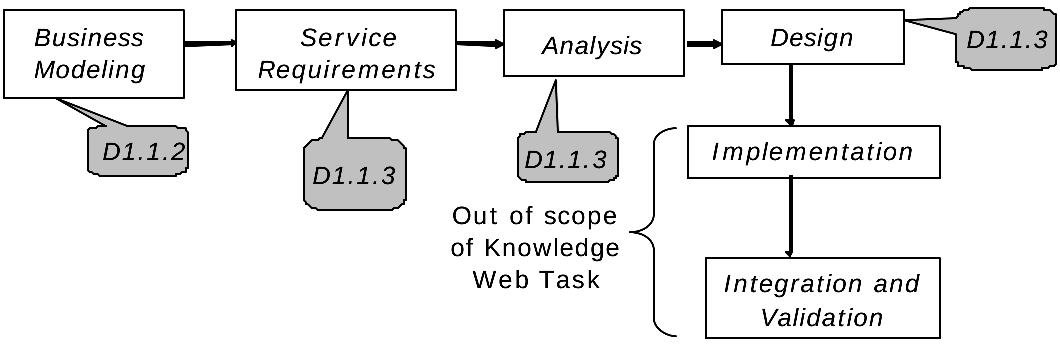

Fig.1. Development lifecycle in WP1.1

Business Modeling. This is the first step of the development lifecycle. Usually, business modeling is performed during the face to face meetings with industrial partners (belonging to industrial board). This activity and its results are described in deliverable D1.1.2.

Service Requirements. These are a set of services available through a system in order to implement a business case. They are determined through analysis of functional needs, which in turn imply some technical constraints (e.g., time response, scalability, number of connected customers) of a system to be developed. Service requirements are expressed in terms of UML (technical) use cases. Determining service requirements involves some additional interactions with industrial partners. This step is specified in deliverable D1.1.3.

Analysis. This step identifies classes, performs initial subsystem partitioning and looks at uses cases in detail. In particular, use cases are refined with the help of sequence diagrams, which incorporate the modules for the architecture proposal and the information flow between these modules to fulfill the use case functionality. This step is also specified in deliverable D1.1.3.

Design. This step anticipates implementation of a scenario. It refines and homogenizes classes, and identifies the architecture design. This step is *partially* specified in deliverable D1.1.3. In the Knowledge Web context, the aim of this step is only to identify knowledge processing tasks and components along with pointing out the technology locks they are stipulated by. Thus, the architecture proposal of the use cases under consideration is out of scope, and hence, we are not reporting class diagrams, etc.

Implementation, Integration and Validation. The aim of these three steps is to produce a robust implementation of a business use case respecting industrial needs. These steps are out of scope of the Knowledge Web project.

Notice, that during the *Design* step, we identify possible knowledge processing tasks. If the industry partner providing the use case has already decided on the technologies to be used, we not only report knowledge processing tasks and corresponding components implied by the technology locks, but also we provide a discussion on them and some references to the literature.

We structure knowledge processing tasks as primary and secondary tasks according to their influence on the architecture of the system. Primary tasks are the common parts for most of actions or parts of actions of the system. Secondary tasks are additional requirements, i.e., extensions of the common parts.

Based on the primary and secondary knowledge processing tasks we build a typology of knowledge-based processing tasks and corresponding high level components for each use case and for all of them together as a final (general) typology.

The results, i.e., (i) a general typology of knowledge-based processing tasks and (ii) high level components needed to fulfill prototypical application requirements will serve as a further guide for the research activities, connecting (emerging) industry problems with research issues.

The remainder of the document is organized as follows. Section 2 discusses in details the use cases under considerations. In particular, each use case is presented in four steps: (i) use case summary, (ii) service requirements, (iii) analysis, and (iv) design. Section 3 reports on a general typology of knowledge processing tasks and a library of high level components summarizing the findings of Section 2. Section 4 provides some conclusions.

2. Knowledge processing tasks analysis

2.1. Use Case 1. Recruitment by *FU Berlin*

2.1.1 Use case summary

This use case considers the needs of firms and jobseekers in the recruitment market, where the ultimate aim is the filling of job vacancies with the best suited candidates. An IT system for recruitment acts as a broker between both actors (a firm and a jobseeker) providing functionalities to:

- Publishing both job vacancies and applicant data;
- Searching for both suitable applicants and suitable vacancies.

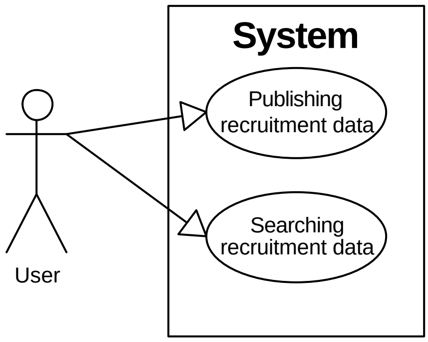

Fig.2.1.1 UML use case diagram for recruitment

Now, based on the functional requirements, it is possible to specify also different technical use cases taking part in the system. Those use cases will next allow for a detailed analysis of the technical needs.

2.1.2 Service requirements

Technical use cases diagram is presented in Figure 2.1.2. It allows identifying technical use cases and associated actors. Let us describe actors of Figure 2.1.2:

Firm: An organization which employs people.

Jobseeker: An individual who is seeking employment at an organization.

Administrator: The operator of the recruitment system.

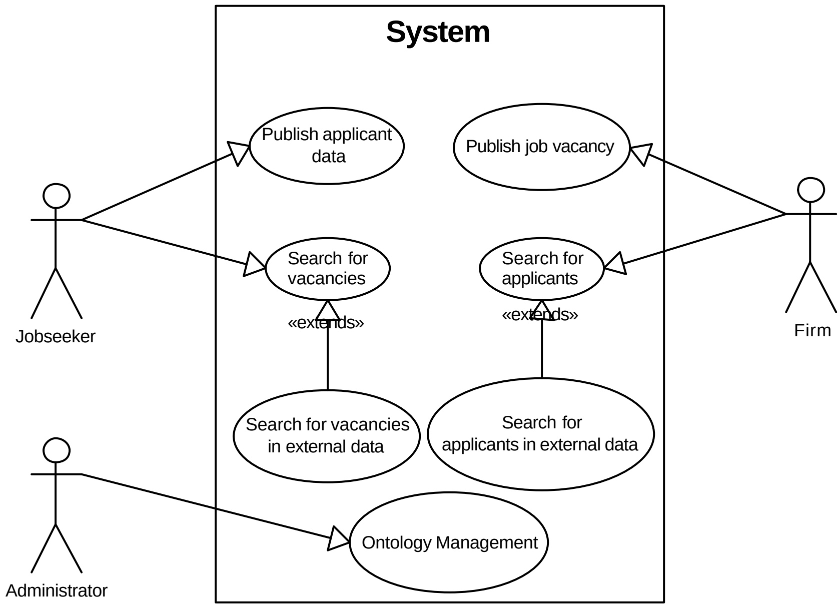

Fig.2.1.2 UML technical use case diagram for recruitment

2.1.3 Analysis

Now we analyze each technical use case of Figure 2.1.2 in detail. In particular, we consider *publishing the job vacancy or job applicant data*, *searching for either vacancies or applicants*, *searching for either vacancies or applicants in external data*, and *ontology management* technical use cases.

For each technical use case we first report the actors it involves, then we provide its summary, inputs and outputs, and finally we discuss with the help of sequence diagrams the flow of its events and possible technology locks.

Publishing the job vacancy / job application

Actors: Firm or jobseeker.

Summary: Publishing the description of a job vacancy or a job application to the recruitment system.

Preconditions and inputs:

- A firm has a vacancy or a jobseeker is seeking a job;
- Job or applicant description.

Post-conditions and outputs:

- The description of the vacancy or the applicant is stored within the system.

The flow of events for the *publishing the job vacancy or application* technical use case is presented in Figure 2.1.3.

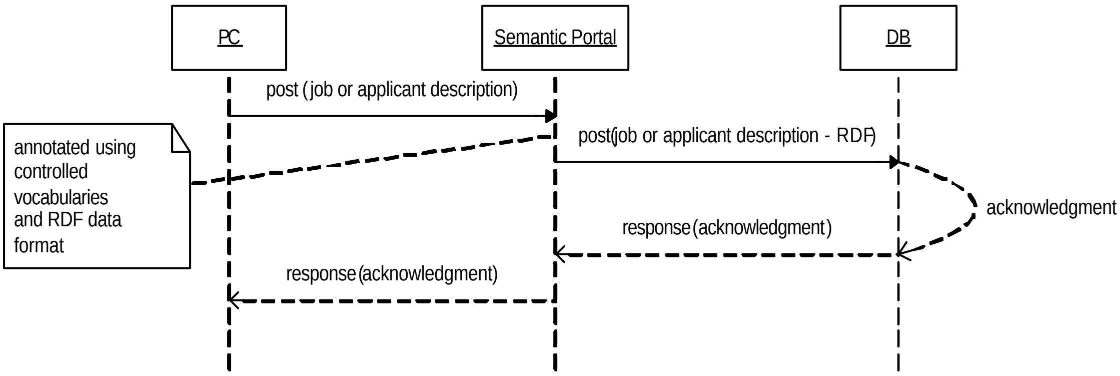

Fig.2.1.3 Flow of events: Publishing the job vacancy or application technical use case

Let us describe events of Figure 2.1.3 in more detail, first from the point of view of a firm publishing a vacancy and then a jobseeker publishing his or her application data. Italics indicate optional events in the use case.

Actor	Response
Firm indicates that it has a vacancy	Request a vacancy description *Make available means to produce this description*
Firm sends a vacancy description	Validate the vacancy description Store the vacancy description within the system Make the vacancy description publicly accessible

Actor	Response
Applicant indicates that (s)he is seeking a job	Request an applicant's personal description *Make available means to produce this description*
Applicant sends a personal description	Validate the personal description Place the personal description in a referenceable location Make the personal description publicly accessible

Technology locks identification: In the case of publishing data to a persistent store such as a back-end database, we find no technology locks. This is primarily due to the homogeneous nature of the data – by controlling the means of user input of the data. In particular, we make the annotation of that input data in terms of the recruitment RDF vocabulary. The use of a shared ontology reduces complexity in storage and data manipulation and helps to ensure a common view on the data for all users.

Search for vacancies or applicants

Actors: Firm or jobseeker.

Summary: The actor performs a search across the data in the recruitment system for a specific request. For example, a firm is looking for a suitable candidate for their job vacancy, or a jobseeker is looking for a suitable job vacancy.

Preconditions and inputs:

- A firm has a vacancy / an applicant is seeking a job.

Post-conditions and outputs:

- The firm has a list of suitable applicants for the vacancy / The applicant has a list of relevant vacancies suitable for his (her) requests.

The flow of events for the *search for vacancies/applicants* technical use case is presented in Figure 2.1.4.

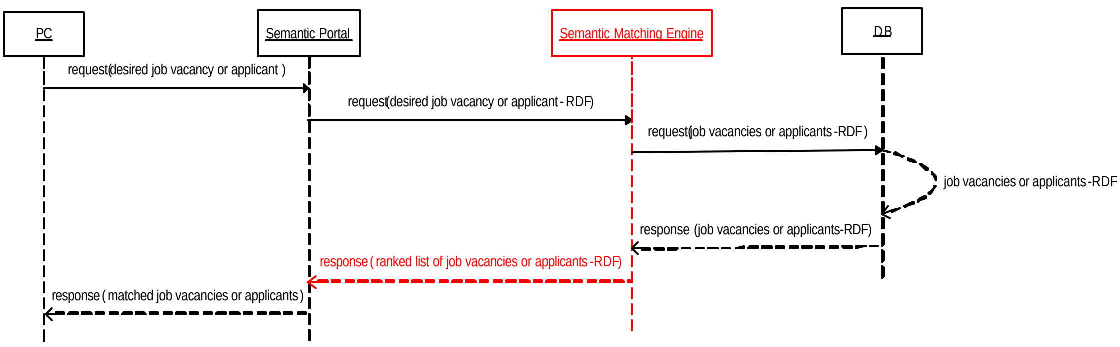

Fig.2.1 .4 Flow of events: Search for vacancies/applicants technical use case

Let us describe events of Figure 2.1.4 in more detail, first from the point of view of a firm searching for an ideal applicant description and then a jobseeker searching for an ideal vacancy description. Italics indicate optional events in the use case.

Actor	Response
Firm identifies a vacancy description or an ideal applicant description	The description is converted into a form suitable for the search mechanism A search mechanism is used to find applicants published on the system which "match" the input description Search results are organized according to

	a similarity ranking *Include information about on what basis the result was matched and ranked against the original query* Search results are passed back to the firm in a suitable format *The firm may wish to examine further on what basis the result was matched and ranked against the original query*

Actor	Response
Applicant identifies a personal description or an ideal vacancy description	The description is converted into a form suitable for the search mechanism A search mechanism is used to find vacancies published on the system which "match" the input description Search results are organized according to similarity ranking *Include information about on what basis the result was matched and ranked against the original query* Search results are passed back to the applicant in a suitable format *The jobseeker may wish to examine further on what basis the result was matched and ranked against the original query*

Technology locks identification: Technology locks are marked in red in Figure 2.1.4. We consider as a potential technology lock the *similarity ranking* carried out in the Semantic Matching Engine and the response of the ranked results list which includes *explanations* of the basis for the ranking. The use of a single common vocabulary for recruitment avoids issues of heterogeneity. Therefore, matching itself is not a lock here as long as (job vacancy/ job application) expressiveness needs are low, i.e., satisfied by the common vocabulary. However, it is still important to be able to determine weighs on conceptual matches and express factors of similarity between different, but related concepts. For example, when an applicant states that (s)he has a proficiency in C++, how would this rank differently against vacancies requiring persons with skills in Java, Microsoft .NET or 'object oriented programming'?

Search for vacancies or applicants in external data

This use case is the same (i.e., actors, inputs and outputs) as the one above except that external, heterogeneous content is additionally included in the search. Hence it is modeled as an extension of the previous use case. The flow of events for the *search for vacancies or applicants in external data* technical use case is presented in Figure 2.1.5.

