REGNET

Cultural Heritage in REGional NETworks

Deliverable D2

The REGNET System: Specifications and State-of-the-art

Project acronym	REGNET	**Contract nr.**	IST-2000-26336
Type and Number	D2 – The REGNET System: Specifications and State-of-the-art		
Work package	WP1 - Analysis of the State-of-the-art and Development of Concepts		
Task	T 1.3 Identification of standards to be used T 1.4 Development of the System Specification		
Date of delivery	2001-09-30		
Code name	RN_D2v01.doc	**Version** 01 draft □ final ☑	
Objective	Report		
Distribution Type	Restricted		
Authors (Partner)	AIT, ONB, SR, IMAC, TARX, MOT, SPAC, IAT, ZEUS, SI CERT, VALT		
Abstract	Contains the results of Work Area B (Platform Engineering) related to Work Package 1. It refers to the design work to be done for the REGNET demonstrator and a study work related to standards, methods, and tools necessary for the REGNET implementation. The deliverable includes the basic design principles used for the design process and the baseline design as well as the detailed design for all of its components as outlined in the attachment to the REGNET contract.		
Keywords List	Standards, metadata, XML, REGNET functional architecture, themes, functional requirements, e-Business, wireless communication		

ist information society technologies

Table of Contents

Executive Summary

This document includes those activities of Work Package 1 (Analysis of the State-of-the-art and Development of Concepts) which have been carried out within Work Area B (Platform Engineering) of the REGNET project. There are two parts, one dealing with "Identification of standards to be used", and the other related to "Development of the System Specifications". The latter one includes an annex which refers especially to the software engineering process (state-of-the-art development environments, programming languages, architectures, risks, web services, etc) and evolving industry standards in the field of e-Business, wireless technology, ontology, and semantic WEB.

The domain related standards in the first part cover a wide range of specifications which evolved mainly during the last years and can be considered as stabilised. Because there was a dramatic evolution of standards since one of the key activities in the area of Cultural Heritage has been published (CIMI – Standards Framework, 1993) an introductory section deals with the use of standards, metadata systems, information interchange and e-Business. The Document Standards are grouped into application domains (libraries, archives, museums, artists) and especially those standards have been dealt with where commonly approved DTDs (document type definitions) are already available. Dublin Core (DC) Metadata have been selected as access points for distributed searches over different application domains since there exist already road maps supporting the mapping of DC into domain specific metadata and several projects like COVAX have already demonstrated the usefulness of this approach. On the technical side the Z39.50 protocol has been analysed and a further – experimental – development of this concept (XER: use of XML-wrappers instead of ASN.1 based Encoding Rules) has been chosen as basis for the specification of query formulations. XML and related standards have been selected as basis for all information engineering and information management procedures as they are commonly accepted and widely adopted in the industry. Publishing standards are also included in this part of the document. Since the evolution of the "Semantic Web" has generated a lot of activities several projects in the field of knowledge presentation have been looked into. The XML Topic Map (XTM) has been chosen as prime candidate for the unified implementation of "themes", thesauri, authority files, and similar applications. The practical use of the XTM-Standards has been demonstrated by testing different products (XTM-Servers/Clients) available on the Market.

The second part of this document is dedicated to the development of system specifications. The originally elaborated REGNET functional architecture has been slightly modified and includes now a new component (REGNET-Connector) in addition to the others (REGNET-Portal, REGNET-Cultural Heritage Data Management, REGNET-e-Business Data Management, REGNET-Ontology System, REGNET-Electronic Publisher). This was necessary for the implementation of a REGNET Network consisting of several REGNET-sites and supports a scalable technical infrastructure of a REGNET-Site according to the needs of a Cultural Service Centre (CSC). A specialised REGNET-Site (Ontology Master Site) will assure that the needed domain knowledge is available within the REGNET Network. The Ontology Node of each REGNET-Site is a very important component since it triggers most interactions with the systems, e.g.: data entry (meta data control), e-Business and searching. An adaptable search interface ("Cultural Online Browser") will be provided in case a thematic search is requested (provided XTM-formatted themes' information has been prepared and is available). Themes' related information included in the REGNET knowledge base will also be used for storyboard supported electronic publication processes. A further use of "thematic information" will be the creation of "personalised tours", which will be investigated in the 2nd REGNET Work Package. External Systems (Data Bases, Collection Management Systems, e-Shop Solutions, etc) will be connected via the REGNET-Connector to the REGNET Network. This enables the inclusion of proprietary systems of commercial vendors (e.g. ADLIB, INDEX+, etc) into the REGNET Network. In different chapters of the document the REGNET-Building Blocks, the Requirements, Use Cases (related to the functional requirements), the Functional Architecture (UML-Activity Diagrams), the Technical Architecture, the System Architecture and Tools, Deployment (several application possibilities of a REGNET-Site), and Interfaces are described. To enable the interconnection of REGNET Nodes which are developed on different platforms (JAVA, PHP) the SOAP-protocol (Simple Object Access Protocol) has been chosen. For the specifications UML (Unified Modelling Language) has been used as far as possible. The UML also is a important method of the iterative development process chosen for the System development as it is derived from the Rational Unified Process and the ISO-12207 V-Model approach. Tools (Web-Servers, XML-Data Base Systems, Relational Data