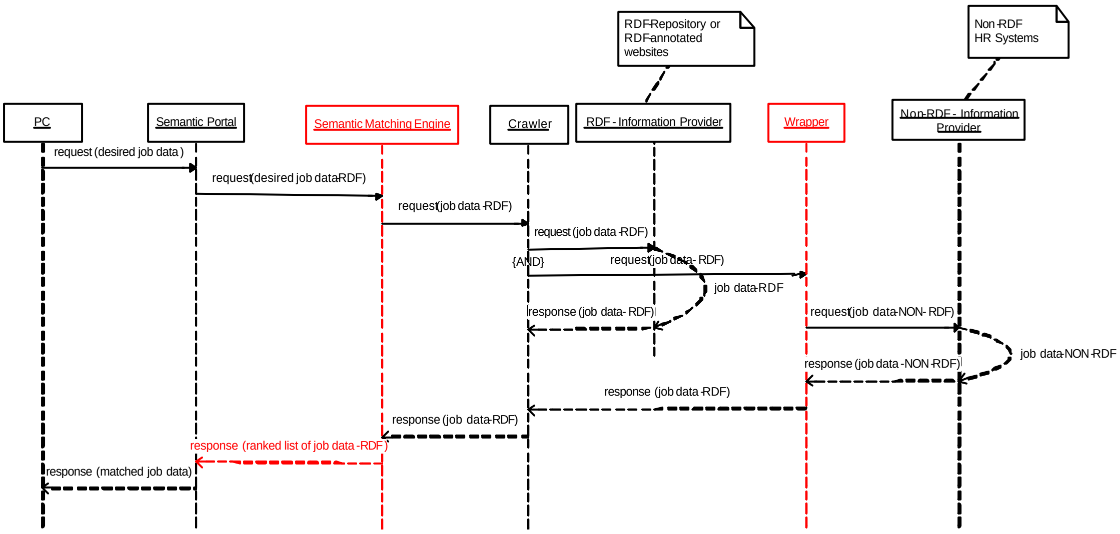

Fig.2.1.5 Flow of events: search for vacancies/applicants in external data technical use case

Technology locks identification: Technology locks are marked in red in Figure 2.1.5. In addition to the locks mentioned in the previous use case, we identify here *Semantic Matching Engine* and *Wrapper* as technology locks. Semantic Matching Engine appears hear as a lock, since there is no more assumption of the use of a common ontology. Thus, matching engine should determine correspondences between descriptions of applicant's qualifications and vacancies coming from heterogeneous sources. Wrapper enables the system for a translation/exchange of data instances with the heterogeneous data sources.

Ontology management

Actors: Administrator.

Summary: The actor wants to modify the recruitment ontology being used by the system.

Preconditions and inputs:

- There is a modification to be introduced in the recruitment ontology.

Post-conditions and outputs:

- Either the recruitment ontology has been changed or an error message has been generated indicating that the change can not be allowed.

The flow of events for the *ontology management* technical use case is presented in Figure 2.1.6.

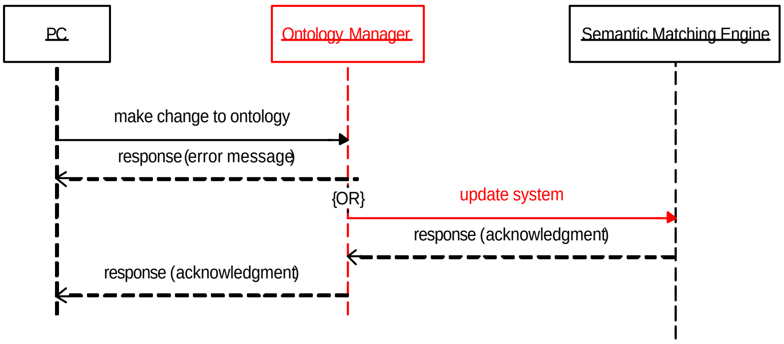

Fig.2.1.6 Flow of events: ontology management technical use case

Let us describe events of Figure 2.1.6 in more detail. Italics indicate optional events in the use case.

Actor	Response
Administrator identifies a change that needs to be made to the recruitment ontology Administrator tries a different change in the ontology to avoid a logical error	The change is made with the help of ontology management tool *The tool guides the administrator in carrying out the change* EITHER an error message is generated if the change can not be permitted (e.g. it makes the ontology logically inconsistent) OR the updated ontology is loaded into the system for use with the next system activity and an acknowledgement is sent to the administrator

Technology locks identification: Technology locks are marked in red in Figure 2.1.5. We identify here the *Ontology Manager* and *updating the system* with the changes to be introduced in the ontology as a technology lock. An Ontology Management tool must be able to support the administrator in maintaining the ontology without requiring a specialized knowledge of ontology modeling and its representation in a given knowledge representation (KR) formalism. Rather, if ontology-based systems are to be administrated and maintained over the long term, tools should simplify the process by being able to guide their users to model the domain knowledge correctly and consistently. The tool must also identify inconsistencies and prevent them from being inserted into the system. Updates of the system by changing the ontology must be able to take place with as little manual overhead as possible (i.e., avoiding re-coding of the application), while taking into account how changes will affect tasks such as querying, matching, and ranking.

2.1.4 Design

2.1.4.1 The use case knowledge processing tasks

Having identified major technology locks of the Recruitment system, now we are able to state knowledge processing tasks required in order to develop plausible Semantic Web solutions to those technology locks. These tasks are listed below and are described with the help of examples. Also for each task we refer to the system use where it has to be executed.

Data translation is a task of translating data from different information sources into RDF exploiting methods which are able to preserve semantics of an information source. For example, there is a large body of recruitment data being produced currently using the HR-BA-XML (human resources) vocabulary. While this data uses a commonly agreed set of terms in describing jobs and job applicants, it is expressed in XML, and hence it can not be suitably manipulated, with enough details describing semantics of the data, as required in the use case. Therefore, a data translation task aims at taking this data as input and producing RDF conforming to the recruitment ontology (using a mix of direct term equivalence and natural language parsing) [16,33] as output, such that the added knowledge-based functionalities would be made available.

We see this data translation occurring in the *Wrapper* component in the sequence diagrams.

Ontology management [31,38,7,6] is a task of maintaining the base ontology used in the recruitment system. As the job market or aspects of the recruitment domain such as qualifications alter, the ontology might evolve and has to be realigned with the other ontologies. For example, with a globalization of the job market, recruitment applications might be submitted from new countries which have different educational systems. Higher level qualifications must be identified within the system and related to existing qualifications such that the applications from new countries could be matched to vacancies.

We see this ontology management occurring in an *Ontology Manager* component in the sequence diagrams, accessible only to the Administrator actor in the use case.

Matching. There are two matching tasks in this use case; however it can be argued that both of them can be viewed as the same graph matching problem [15]. The first task is of discovering relationships (e.g., equivalence, less general and so on) between the entities (e.g., classes, properties) of the system ontology and external relevant schemas/ontologies. Good surveys on schema/ontology matching are provided in [19, 34,37]. This matching task can be considered as an ontology management subtask.

The second and the most important matching task in this use case is of finding similarities between the description of an applicants' qualifications, work experience and the description of a vacancies' requirements (also in terms of qualifications and work experience). Additional factors may also be taken into account (e.g., requirement of a driving license). Matching might be performed with the help of a common ontology (therefore, there is no more semantic heterogeneity problem) or without it. For example, a requirement for Java programming skills may be matched against C++ programming skills (as "similar"), or the requirement for health care experience with previous work in a hospital (as "possibly relevant").

We see ontology matching occurring in the *Semantic Matching Engine* component in the sequence diagrams.

Ranking matching results is a task of ordering matching results according to a desired criterion. The complexity of qualifications and work experience mean that exact matches between job requirements and applicants are unlikely to happen; rather a ranking mechanism is used to express the extent to which the equivalence might be assumed. This differentiating mechanism of matching results was indicated in the previous paragraph referring terms as "similar" and "possibly relevant". The ranking is used to determine which search results are delivered first to the actor (e.g., the most relevant matches to the query).

We see this matching result analysis occurring in the *Semantic Matching Engine* component in the sequence diagrams, and is indicated by stating that the response from a search activity is a "ranked list" of results.

Schema/ontology merging [26] is a task of integrating other ontologies into an existing ontology. The use case supposes exploiting of a single ontology. However, in a decentralized distributed environment such as Semantic Web it is reasonable to expect existence of multiple ontologies, even on the same topic. Some of these ontologies might be useful for extending the recruitment ontology (e.g., the ontology for computer programming languages integrated with the other ontologies on the same topic would be a useful extension for matching among programming skills). In this case, the user (the administrator) would need to (semi-automatically) identify how the concepts in the imported ontology relate to the concepts in the existing ontology.

We see ontology merging occurring in the *Ontology Manager* component in the sequence diagrams. This task can be considered as an ontology management subtask.

Producing explanations. When web applications return answers, many users do not know what information sources were used, when they were updated, how reliable the source was, or what information was looked up versus derived. Let us consider the example of matching systems. State of the art matching systems (e.g., QOM [10], OLA [12], COMA [9], Cupid [24], S-Match [15,14]) perform well for many real world applications. However, matching systems may produce mappings that may not be intuitively obvious to human users. In order for users to trust the mappings (and thus use them), they need information about them. They need access to the sources that were used to determine semantic correspondences between terms and potentially they need to understand how deductions are performed. The issue here is to present explanations in a simple and clear way to the user. One possible solution for producing explanations is to use the Inference Web [27] framework. An example of how a matching system can be explained is provided in [28].

Explanations help actors to make informed decisions on how a result was obtained, e.g., a job vacancy or a job applicant, fulfils the necessary requirements determined relevant by the system in respect to the actor's request.

This could be included within the query response of the *Semantic Matching Engine* in the sequence diagrams.

2.1.4.2 The use case typology of knowledge processing tasks

We build a typology of knowledge processing tasks by splitting all the knowledge processing tasks in to primarily and secondary tasks according to the business logic of the system. Let us consider them in turn.

Primary Tasks

- Data translation. This task is necessary to ensure syntax and (partially) semantic homogeneity.
- Ontology management. This task is necessary to keep the ontology-based recruitment process functionality up-to-date.
- Matching. This task is necessary to determine “correspondences” between vacancies and jobseekers (with possibly heterogeneous descriptions).
- Ranking matching results. This task is necessary to highlight the best matches in order to facilitate efficient recruitment.

Secondary Tasks

- Merging. This task may be required when other ontologies which are (partially) relevant to the recruitment domain are to be re-used.
- Producing explanations. This task may be required when users desire to see how a search result was determined to be relevant.

2.1.4.3 The use case library of high level components

Table 2.1 briefly summarizes the use case typology of knowledge processing tasks and their corresponding high level components.

Let us discuss in detail each of the components presented in Table 2.1 in terms of their inputs and outputs, leaving the algorithms they have inside as a black box, because these issues are irrelevant for the goals of the report. We refer an interested reader to the research work packages for the algorithmic part, for example a survey of matching (alignment) algorithms can be found in WP2.2, D2.2.3.

Wrapper translates the input data such that it appears within the system as RDF files produced with the help of the recruitment ontology. It acts as an interface to the input data such that both requests from and responses to the system may be expressed in the RDF vocabulary while the underlying data continues to be stored in its original format, see [32,33].

Ontology manager is a tool that allows an ontology expert to modify the recruitment ontology (adding and removing concepts and properties or changing their values). It provides functionalities to identify inconsistencies, classify new concepts, and to import/merge other ontologies. To facilitate ontology maintenance, the ontology manager should be as easy to use as possible. It should provide a graphical interface for intuitive ontology visualization and modification.

The querying and matching functionality of the Semantic Matching Engine may be tied to the recruitment ontology. Hence, it is important to take into consideration how ontology level changes made in the ontology manager may affect the component functionality and how this may be resolved with a minimal amount of manual effort.

Table 2.1 Use case 1. Knowledge processing tasks & components

	Knowledge processing tasks	Components
Primary tasks	Data Translation	**Wrapper ("2RDF)**
	Ontology Management	**Ontology Manager**
	Matching	**Match Manager**
	Ranking Matching Results	**Match Manager**
Secondary tasks	Schema/Ontology Merging	**Ontology Manager**
	Producing Explanations	**Match Manager**

Match manager coordinates the matching process using in the appropriate manner available matching algorithms, sources of auxiliary information, etc. Match manager supports two primarily functionalities: (i) match the input data or conceptual models/descriptions of vacancies or applicant's qualifications (*Match* operator), and (ii) rank the matching results according to a given criteria (*Rank* operator). The third (secondary) functionality concerns producing explanations of matching results.

Match is an operator which takes two graph-like structures (e.g., ontologies or descriptions of vacancies or applicant's qualifications) as input and returns a *similarity relation* between the nodes of the graphs that "correspond" semantically to each other as output. A similarity relation can be either in the form of *coefficient* in the [0,1] range, rating match quality (i.e., the higher the coefficient, the higher the similarity between the nodes, see [11,10,30,29] for some particular implementations) or in the form of a *semantic relation* (e.g., set theoretic relations, *equivalence*, *more general, less general, overlapping*. See [15,14,3]).

Rank is an operator which takes as input a set of mappings, i.e., pairs of nodes and similarity relations holding between them, and criteria according to which the mappings are to be filtered; and returns a list of mappings as output. A typical criteria used in the NLP community is that of the closeness of two concepts in an 'is-a' taxonomy [36], which has been extended into new approaches for the Semantic Web [25].

The explanation module takes as input a mapping and produces as output simple natural language statements which describe the mapping to a user.

2.2 Use Case 2. Multimedia content analysis and annotation by *CERTH*

2.2.1 Use case summary

The UML use case diagram in Figure 2.2.1 shows how the aceMedia[1] system exploits knowledge inherent to multimedia content in order to enable new services. The example is of a user (a school student) who aims at creating a retrospective in the form of a multimedia presentation. In order to achieve this goal, the user needs the functionalities of (i) multimedia search and retrieval, and (ii) automated information organization. Additionally, if some content involves digital rights, then it must be paid for.

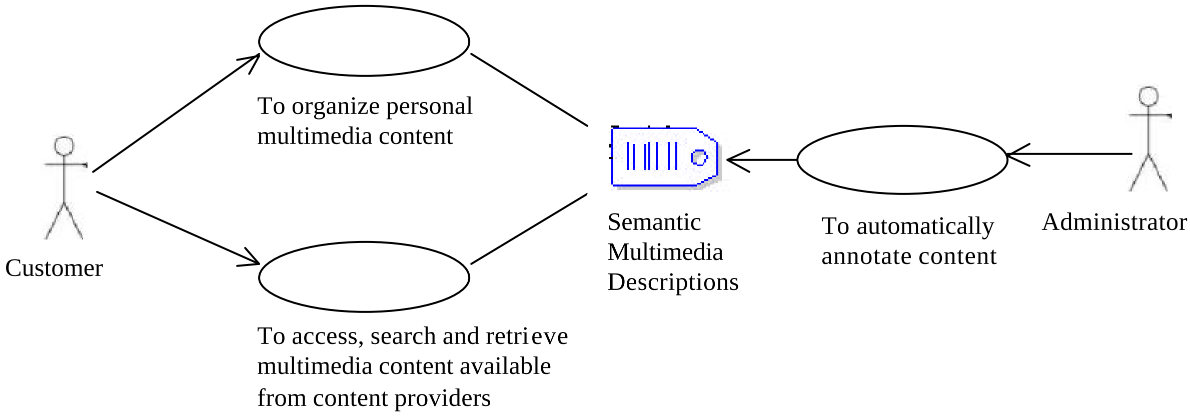

Fig.2.2.1 UML use case diagram for multimedia analysis and annotation

Now, based on the functional requirements, it is possible to specify also different technical use cases taking part in the system. Those use cases will next allow for a detailed analysis of the technical needs.

2.2.2 Service requirements

Technical use cases diagram is presented in Figure 2.2.2. It allows identifying technical use cases and associated actors.

Let us discuss actors of Figure 2.2.2. The aceMedia system consists of two main parts, namely, the *online system* where the main actor involved is the user; and the *offline system* where the two actors involved are the content providers and the administrator.

Offline system: Administrator, Content Providers. The aceMedia administrator is responsible for the automatic annotation of the content provided by the content providers.

Online aceMedia system: User. The user with the help of its access interface (e.g., PC, mobile device) can access the services offered by the online aceMedia system.

---

[1] FP6-001765 aceMedia: Integrating knowledge, semantics and content for user centred intelligent media services (URL: http://www.acemedia.org)

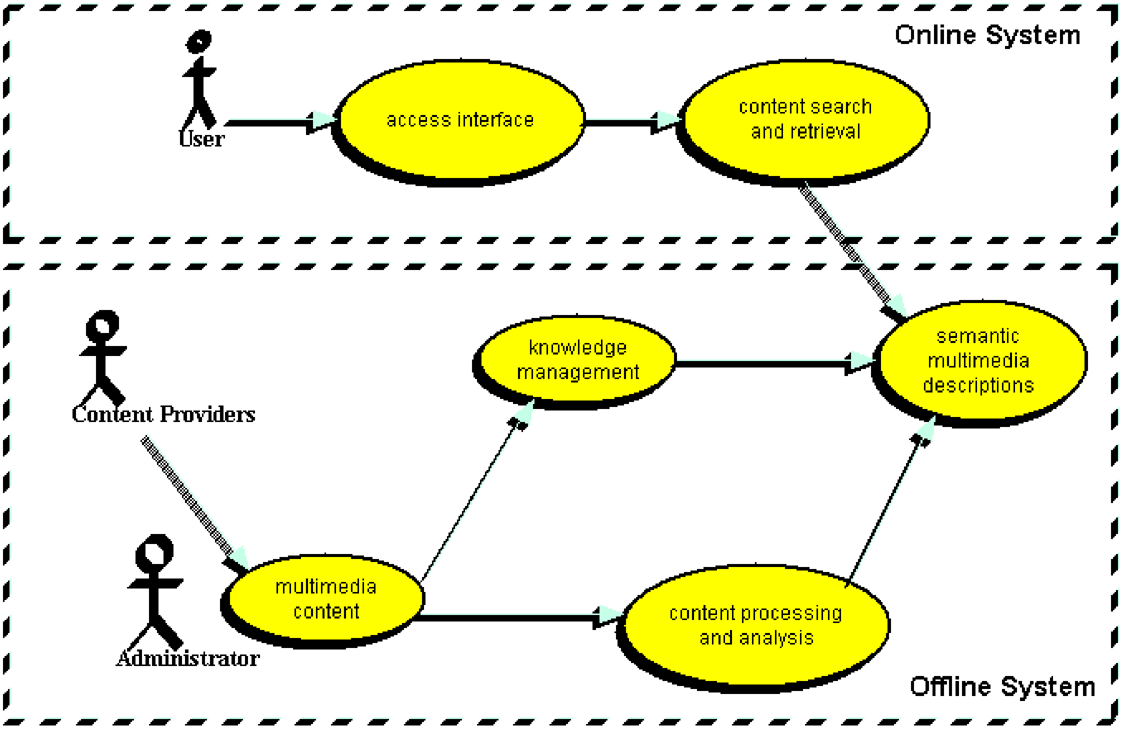

Fig.2.2.2 UML technical use case diagram for multimedia analysis and annotation

2.2.3 Analysis

Now we analyze technical use cases of Figure 2.2.2 in detail. In particular, we identify two major technical use cases. We first consider the offline process through which the semantic descriptions of the available content are produced (*Multimedia Content Analysis and Semantic Annotation* technical use case). Then we discuss the online process that enables the aceMedia user to access the available content based on the intelligent search and retrieval mechanisms, as well as the personalization and media adaptation services provided by the system (*Content Search and Retrieval* technical use case).

For each technical use case we first report the actors it involves, then we provide its summary, inputs and outputs, and finally we discuss with the help of sequence diagrams the flow of its events and possible technology locks.

Multimedia content analysis and semantic annotation

Actors: System administrator, content providers.

Summary: Knowledge-assisted multimedia analysis is initially performed to produce the annotation of the content provided by the content providers. Then, a reasoning process takes as input the automatically annotated content, usage-based annotations provided by the user and other information sources (e.g., spatio-temporal relations) and produces semantic multimedia descriptions.

Preconditions and inputs:

- Descriptions of specific application domains (beach holiday, racing, etc.) should be available;
- Existence of content providers is assumed.

Post-conditions and outputs:

- A database consisting of the semantic descriptions of the available content.

The flow of events for the *multimedia content analysis and semantic annotation* technical use case is presented in Figure 2.2.3.

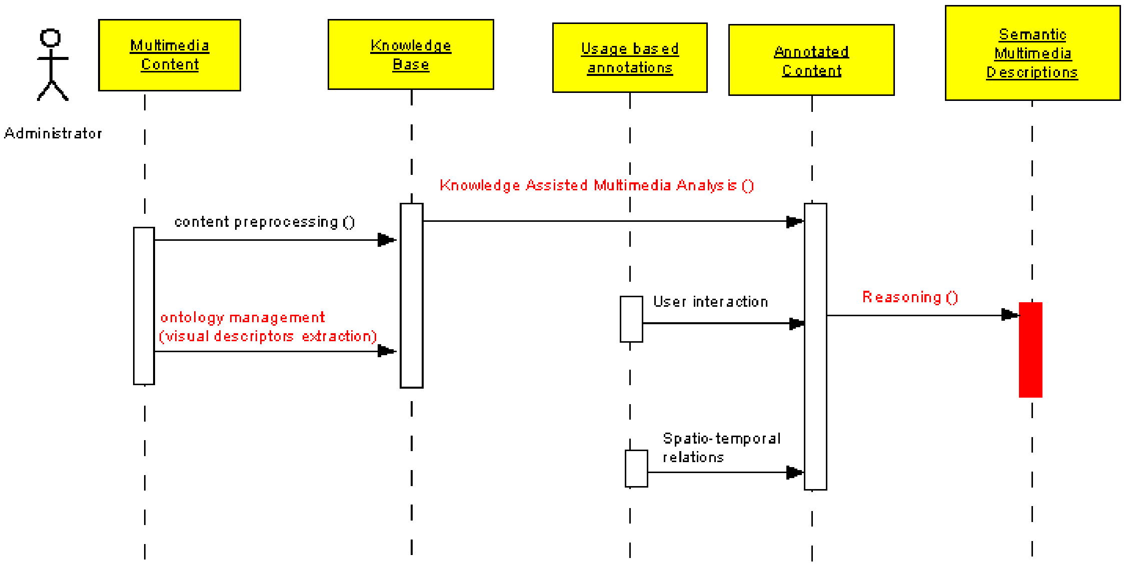

Fig.2.2.3 Flow of events: Multimedia content analysis and semantic annotation technical use case

Let us describe events of Figure 2.2.3 in more detail and in turn as they appear:

- The system is performing the initial content preprocessing (e.g., segmentation) and integrates the multimedia and domain ontologies (*ontology management*). The system knowledge base is populated by the descriptors of the visual objects included in the domain knowledge with the use of a *visual descriptors extraction*, VDE, tool (see for details Section 2.2.4.1 Ontology management task).
- *Knowledge-assisted multimedia analysis* is performed with the help of the knowledge base built in the previous stage.
- *Reasoning* has as input the annotated content, the spatio-temporal relations and/or user input. The reasoning process aims at checking the consistency of the objects detected in the content annotation process depending on their spatio-temporal relations. The output of the reasoning process provides high-level semantic multimedia descriptions.

Technology locks identification: Technology locks are marked in red in Figure 2.2.3. The main technology locks here are the *knowledge-assisted multimedia content analysis, ontology management* (including the extraction of low-level visual descriptors and their integration within the ontology structure), and *reasoning*.

Notice that the problem of integrating information from heterogeneous sources is not addressed in the aceMedia framework. This is because all information sources (i.e., the content providers) are subscribed to one global ontology.

An example motivating technology locks, e.g., reasoning, is as follows. Let us discuss the Formula One domain, where the automatic multimedia annotation process has produced for a particular image or video frame the detection of the objects "car" and "road", which are defined in the corresponding domain ontology. If these two objects do not satisfy the defined spatial relationship that the car is "above" the road, then this could lead to the generation of the semantic event "the car getting out of the road". Such semantic descriptions enable more sophisticated semantic querying of the multimedia content (images and video) in terms, which are more intuitive to the user.

Content search and retrieval

Actors: AceMedia user.

Summary: Based on the semantic multimedia annotations produced by the offline system, the online system supports matching of textual, ontology-based and audiovisual metadata, thus enabling the aceMedia user to perform a hybrid search. The hybrid query consists of the semantic query (i.e., the ontological representation of the user's textual query) and the query by visual example (where the user uses an external image and looks for visually similar images). The final result of the query (i.e., images and videos) returned to the user also takes into account personalization and media adaptation issues.

Preconditions and inputs:

- Availability of the semantic multimedia descriptions produced by the offline part of the system.

Post-conditions and outputs:

- A set of images and videos that will be returned to the user in response to his hybrid search.

The flow of events for the *content search and retrieval* technical use case is presented in Figure 2.2.4.

Let us describe events of Figure 2.2.4 in more detail and in turn as they appear:

- The user can perform either a query by *visual example* (e.g., by providing a sample image and asking the system to return similar ones) or a textual query specified in *natural language* (e.g., "give me the players wearing red") that needs to be semantically interpreted.
- In the case of a query in natural language, textual query analysis techniques are applied in order to produce the ontological representation of the textual query.
- The *hybrid query* consists of two parts: the semantic query and the query by visual example.
- *Intelligent search and retrieval* is then performed combining both ontology-based and audiovisual (e.g., low-level features) metadata in order to produce an initial set of results (e.g., images and videos).

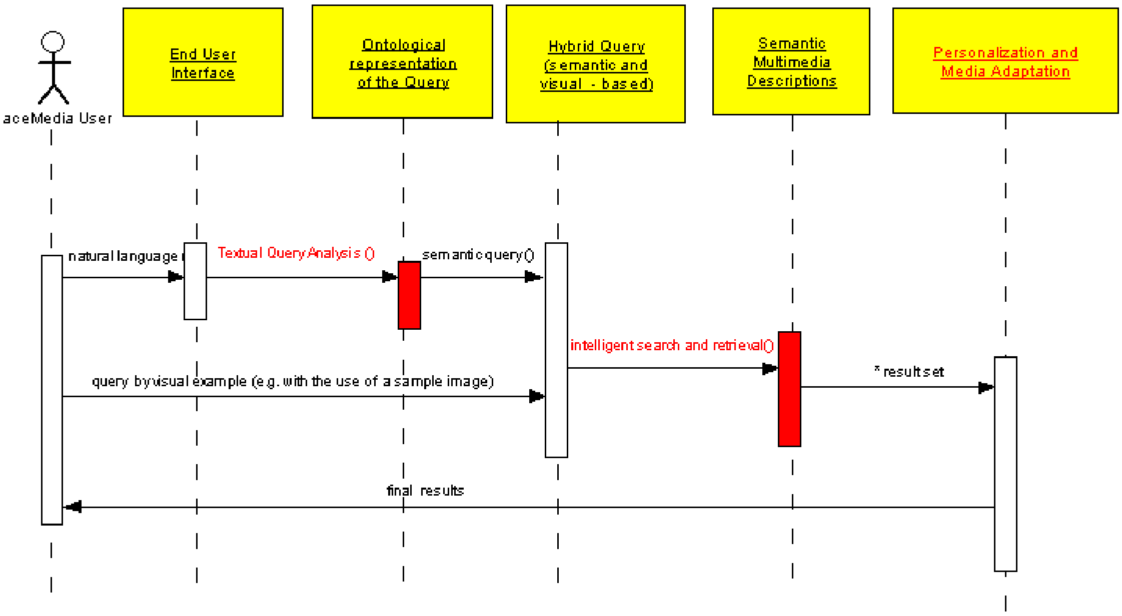

Fig.2.2.4 Flow of events: Content search and retrieval technical use case

- The *final result* (e.g., images and videos) is also a subject of media adaptation and personalization conditions before being delivered to the user's terminal.

Technology locks identification: Technology locks are marked in red in Figure 2.2.4. The main locks here are the *textual query analysis* that translates the user' requests into language independent ontological representations and processing the hybrid query with *intelligent search and retrieval* mechanisms. Finally, the last (minor in the business context of the system) lock is *personalization and media adaptation* process which acts as a filter that prunes out the irrelevant results from the perspective of the user.

Some examples motivating the locks are as follows. As long as the user can submit the query as a natural language sentence, then there is a need of interpreting terms used in the query meaningfully (e.g., with the help of the domain ontology), see [20,17,22]. Also the use case shows that users may want to pose queries either by specifying abstract notions such as the name of an event or an activity (e.g., give me the player who won this game) or by specifying visual patterns (e.g., indicating one or more sample images and then looking for similar images), see [4,5,13]. Therefore, the system must be able to deal with both types of queries.

2.2.4 Design

2.2.4.1 The use case knowledge processing tasks

Having identified major technology locks of the aceMedia system, now we are able to state knowledge processing tasks required in order to develop plausible Semantic Web solutions to those technology locks. These tasks are listed below and are described with the help of examples. Also for each task we refer to the system use where it has to be executed.

Content annotation. Using the multimedia ontology infrastructure and domain knowledge captured in ontologies, the knowledge-assisted analysis of audiovisual content will result in the generation of semantic metadata. Thus, for example, a user may search for multimedia content using high-level concepts like "holiday", "beach", "racing cars" or events by asking for a "player scoring a goal".

In addition to the automatic production of content metadata, interaction with the user and prompt mechanisms will also be developed to enable the user to enrich the knowledge base with additional metadata that cannot be automatically detected. For example, the user will be prompted to annotate the visual content with metadata such as title, date (when shot), name of a location, title of an event, names of people, names of pets, comments, subject fields. Besides, for commercial usage scenarios the user will be prompted to give additional information related to date of creation, date of production, location, cameraman name, director name, perspective, camera parameters, lighting.

This task appears in Figure 2.2.3. Interaction with the user appears in the same diagram as usage-based annotations component.

Ontology management. Ontology management in aceMedia project is mainly related to the task of maintaining domain ontologies with multimedia features. A visual descriptor extraction module of the system enables experts to extract the audiovisual features of visual objects and to assign them to the semantically equivalent concepts defined in the domain ontologies. The *Multimedia Ontologies* model the domain of multimedia data, especially the visualizations in still images and videos in terms of low-level features and media structure descriptions. The structure and semantics are carefully modeled to be largely consistent with existing multimedia description standards, such as MPEG-7.

aceMedia has already designed an appropriate knowledge representation approach for multimedia and is building the tools allowing extraction of visual descriptors and linking them to the domain ontologies.

The ontology management task appears in Figure 2.2.3.

Reasoning with annotations. The objective of this task is the development of reasoning tools that will handle the metadata generated by the knowledge-assisted multimedia analysis and will use the domain ontologies to generate high-level, semantic representations of aceMedia content. The reasoning process will check the consistency of the content annotation process against a set of spatial (e.g., left, right, above, adjacent, overlaps) and temporal (e.g., before, after, during, co-start, co-end) relations that will be defined in order to ensure that the objects detected in the multimedia content correspond semantically to the objects defined in the domain ontologies, [1,8,21]. For example, in the racing domain it should be checked whether the car is "above" the road or whether the grass and sand are "adjacent" to the road. The use of spatio-temporal relations also allows for the definition of semantically important events that might be interesting to the user in the corresponding domains. For example, the reasoning process could be used for the detection of events like "one car overtaking another" and "the car getting out of the road" in the racing domain or "the player scoring a goal" or "the player falling down" in the sports domain.

This task appears in Figure 2.2.3.

Intelligent search and retrieval. In aceMedia, search of multimedia content should be based on the development of tools that support matching of both semantic (ontology-based) and audiovisual (low-level) metadata. Semantic query processing supports textual queries that need to be semantically interpreted. For example, in the tennis domain, the system should be able to process queries such as “give me all games played on grass” or “give me all games of double players”.

In addition to the semantic query processing, there should be a user-friendly interface that will allow users to provide an example image and then the system will be able to perform search by similarity using various criteria including: same objects contained; same event or type of activity; same persons occur; same location; similar visual appearance; same/similar color scheme or same background.

This task appears in Figure 2.2.4 and refers to the hybrid (semantic and visual-based) user query processing.

Personalization and media adaptation. The aceMedia system should adapt its behavior to user preferences and profiles. For example, users may want to set restrictions on access to all their contents, query history, or to use standard or professional profile. Therefore, a model of user preferences and profiles suitable for aceMedia adaptivity is to be devised. As an additional means of personalization, the interface of aceMedia shall offer end users the option to annotate content by metadata of their own, and to share annotations and bookmarks within trusted user networks (personal metadata management and contact recommender). Media adaptation includes a delivery of contents to the user’ terminal (e.g., palm) suitably manipulated by exploiting semantic techniques.

This task appears in Figure 2.2.4.

2.2.4.2 The use case typology of knowledge processing tasks

We build a typology of knowledge processing tasks by splitting all the knowledge processing tasks in to primarily and secondary tasks according to the business logic of the system. Let us consider them in turn.

Primary Tasks

- Ontology management. This task is necessary to instantiate the domain and multimedia ontologies in order to use them in the knowledge-assisted analysis process.
- Content annotation. This task is necessary to produce the semantic metadata for the user.
- Reasoning. This task is necessary for the extraction of high-level consistent annotation for the content.
- Intelligent search and retrieval. This task is necessary for the semantic and visual based (hybrid) search.

Secondary Tasks

- Personalization and media adaptation. This task may be required for the adaptation of the system to the user preferences, network resources and device capabilities.

2.2.4.3 The use case library of high level components

Table 2.2 briefly summarizes the use case typology of knowledge processing tasks and their corresponding high level components.

Let us discuss in detail each of the components in terms of their inputs and outputs, leaving the algorithms they have inside as a black box, because these issues are irrelevant for the goals of the report.

Table 2.2 Use case 2. Knowledge processing tasks & components

	Knowledge processing tasks	Components
Primary tasks	Content Annotation	**Content Annotation Manager**
	Ontology Management	**Ontology Manager**
	Reasoning with Annotations	**Annotation Reasoner**
	Intelligent Search and Retrieval	**Hybrid Query Processor**
Secondary tasks	Personalization and Media Adaptation	**Personalization and Media Adaptation**

Ontology manager. The ontology manager is the offline component responsible for the extraction of a set of low-level visual descriptors for different domain concepts and their integration into the ontology structure. This component is responsible for building up the knowledge base, which will be used as input to the content annotation manager. The inputs of this component are the domain ontologies and the multimedia content (images and videos). The output of this component is a set of low-level visual features, which are extracted for the different objects defined in the domain ontologies with the help of the visual extraction tool and the domain experts.

Content annotation manager. This is an offline component responsible for performing the knowledge-assisted multimedia analysis. The input of this component is the processed audiovisual content and the domain knowledge. The output is the database of content annotations.

Annotation reasoner. This is an offline component. It has as input annotated content and/or user input, and additional information such as spatio-temporal relations. Then, it checks the consistency of the output of the content annotation manager, i.e., whether the objects detected during the automatic content annotation process correspond to the semantic objects defined in the domain ontologies. In addition, the reasoner detects semantically important events represented in the multimedia content (e.g., the event of "scoring a goal"). The output of this component is the consistent database of the semantic multimedia descriptions.

Hybrid query processor. This is the online component. It takes as input the semantic query and the query by visual example and uses the semantic multimedia descriptions database to perform the hybrid (semantic and visual) search in order to output the initial result set (images and videos) in response to the user' query.

Personalization and media adaptation module. This is the online component. It has as input the initial set of results returned from the hybrid (semantic and visual) query by the user. The output of this component is the final set of results (images and video) returned to the user and tailored to its profile.

2.3 Use Case 3. B2C marketplace for tourism by *FT*

2.3.1 Use case summary

The main two uses of the B2C market place system for tourism are summarized in Figure 2.3.1.

The first use case, which is called "to plan a nice week-end", constitutes the entry point inside the market place allowing customers to define their personal needs. The platform takes care of identifying potentially useful contents and services, accessing multiple providers and selecting the only relevant ones.

The second use case, which is called "to package and purchase a nice week-end", requires (i) a dynamic aggregation of relevant contents and services (transport, accommodation, leisure activities, etc.), (ii) an automated packaging of week-end proposals, and (iii) facilities of purchasing them on-line.

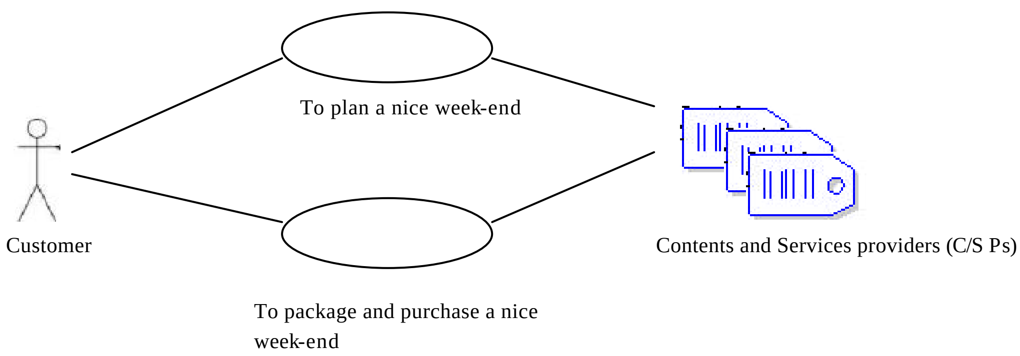

Fig.2.3.1 UML use case diagram for B2C market place for tourism

Now, based on the functional requirements, it is possible to specify also different technical use cases taking part in the platform. Those use cases will next allow for a detailed analysis of the technical needs.

2.3.2 Service requirements

Technical use cases diagram is presented in Figure 2.3.2. It allows identifying technical use cases and associated actors. Let us discuss actors of Figure 2.3.2.

Customer and Access Interface. A customer with the help of its access interface (e.g., mobile phone) accesses services available within the system through the authentication mechanism, personalization, and session management.

Contents and Services providers (C/S Ps). Contents and services providers manage their offers autonomously, i.e., the system does not impose any constraints. Each contents and services provider has its own rules for structuring information at the protocol, syntactical, and semantic levels.

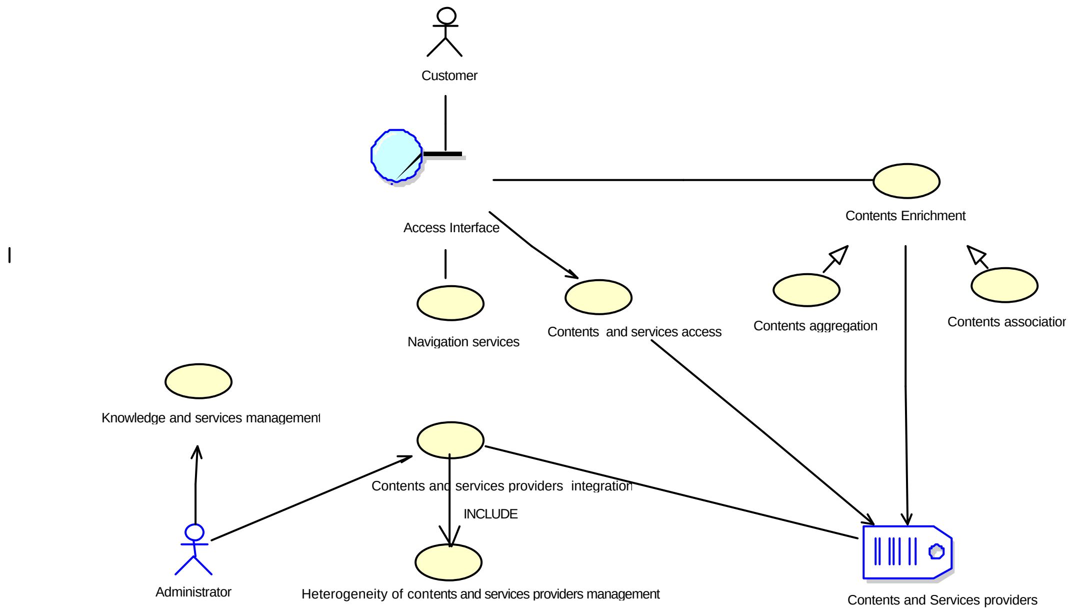

Fig.2.3.2 UML technical use case diagram for B2C market place for tourism

Administrator is in charge of the administration of the platform. It performs (i) referencing of new contents and services providers, (ii) internal knowledge representation and management, and (iii) orchestration between different contents and services providers.

2.3.3 Analysis

Now we analyze each technical use case of Figure 2.3.2 in detail. In particular, we consider *contents and services access*, *contents aggregation*, *contents association*, *knowledge and services management*, *content and services provider's integration*, *heterogeneity of contents and services provider's management*, *and knowledge and services management* technical use cases. The technical use case *navigation services* is not analyzed, since it does not contain any technology locks and it does not obscure further discussions.

For each technical use case we first report the actors it involves, then we provide its summary, inputs and outputs, and finally we discuss with the help of sequence diagrams the flow of its events and possible technology locks.

Contents and services access

Actors: Customer and access services, C/S P Service.

Summary: A customer request "next week-end I'm going to Brittany" is submitted to the system that checks possible (correct) interpretation within the tourism domain and accesses the requested services or contents.

Preconditions and inputs:

- A domain description is available;
- The contents and services providers are registered in a catalogue or directory.

Post-conditions and outputs:

- The contents and services are located. Only the relevant ones are selected in a limited (mastered) processing time.

The flow of events for the *contents and services access* technical use case is presented in Figure 2.3.3.

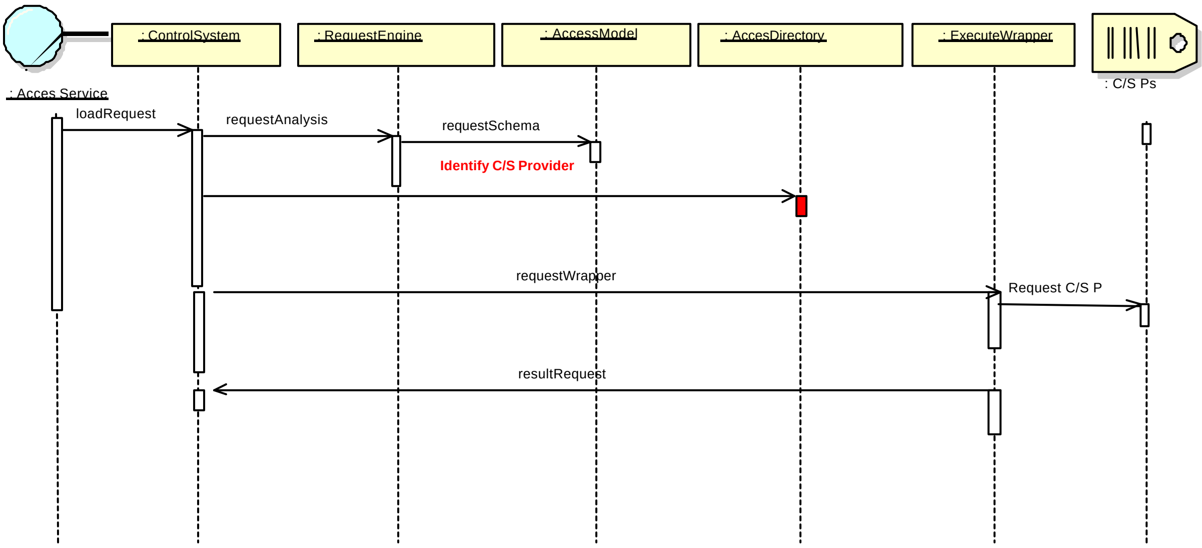

Fig.2.3.3 Flow of events: Contents and services access technical use case

Let us describe events of Figure 2.3.3 in more detail and in turn as they appear:

- The request is reformulated (*requestAnalysis*) to conform to the internal knowledge representation format and is further processed (*requestSchema*) with the help of knowledge-based techniques against the domain knowledge (e.g., tourism) available in the system;
- A list of C/S Ps is identified (*identify C/S Provider*);
- A query plan generation is processed (*requestWrapper*) and executed on the appropriate C/S P (*requestC/S P*);
- The contacted C/S P returns instance data to the system (*resultRequest*).

Technology locks identification: Technology locks are marked in red in Figure 2.3.3. The key lock here is in the detection of relevant contents provider(s) from the customer request before accessing the C/S Ps directory. The problem is to determine whether a directory would be able to first *centralize* all C/S Ps of a particular domain and second to *match* semantically a user' request against some of the C/S Ps. For example, if a user is looking for a gastronomic restaurant in Berlin, the directory should be able to identify C/S Ps dealing with restaurants in Berlin, and more accurately with only gastronomic restaurants, thus, excluding McDonalds, etc.

Contents aggregation

Actors: Customer and access services, C/S P Service.

Summary: The use case contents aggregation is inherited from the use case combine contents. The contents aggregation must carry out the fusion of the same information issued by different C/S Ps. The aim is to propose to the user as richest as possible information with the following characteristics:

- No duplication and redundant information;
- A homogenous information structure;
- Avoid the user having to aggregate the contents issued from different C/S Ps.

The global scheme, which is the model for common data for all the C/S Ps, captures the knowledge of the domain.

Preconditions and inputs:

- The use case contents and services access has been executed;
- The contents are expressed in the global scheme.

Post-conditions and outputs:

- The restructured contents are transferred to the access interface.

The flow of events for the *contents aggregation* technical use case is presented in Figure 2.3.4.

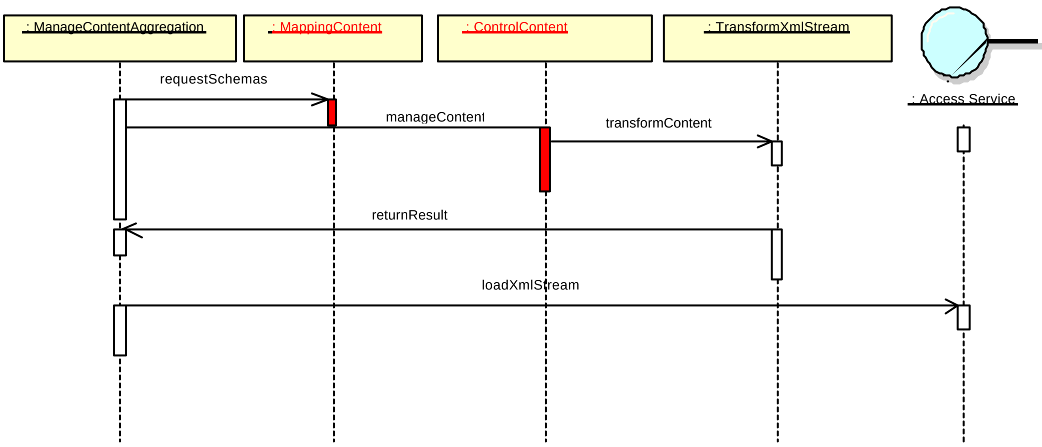

Fig.2.3.4 Flow of events: Contents aggregation technical use case

Let us describe events of Figure 2.3.4 in more detail. Before operating the contents aggregation, the system (*ManageContentAggregation* component) needs to compare the data (expressed in different data models) of each C/S P involved in the processing of the user request. This step is essential in order to evaluate the contents of each C/S P, and hence, detect redundancies, complementary information, etc. The flow of events is as follows:

- Identification of the mappings between different data models (*requestSchemas*);
- Content aggregation (*manageContent*): check for duplicated information, fusion of complementary information operated by the component *ControlContent*;
- Transformation of the result of content aggregation inside a XML formalism;
- XML data flow transfers to the access service (*loadXmlStream*).

Technology locks identification: Technology locks are marked in red in Figure 2.3.4. It is crucial to be able to *dynamically discover mappings* between the contents of different C/S Ps. This step is essential before the contents aggregation. The current solution follows the data integration approach which is to create static correspondences between data models [23]. However, this solution it is not scalable. Thus, the question is how to determine those correspondences dynamically. For example, given two XML schemas, suppose in the first schema the address element consists of the name, town, and postcode attributes, in the second schema the address element is split down into three sub elements: street name, post code and town. Then, a solution should be developed in order to determine correspondences between the semantically related entities, e.g., the address element in the first schema should be mapped to the address element in the second schema. A more complex solution is required to determine that attributes of the first schema are to be mapped (notice!) to the elements of the second schema.

Contents association

Actors: Customer and access services, C/S Ps.

Summary: The use case contents association inherits the use case contents enrichment. The association between contents affects relations which can be established *naturally* (for example, an exhibition of pictures taking place in a museum) or *context-sensitively* (for example, a museum near where I am staying). The first type of combination is defined at a global schema level. The second type assumes management at a spatio-temporal level.

Preconditions and inputs:

- The use case contents and services access has been executed;
- The contents are related with the global schema.

Post-conditions and outputs:

- The contents and the associations created are transmitted to access services.

The flow of events for the *contents association* technical use case is presented in Figure 2.3.5.

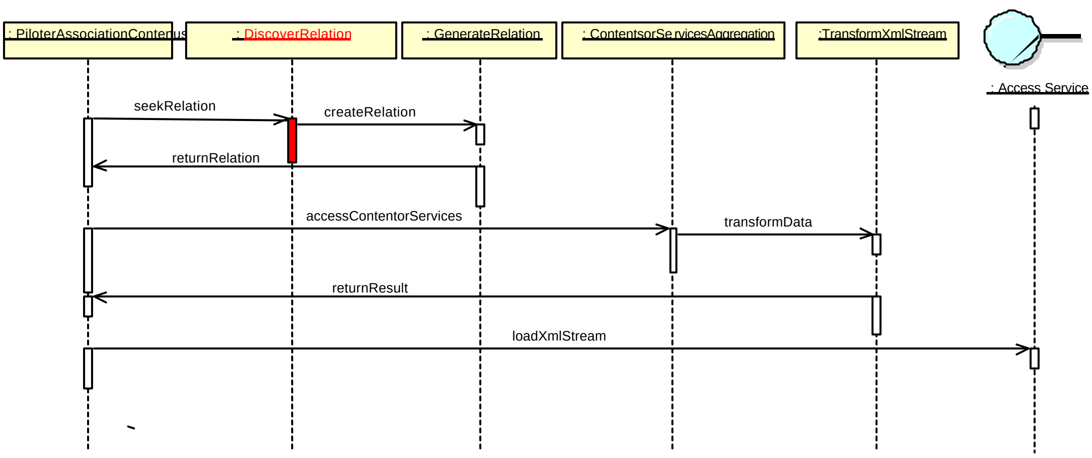

Fig.2.3.5 Flow of events: Contents association technical use case

Let us describe events of Figure 2.3.5 in more detail and in turn as they appear:

- Dynamic discovery of relationship between contents criteria (*seekRelation*);
- Generation of relationships (*createRelation*) in the context-sensitively case;
- Contents or services aggregation (*accessContentorServices*) from the relationships established;
- XML flow is transferred to the access service (*loadXmlStream*).

Technology locks identification: Technology locks are marked in red in Figure 2.3.5. The contents association is relatively obvious when it is defined at a global schema level. It differs substantially when contents association depends on the interpretation of the requested context. In the latter case the system must be able first to *deduce information about users* from their requests, second to *infer contextual relations* between concepts (essentially from spatio-temporal criteria and information about users). For example, from previous user requests, the system deduced that the user was attracted by a museum and more specifically by paintings. Thus, when visiting a city and looking for the presence of a museum, the system should inform the user of local exhibitions of paintings.

Knowledge and services management

Actors: Administrator.

Summary: There are two tasks of this use case. Let us consider them in turn.

Services administration. This task aims at specifying the orchestration process, i.e., a permitted chaining flow of the services and their associated pre/post conditions. For example, the selection of a travel package can initiate the payment procedure among different partners (e.g., hotel booking and leisure activities) through web services, the final validation starts when all the transactions have been already passed successfully.

Knowledge administration. This task aims at modeling and maintaining the domain knowledge of the portal.

Preconditions and inputs:

- The administrator knows the application domain;
- Existence of C/S Ps is assumed.

Post-conditions and outputs:

- The workflow is specified and verified.

The flow of events for the *knowledge and services management* technical use case is presented in Figure 2.3.7.

Let us describe events of Figure 2.3.7 in more detail. The knowledge and services management is essentially made up of two tasks. First, the system administrator models and maintains the domain knowledge of the portal (*updateModel*). Second, the system is able to dynamically specify the orchestration process (*orchestrationWorkflow*) between the different C/S Ps involved in the processing of the customer request.

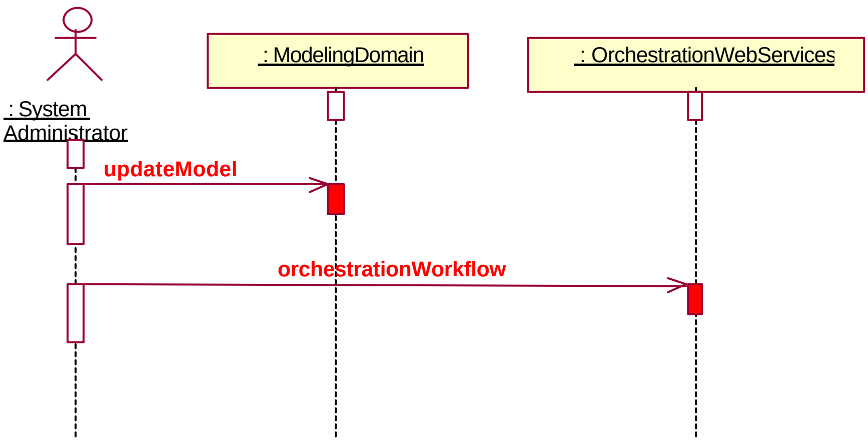

Fig.2.3.7 Flow of events: Knowledge and services management technical use case

Technology locks identification: Technology locks are marked in red in Figure 2.3.7. The dynamic building of the customer selection of elements composing its travel package entails the translation in a corresponding workflow for the web services orchestration. The problems are of (i) *generating dynamically a workflow* and (ii) *mediating possible and verifiable orchestration*. For example, booking a room in a hotel for holidays and renting a car for the same period require coordination between two different web services which have to understand themselves about the time period, prices and location according to the customer preferences. The other (minor in the business context of the system) technology lock includes management of the domain knowledge.

Contents and services providers' integration

Actors: Administrator, C/S Ps.

Summary: The administrator saves the new C/S Ps in the definition catalogue of the platform. Specific wrappers are automatically generated for each new C/S P.

Preconditions and inputs:

- C/S Ps have a structured or semi-structured dataflow for the contents;
- C/S Ps have a structure of web services description that is defined in a catalogue (e.g., WSDL file in UDDI);
- There is a catalogue, which lists C/S Ps.

Post-conditions and outputs:

- New C/S Ps are suitably entered into a catalogue;
- Wrappers are available for the new C/S Ps.

The flow of events for the *contents and services providers' integration* technical use case is presented in Figure 2.3.8.

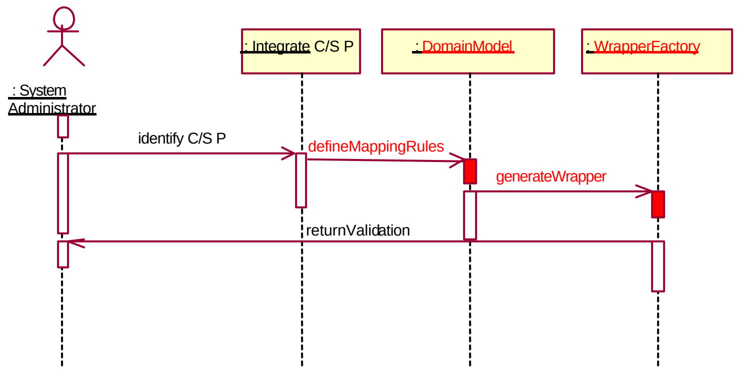

Fig.2.3.8 Flow of events: content and service provider integration

Let us describe events of Figure 2.3.8 in more detail and in turn as they appear:

- The system administrator have to integrate a new C/S P (*identify C/S P*);
- The component in charge of the integration (*integrate C/S P*) defines mapping rules between the global schema and the data model of the new C/S P (*defineMappingRules*);
- Once the mapping rules are defined, the system through the *WrapperFactory* component automatically generates a wrapper dedicated to this new C/S P (*generateWrapper*).

Technology locks identification: Technology locks are marked in red in Figure 2.3.8. This technical use case comprises two technology locks. The first is concerned with the capability of producing *mapping rules* between the global schema and the data model of a C/S P. Difficulties come from the fact that each C/S P uses its own terminology, its own data model, etc. The second lock is based on the difficulties of *automatic generation of a wrapper*. For example, a new C/S P on tourism (tourist sites with opening hours, prices, tourist descriptions, etc.) is entered into the catalogue. It could be very useful for customers to be aware of tourism capabilities near their hotels. However, integrating this new C/S P within the system requires aligning its terminology and data model with other C/S Ps in order to cooperate (e.g., translate/exchange the instance data) with others services and interpreting the input data uniformly (e.g., opening hours).

2.3.4 Design

2.3.4.1 The use case knowledge processing tasks

Having identified major technology locks of the B2C tourism marketplace system, now we are able to state knowledge processing tasks required in order to develop plausible Semantic Web solutions to those technology locks. These tasks are listed below and are described with the help of examples. Also for each task we refer to the system use where it has to be executed.

Planning of web services. This task aims at providing a flexible approach for connecting (composing) web services in order to create higher-level business processes. In this use case, planning of web services is needed when organizing a travel journey. In fact, the organization of a journey requires the "cooperation" of different C/S Ps (e.g., train, hotel reservations). This task appears in the *knowledge and services management* technical use case.

Global schema management. This is a task of maintaining the global schema expressing the domain knowledge of the B2C tourism marketplace system. It appears in the *knowledge and services management* technical use case.

Semantic query processing. This is a task of interpreting (rewriting) a query generated by a customer in terms of the global schema of the system. First, the system has to validate the request of the customer with respect of the global schema and then to rewrite it into queries against the C/S Providers' local schemas. A customer poses a query related to a tourism scenario into the system by a selection of terms from a predefined list that belongs to the global scheme. Then by means of concept's mapping, the user' input is reformulated first as a semantic query in terms of the tourism global schema and second as queries understandable by C/S Providers. This task appears in the *contents and services access* technical use case.

Mapping rules definition. This task aims at determining semantic relations (e.g., equivalence, less general, and so on) between the contents of C/S Providers and the global schema. This task appears in the *contents aggregation, contents association, and contents and service provider's integration* technical use cases.

Data translation. This task aims at translating data from different information sources exploiting (i) methods which are able to preserve semantics of an information source and (ii) mapping rules which state correspondences between local and global schemas. This task appears in the *contents and service provider's integration* technical use case.

Results reconciliation. This task is based on the ability of the system to detect redundancies and complementarities between data coming from different C/S Ps which are involved in the processing of a request. In the use case under consideration, this task prevents customers to encounter several responses about the same restaurants or to encounter different opening times information for the same museum. We distinguish between two types of reconciliation: aggregation as mentioned in the *contents aggregation* technical use case and association as mentioned in the *contents association* technical use case.

2.3.4.2 The use case typology of knowledge processing tasks

We build a typology of knowledge processing tasks by splitting all the knowledge processing tasks in to primarily and secondary tasks according to the business logic of the system. Let us consider them in turn.

Primary Tasks

- Mapping rules definition. This task is necessary to ensure semantic homogeneity.

- Data translation. This task is necessary to translate/exchange instance data between different C/S Ps.
- Semantic query processing. This task is necessary to enable customers to identify intelligently the pertinent C/S Ps.
- Planning of services. This task is necessary in order to offer high-level business processes involving several C/S Ps.
- Results reconciliation. This task is necessary in order to offer customers a way of accessing data, which is more accurate (e.g., avoiding duplications) than a basic search engine, such as, Google.

Secondary Tasks

- Global schema management. This task may be required in order to update the knowledge of the domain.

2.3.4.3 The use case library of high level components

Table 2.3 briefly summarizes the use case typology of knowledge processing tasks and their corresponding high level components.

Let us discuss in detail each of the components in terms of their inputs and outputs, leaving the algorithms they have inside as a black box, because these issues are irrelevant for the goals of the report. We refer an interested reader to the research work packages for the algorithmic part, for example a survey of matching (alignment) algorithms can be found in WP2.2, D2.2.3, while issues of composition of web-services is discussed in WP2.4, D2.4.2.

Match manager produces mapping rules. This module takes two data/conceptual models as input and returns a *similarity relation* between the entities of those models that "correspond" semantically to each other. A similarity relation can be either in the form of *coefficient* in the [0,1] range, rating match quality (i.e., the higher the coefficient, the higher the similarity between the nodes, see [11,10,30] for some particular implementations) or in the form of a *semantic relation* (e.g., set theoretic relations, *equivalence, more general, less general, overlapping*. See [15,14,3]).

Wrapper. Wrappers are software modules, each serving for one component data provider. The main task of a wrapper is to control and facilitate external access to the information providers through the local schema of the provider. A wrapper is in charge of reformulating the rewriting query of the customer in terms of the local schema of the C/S P and transforming responses returned by the C/S P into facts expressed in terms of the global schema.

Results reconciler. Sometimes, the answers the mediator returns are not what the user expects. This is the case when there are too many answers, no answers, or because some information is missing. Semantic Web technologies permit ordering answers logically, to removing irrelevant answers, etc.

Input of the results reconciler constitutes:

- The responses of each C/S P involved in the processing of the request.

Output of the results reconciler constitutes:

- A reconciled response.

Table 2.3 Use case 3. Knowledge processing tasks & components

	Knowledge processing tasks	Components
Primary tasks	Mapping Rules Definition	**Match Manager**
	Data Translation	**Wrapper**
	Results Reconciliation	**Results Reconciler**
	Semantic Query Processing	**Mediator**
	Composition of Web Services	**Planner**
Secondary Tasks	Global-schema Management	**Mediator**

Mediator is a module which provides a uniform query interface to a collection of distributed and heterogeneous information sources. This interface enables users to focus on specifying their demand by freeing them from having to find the relevant provider and possibly combine data from multiple providers. A mediator is based on the specification of a global schema describing a domain of interest, and on a set of mapping rules expressing how the content of each source available is related to the domain of interest. Mediator is in charge of (partially) the semantic query processing: it rewrites the user's query in the most specific terms of the domain in order to map any term of the query with a term used in a services provider. It is based on properties obtained with the use of rules and description logics. The mediator is also in charge of the domain knowledge management.

Input of the mediator constitutes:

- A global schema describing the domain of interest,
- A set of mapping rules expressing how the content of each source available is related to the domain of interest.

Output of the mediator constitutes:

- The queries of customers reformulated as semantic queries in terms of the tourism global schema.

Planner is in charge of a composition of web services. Service providers express their contents through the terminology of the domain used by the mediator. The planner uses this terminology and the domain ontology in order to define a sequence of queries to ask providers. Schematically, from a user request, the planner first identifies (using a C/S Ps catalogue) a list of pertinent C/S Ps and second splits the original request in a sequence of queries to be executed.

Input of the planner constitutes:

- A catalogue with all entered C/S Ps,
- The original user request.

Output of the planner constitutes:

- A sequence of queries to be executed.

2.4 Use Case 4. E-Photo album automation services on a portal *by FT.*

2.4.1 Use case summary

The diagram of Figure 2.4.1 summarizes the business context of E-photo album automation services on a portal use case. Its key functionalities are as follows:

- Automatic creation of photo albums;
- Dynamic discovery of contents and services providers (C/S Ps);
- Semi-automatic aggregation/association of the albums with contents exterior to the portal.

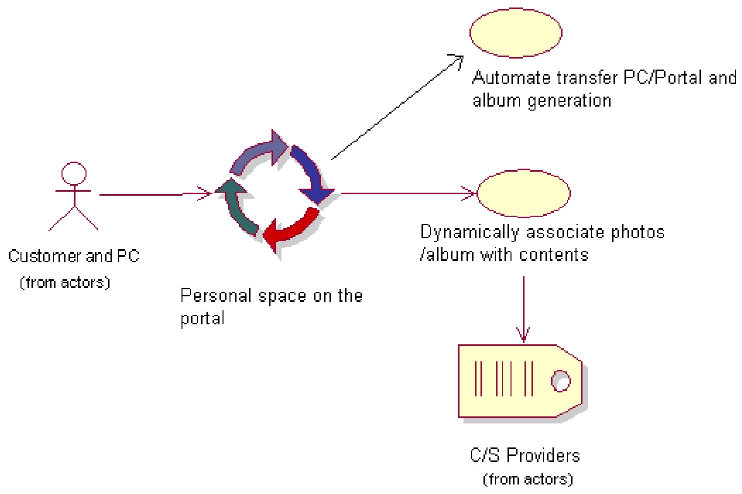

Fig. 2.4.1 UML use case diagram for E-photo album automation services on a portal

Now, based on the functional requirements, it is possible to specify also different technical use cases taking part in the platform. Those use cases will next allow for a detailed analysis of the technical needs.

2.4.2 Service requirements

Technical use cases diagram is presented in Figure 2.4.2. It allows identifying technical use cases and associated actors. Let us discuss actors of Figure 2.4.2 in detail.

Customer and PC. The customer is registered on the portal and arranges its personal contents. The customer uses its PC to access the portal.

Personal space on the portal. The portal gives access to the services available within the system through authentication mechanism, personalisation, and session management.

Contents and Services providers. The contents and services providers manage their contents autonomously, i.e. the system does not impose any constraints. Each C/S P has its own rules for structuring information at the protocol, syntactical, and semantic levels.

Administrator is in charge of the administration of the platform. It performs (i) referencing of new contents and services providers, (ii) internal knowledge representation and management, and (iii) orchestration between different contents and services providers.

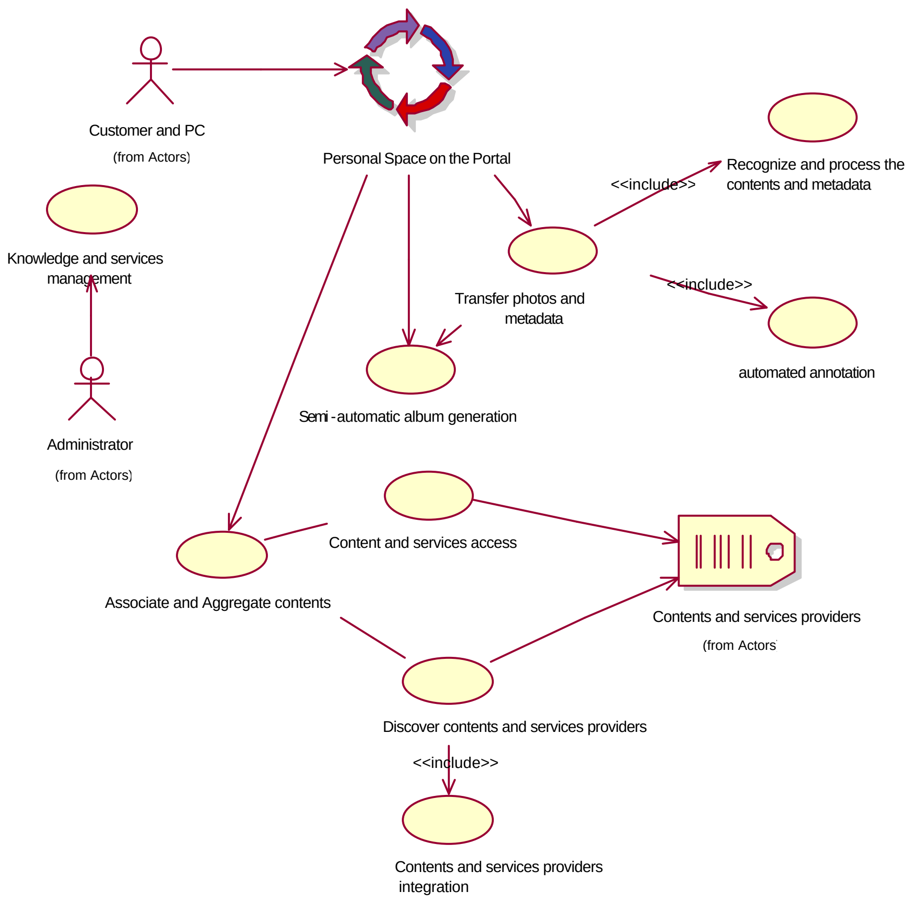

Fig.2.4.2 UML technical use case diagram for E-photo albums

2.4.3 Analysis

Now we analyze each technical use case of Figure 2.4.2 in detail. In particular, we consider *transfer photos and metadata*, *recognize and process the contents and metadata*, *automated annotation*, *semi-automatic album generation*, *discover the C/S P* technical use cases. Notice, that in Figure 2.4.2 the following technical use cases: *associate and aggregate contents*, *contents and services access*, *knowledge and services management*, *contents and services providers' integration* appear without any modifications with respect to the use case of Figure 2.3.2 (B2C marketplace for tourism use case), and hence we do not discuss them here.

For each technical use case we first report the actors it involves, then we provide its summary, inputs and outputs, and finally we discuss with the help of sequence diagrams the flow of its events and possible technology locks.

Transfer photos and metadata

Actors: Customer and PC, Personal space on the portal.

Summary: The photos and metadata are transferred from the customer PC's to the customer's personal space on the portal.

Preconditions and inputs:

- The customer is registered on the portal;
- The customer arranges its personal storage space.

Post-conditions and outputs:

- The contents are stored on the portal;
- The use cases *recognize and process the contents*, and *automated annotation* are activated.

The flow of events for the *transfer photos and metadata* technical use case is presented in Figure 2.4.3.

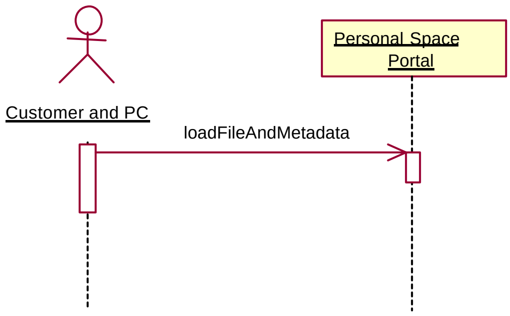

Fig.2.4.3 Flow of events: Transfer photos and metadata technical use case

Let us describe events of Figure 2.4.3 in more detail and in turn as they appear:

- The customer connects to its personal space on the portal;
- The customer loads its content with metadata on the portal.

Technology locks identification: There are no technology locks in this use case, however we have presented it here for clarity of the further discussions.

Recognize and process the contents and metadata

Actors: Personal space on the portal.

Summary: The contents and associated metadata are recognized and if necessary are converted as follows:

- Recognition of file format (e.g., jpeg, tiff);
- Identification of the metadata (date/time, location) attached to files;
- Conversion of voice to text, if there are such files attached to the photos;

- Low level features analysis (e.g., person identification and face recognition) of photos provides prompts to the user for associating, for example, people on the photos with appropriate entries in its address book.

Preconditions and inputs:

- The photos are transferred to the customer's personal space on the portal;
- The formats of files are standard (e.g., jpeg, tiff);
- There are points of reference for interpreting date/time, location parameters within the portal.

Post-conditions and outputs:

- Use case *automated annotation* is activated.

The flow of events for the *recognize and process the contents and metadata* technical use case is presented in Figure 2.4.4.

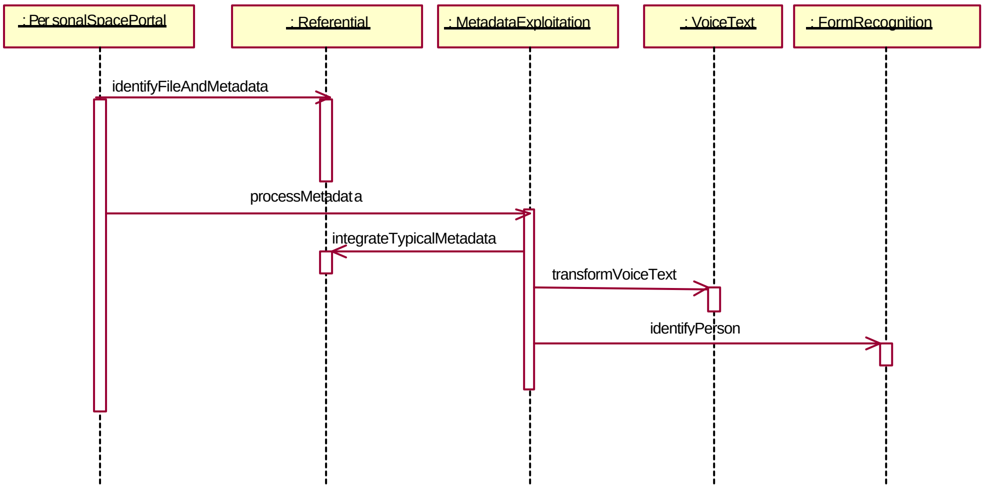

Fig.2.4.4 Flow of events: Recognize and process the contents and metadata technical use case

Let us describe events of Figure 2.4.4 in more detail and in turn as they appear:

- Identify the metadata associated to the contents (*identifyFileAndMetadata*);
- Process the metadata (*processMetadata*):
  - Typical metadata parameters (e.g., date-time, geolocalization, file format) are associated with the existing points of reference on the portal (*integrateTypicalMetadata*);
  - The voice/text conversion (*transformVoiceText*) and form recognition (*identifyPerson*) require particular processes to be exploited.

Technology locks identification: There are no technology locks in this use case, however we have presented it here for clarity of the further discussions.

Automated annotation

Actors: Personal space on the portal.

Summary: The textual contents extracted from the speech/text conversion are indexed in order to enrich metadata with some meaning.

Preconditions and inputs:

- The photos are transferred to the customer's personal space on the portal;
- The format of the files are standard;
- There are points of reference for interpreting metadata within the portal.

Post-conditions and outputs:

- Use case *semi-automatic generation of the albums* is activated. The portal suggests possible topics for photo albums depending on the generated metadata.

The flow of events for the *automated annotation* technical use case is presented in Figure 2.4.5.

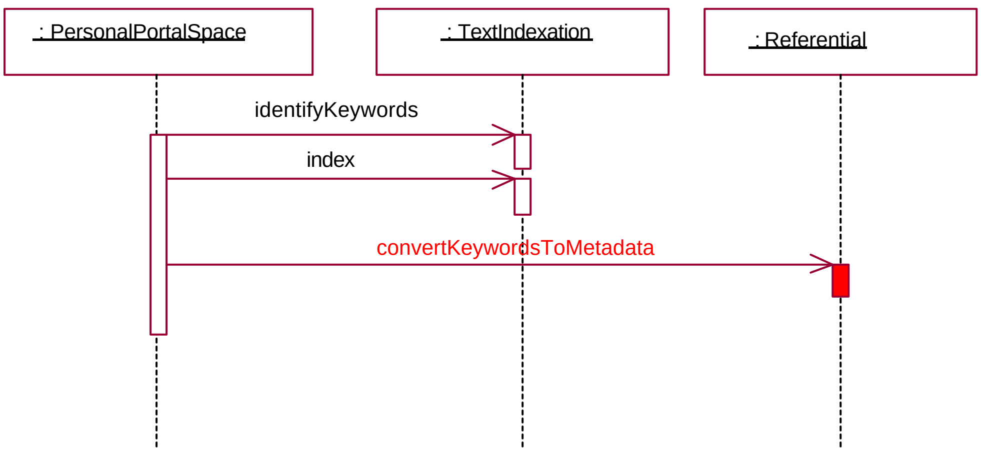

Fig.2.4.5 Flow of events: A utomation annotation technical use case

Let us describe events of Figure 2.4.5 in more detail and in turn as they appear:

- Identify the keywords (*identifyKeywords*) extracted of the text files associated to photos;
- Index the keywords attached to the photos (*index*);
- Identify the context in which keywords are used for descriptions of photos. Extend the context of the photo by adding new metadata (*convertKeywordsToMetadata*).

Technology locks identification: Technology locks are marked in red in Figure 2.4.5. The main idea of this technical use case is based on the capability of extending the context of metadata attached to photos from the text produced as a result of the speech/text conversion. However, indexing the keywords does not provide the structure of information. The *context should be determined* in order to choose the correct senses (meanings) of keywords such that they become a part of metadata. For example, if the text associated to a photo contains the sentence: "this is a photo of the Eiffel tower taken at an evening of December 2004, when it is snowing" the system has to convert only Eiffel tower into metadata attached to the photo.

Semi-automatic album generation

Actors: Personal space on the portal, Customer and PC.

Summary: The portal offers to the customer possible solutions for organising its photo albums with the help of an interactive interface.

Preconditions and inputs:

- The metadata associated with each photo is updated.

Post-conditions and outputs:

- The photo albums are organized.

The flow of events for the *semi-automatic album generation* technical use case is presented in Figure 2.4.6.

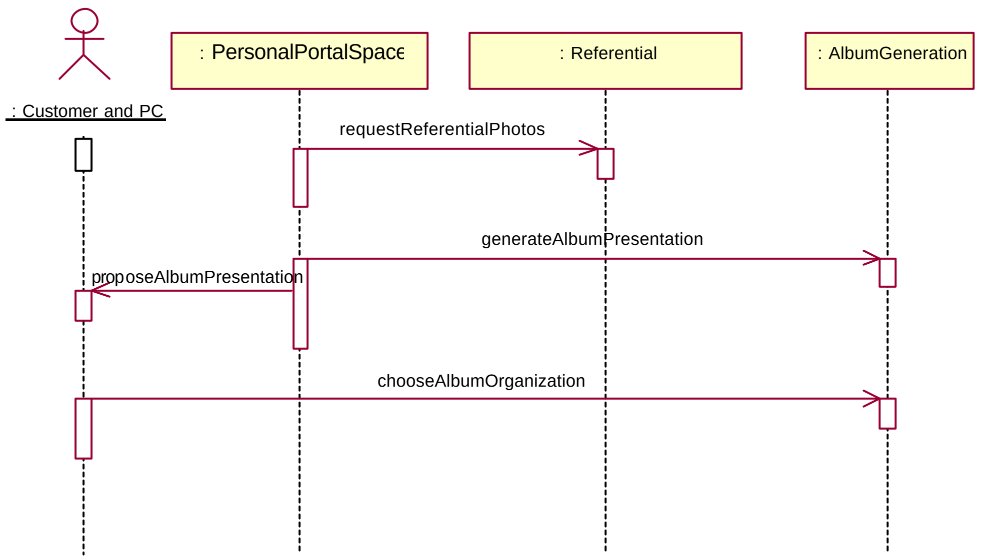

Fig.2.4.6 Flow of events: Semi-automatic album generation technical use case

Let us describe events of Figure 2.4.6 in more detail and in turn as they appear:

- Load the metadata associated to the photos (*requestReferentialPhotos*);
- Generate the thematic or hierarchical organisation of the albums from metadata (*generateAlbumPresentation*) and propose it to the customer;
- The customer chooses the elements that (s)he wants to use (*chooseAlbumOrganization*) to organize the albums (for example, a category of photos in their albums by location and/or date).

Technology locks identification: There are no technology locks in this use case, however we have presented it here for clarity of the use case essence.

Discover the C/S Ps

Actors: C/S Ps.

Summary: Portal searches on-the-fly for the relevant C/S Ps within directories (e.g., UDDI) in order to propose to the costumer additional information related to the context of its photos.

Preconditions and inputs:

- The context associated with photos is given.

Post-conditions and outputs:

- The C/S Ps are identified.

The flow of events for the *discover C/S P* technical use case is presented in Figure 2.4.6.

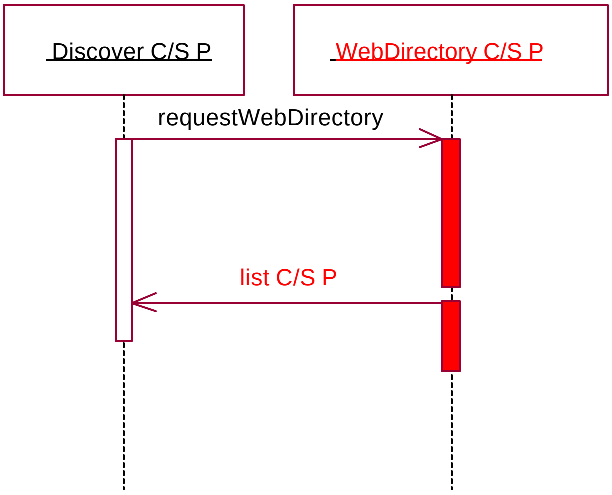

Fig.2.4.7 Flow of events: Discover C/S P technical use case

Let us describe events of Figure 2.4.7 in more detail. In particular, C/S P is determined by analyzing its description files (e.g., WSDL) in a directory (e.g., UDDI).

Technology locks identification: Technology locks are marked in red in Figure 2.4.7. They are the *directory of C/S Ps* and the *search through the directory of C/S Ps* in order to determine a list of only relevant ones with respect to the query. The keyword-based search on the Web (e.g., Google) does not allow exact identification of relevant C/S Ps. The main idea of this technical use case is based on the capability of identifying several pertinent C/S Ps by using the set of metadata attached to photos. The current solutions, for example UDDI directory, do not allow semantic interpretation of contents distributed by C/S Ps.

2.4.4 Design

2.4.4.1 The use case knowledge processing tasks

Having identified major technology locks of the E-Photo album system, now we are able to state knowledge processing tasks required in order to develop plausible Semantic Web solutions to those technology locks. These tasks are listed below and are described with the help of examples. Also for each task we refer to the system use where it has to be executed.

Notice we omit in our further discussions the knowledge processing tasks identical to the previous use case, i.e., B2C marketplace for tourism. These tasks are *mapping rules definition, data translation, semantic query processing, and results reconciliation.* Thus, knowledge processing tasks provided below are only extensions of the tasks of the previous use case.

Metadata generation. This is a task of (i) detecting the relevant keywords within the speech/text attached to photos and (ii) interpreting them in function of the context. This task appears in the *automated annotation* technical use case.

Searching for content providers. This is a task of searching for additional information related to the context of photos. This search relies on metadata attached to photos, and its meaningful interpretation in order to identify relevant C/S Ps. All C/S Ps are entered in a directory (e.g., UDDI). This directory can be requested using a set of metadata attached to photos. The result of such a query is a list of relevant C/S Ps that the system must thereafter integrate by respecting the knowledge processing tasks described in the "B2C marketplace for tourism" use case (e.g., semantic query processing, results reconciliation). Essentially, this task represents a combination of the above mentioned tasks of the "B2C marketplace for tourism" use case, although in the context of E-photo scenario; therefore we mention it here for the clarity of the use case under consideration. This task appears in the *discover the C/S Ps* technical use case.

Content provider's directory management. This is a task of managing a set of content providers which respect standard protocols of content description and invocation. The directory must allow for a discovery of content providers, and must authorize addition and retrieval of content providers. The discovery of content providers must be performed at the semantic level. This task appears in the *discover the C/S Ps* technical use case.

2.4.4.2 The use case typology of knowledge processing tasks

We build a typology of knowledge processing tasks by splitting all the knowledge processing tasks in to primarily and secondary tasks according to the business logic of the system. Let us consider them in turn. Notice that here we discuss knowledge processing tasks as extensions of the tasks of the "B2C marketplace for tourism" use case.

Primary Tasks

- Metadata generation. This task is crucial because: the richer metadata is, the more possibilities of album presentation is available.

Secondary Tasks:

- Content provider's directory management. This task is secondary as the directory of C/S Ps is an external information resource with respect to the portal.

2.4.4.3 The use case library of high level components

Table 2.4 briefly summarizes primarily knowledge processing tasks and their corresponding high level components. Notice that here we discuss knowledge processing tasks and corresponding components as extensions of the tasks and components of the "B2C marketplace for tourism" use case. Also *searching for content providers* task mentioned above requires components which were already mentioned in the "B2C marketplace for tourism" use case (e.g., semantic query processing, results reconciliation). Therefore, we are not reporting it here.

Let us discuss in detail each of the components in terms of their inputs and outputs, leaving the algorithms they have inside as a black box, because these issues are irrelevant for the goals of the report.

Table 2.4 Use case 4. Knowledge processing tasks & components.

Knowledge processing tasks		Components
Primary tasks	Metadata Generation	**Metadata Generator**
Secondary Tasks	Content Provider's Directory Management	**Directory Manager**

Metadata generator. This is a module which is in charge of text analysis in order to detect keywords and produce relevant metadata. The input of this component is free text; the output is a set of metadata.

Directory manager. This is a service allowing for a semantic B2B C/S P integration. It defines a standard interface for accessing a "database" of C/S Ps. All classical operations on the database are allowed. We can add, retrieve, modify, and search for C/S Ps.

3. Typology of knowledge processing tasks and a library of high level components

General typology of knowledge processing tasks is structured in terms of primarily and secondary tasks. It is composed of knowledge processing tasks which have been identified for each use case discussed in the deliverable. In particular:

- In the "Recruitment" use case we have identified the following primarily tasks: *data translation*, *ontology management*, *matching*, *ranking matching results*; and the following secondary tasks: *schema/ontology merging*, *and producing explanations*.
- In the "Multimedia content analysis and annotation" use case we have identified the following primarily tasks: *content annotation*, *ontology management*, *reasoning with annotations*, *intelligent search and retrieval*; and *personalization and media adaptation* as a secondary task.
- In the "B2C market place for tourism" use case we have identified the following primarily tasks: *mapping rules definition*, *data translation*, *semantic query processing*, *composition of web services*, *results reconciliation*; and *global schema management* as a secondary task.
- In the "E-photo album automation services on a portal" use case we have identified knowledge processing tasks as extensions of the "B2C market place for tourism" use case. In particular, the primary task is *metadata generation*. The secondary task is *directory management*.

A general typology of knowledge processing tasks identified in the use cases under consideration is briefly summarized in Table 3.1.

If the same knowledge processing task occurs in more than one use case (for example, ontology management task appears in all the use cases) we report it only once. Also in Table 3.1 we give more general names to some tasks and components, rather than they appear in the analysis and design parts of Section 2. Therefore, at this stage we hide specificity of each use case, and emphasize only high level (typical) requirements of the information systems to be developed. For example, personalization and media adaptation task from the second use case appears in Table 3.1 as the personalization task.

General typology of knowledge processing tasks includes 9 primary tasks and 4 secondary tasks. It is also worth noticing that some tasks are to be implemented within a single component. For example, the following tasks: schema/ontology matching, ranking matching results, and producing explanations of mappings are the functionalities of a match manager component. Thus, the library of high level components contains less components than the number of knowledge processing tasks identified. In particular, it consists of 10 components.

Table 3.1 General typology of knowledge processing tasks & components

Knowledge processing tasks		Components
Primarily tasks	Data Translation	**Wrapper**
	Ontology Management	**Ontology Manager**
	Matching	**Match Manager**
	Matching Results Analysis	**Match Manager**
	Content Annotation	**Annotation Manager**
	Reasoning	**Reasoner**
	Semantic Query Processing	**Query Processor**
	Composition of Web Services	**Planner**
	Results Reconciliation	**Results Reconciler**

Secondary tasks	Schema/Ontology Merging	**Ontology Manager**
	Producing Explanations	**Match Manager**
	Personalization	**Profiler**
	Directory Management	**Directory Manager**

Below we provide short high level descriptions of knowledge processing tasks and corresponding components of Table 3.1, while their detailed descriptions can be found in Section 2.

Data Translation & Wrapper. This task and component are in charge of translating/exchanging instances between heterogeneous information sources storing their data in different formats (e.g., RDF, SQL DDL). These were required by the three use cases under consideration.

Ontology Management, Schema/Ontology Matching, Merging & Ontology Manager. These tasks and component are in charge of ontology maintenance with respect to (evolving) business case requirements. These were required by all the use cases under consideration.

Matching, Matching Results Analysis, Producing Explanations & Match Manager. These tasks and component are in charge of determining mappings between the entities of multiple schemas/ontologies. The mappings might be ordered according to some criteria. In addition, explanations of the mappings might be also produced. The first functionality was required by the three use cases under consideration. The latter two functionalities were required by one use cases under consideration.

Content Annotation & Annotation Manager. This task and component are in charge of automatic production of content metadata. These were required by the two use cases under consideration.

Reasoning & Reasoner. This task and component are in charge of logical reasoning. This task and component were required by one use case under consideration.

Semantic Query Processing & Query Processor. This task and component are in charge of interpreting (rewriting) a query by using terms which are explicitly specified in a model of the domain. These were required by the three use cases under consideration.

Composition of Web Services & Planner. This task and component are in charge of automated composition of web services into executable processes. These were required by one use case under consideration.

Results Reconciliation & Results Reconciler. This task and component are in charge of determining an optimal solution, in terms of contents (no information duplication, etc.), for returning results from the queried information sources. These were required by one use case under consideration.

Personalization & Profiler. This task and component are in charge of tailoring services available form the system to the specificity of each user (e.g., standard vs. professional profiles). These were required by one use case under consideration.

Directory Management & Directory Manager. This task and component are in charge of maintenance and semantic interpretation of instance data distributed by content and service providers. These were required by one use case under consideration.

4. Conclusions

In this deliverable we have demonstrated a methodology for identifying knowledge processing tasks and corresponding high level components within the information systems by analyzing in detail the four use cases of D1.1.2. We have developed a typology of knowledge processing tasks with respect to *each* use case and a general typology covering requirements of *all* the use cases together. Also, whenever possible, we have indicated state of the art solutions and relevant activities being held in the Knowledge Web research workpackages, thus, showing applicability of the knowledge-based technology.

The fact that at present we have considered only some use cases provided in D1.1.2 was taken into account. In particular, a quick analysis of the other use cases has shown that knowledge processing tasks are repeating, therefore we can conclude that material of this deliverable presents the core knowledge processing tasks, i.e., tasks occurring in most of the systems.

However, with emergence of new business cases it is likely that new knowledge processing tasks will appear. For example, web service's discovery, orchestration, and so on. Therefore, future work includes technical analysis of the new use cases of D1.1.2 till the saturation is reached. Notice, that not all the use cases of D1.1.2 will be analyzed in D1.1.3. In particular, we will concentrate only on the most important industry areas and their business cases.

6. References

1. Allen J. F.: Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(1): 832-843, 1983.
2. Booch, G., Rumbaugh J., Jacobson I.: The unified modeling language user guide. Addison-Wesley, 1997.
3. Bouquet P., Serafini L., Zanobini S.: Semantic coordination: A new approach and an application. *Proceedings of ISWC*, 130-145, 2003.

4. Chang S. F., Chen W., Meng H. J., Sundaram H., Zhong D.: VideoQ: An automated content based video search system using visual queries. *ACM Multimedia*, 313-324, 1997.
5. Cruz I. F., Lucas W. T.: A visual approach to multimedia querying and presentation. *ACM Multimedia*, 109-120, 1997.
6. d'Aquin M., Bouthier C., Brachais S., Lieber J., Napoli A.: Knowledge editing and maintenance tools for a semantic portal in oncology. Rapport de recherche A03-R-162, LORIA, 2003.
7. Davies J., Fensel D., van Harmelen F. (eds.): The Semantic Web: ontology-driven knowledge management. John Wiley, 2002.
8. Day Y. F., Dagtas S., Lino M., Khokhar A., Ghafoor A.: Spatio-temporal modeling of video data for on-line object-oriented query processing. *Proceedings of ICMCS*, 98-105, 1995.
9. Do H.H., Rahm E.: COMA – A system for flexible combination of schema matching approaches. *Proceedings of VLDB*, 610-621, 2002.
10. Ehrig M., Staab S.: QOM: Quick ontology mapping. *Proceedings of ISWC*, 2004.
11. Euzenat J., Valtchev P.: An integrative proximity measure for ontology alignment. *Proceedings of SI workshop at ISWC*, 2003.
12. Euzenat J., Valtchev P.: Similarity-based ontology alignment in OWL-Lite. *Proceedings of ECAI*, 2004.
13. Flickner M., Sawhney H., Niblack H., Ashley J., Huang Q., Dom B., Gorkani M., Hafner J., Lee D., Petkovic D., Steele D., Yanker P.: Query by image and video content: the QBIC system. *Computer*, 28(9): 23-32, 1995.
14. Giunchiglia F., Shvaiko P., Yatskevich M.: S-Match: an algorithm and an implementation of semantic matching. *Proceedings of ESWS*, 61-75, 2004.
15. Giunchiglia F., Shvaiko P.: Semantic matching. *The Knowledge Engineering Review journal* 18(3):265-280, 2003.
16. Hartmann J., Sure Y.: A knowledge discovery workbench for the Semantic Web. *Proceedings of workshop on Mining for and from the Semantic Web at the ACM SIGKDD*. 2004.
17. Heinecke J., Toumani F.: A natural language mediation system for e-commerce applications. An ontology-based approach. *Proceedings of workshop on Human Language Technology for the Semantic Web and Web Services at ISWC*, 2003.
18. Jacobson I., Booch G., Rumbaugh J.: The unified software development process. Addisson-Wesley, 1999.
19. Kalfoglou Y., Schorlemmer M.: Ontology mapping: the state of the art. *The Knowledge Engineering Review journal*, 18(1):1-31, 2003.
20. Kamp H., Reyle U.: From Discourse to Logic. Introduction to model theoretic semantics of natural language, formal logic and discourse representation theory. Dordrecht: Kluwer Academic Publishers. 1993.
21. Keh H-C., Shih T. K.: Shape-based image retrieval using spatio-temporal relation computations. *Tamkang journal of Science and Engineering*, 4: 9-18, 2001.
22. Lehtola A., Heinecke J., Bounsaythip C.: Intelligent human language query processing in MKBEEM. *Proceedings of HCI International* 2003: Vol 4. Mahwah, NJ: Lawrence Erlbaum Associates, pp. 750-754.
23. Lenzerini M.: Data integration: a theoretical perspective. *Tutorial at PODS*, 2002.

24. Madhavan J., Bernstein P., Rahm E.: Generic schema matching with Cupid. *Proceedings of VLDB*, 49-58, 2001.
25. Maedche A., Staab S., Stojanovic N., Studer R., Sure Y.: SEAL -- A framework for developing Semantic Web portals. *Lecture Notes in Computer Science*, 2097:1-22, 2001.
26. McGuinness D. L., Fikes R., Rice J., Wilder S.: An environment for merging and testing large ontologies. *Proceedings of KR*, 483-493, 2000.
27. McGuinness D. L., Pinheiro da Silva P.: Infrastructure for web explanations. *Proceedings of ISWC*, 113-129, 2003.
28. McGuinness D. L., Shvaiko P., Giunchiglia F., Pinheiro da Silva P.: Towards explaining semantic matching. *Proceedings of international workshop on DL*, 2004.
29. Melnik S., Rahm E., Bernstein P.: Rondo: A programming platform for generic model management. *Proceedings of SIGMOD*, 193-204, 2003.
30. Noy N., Musen M. A.: Anchor-PROMPT: Using non-local context for semantic matching. *Proceedings of IJCAI workshop on OIS*, 63-70, 2001.
31. Oberle D., Volz R., Motik B., Staab S.: An extensible ontology software environment. Handbook on Ontologies, chapter III, pp. 311-333. Springer, 2004.
32. Petrini J., Risch T.: Processing queries over RDF views of wrapped relational databases. *Proceedings of the 1st International workshop on Wrapper Techniques for Legacy Systems*, Delft, Holland, 2004.
33. Popov B., Kiryakov A., Manov D., Kirilov A., Ognyanoff D., Goranov M.: Towards Semantic Web information extraction. *Proceedings of workshop on Human Language Technology for the Semantic Web and Web Services at ISWC*, 2003.
34. Rahm E., Bernstein P.: A survey of approaches to automatic schema matching. *VLDB journal*, 10(4): 334-350, 2001.
35. Rational Software Corporation. http://www-306.ibm.com/software/rational/
36. Resnik P.: Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of IJCAI*, Montreal, Canada, 1995.
37. Shvaiko P. Euzenat J.: A survey of schema-based matching approaches. Technical Report, DIT-04-087. University of Trento, 2004.
38. Stojanovic N., Hartmann J., Gonzalez J.: OntoManager - a system for usage-based ontology management. *Proceedings of FGML workshop*. Special Interest Group of German Information Society, 2003.

Appendix 1. Dependencies with other deliverables

A number of Knowledge Web deliverables are related to this one:

Project	Deliverable	Title	Relationship
KW	D1.1.1	Industry board members list, clustering and organizational and operational charter (MoU)	Suggestions from D1.1.1 on the key industry sectors and the most important use cases to be analyzed in D1.1.3.
KW	D1.1.2	Prototypical business use cases	Use cases analyzed in this deliverable are taken as input from business cases of D1.1.2.
KW	D1.2.2	Semantic Web framework requirements analysis	High level components described in this report also serve as a partial input to D1.2.2.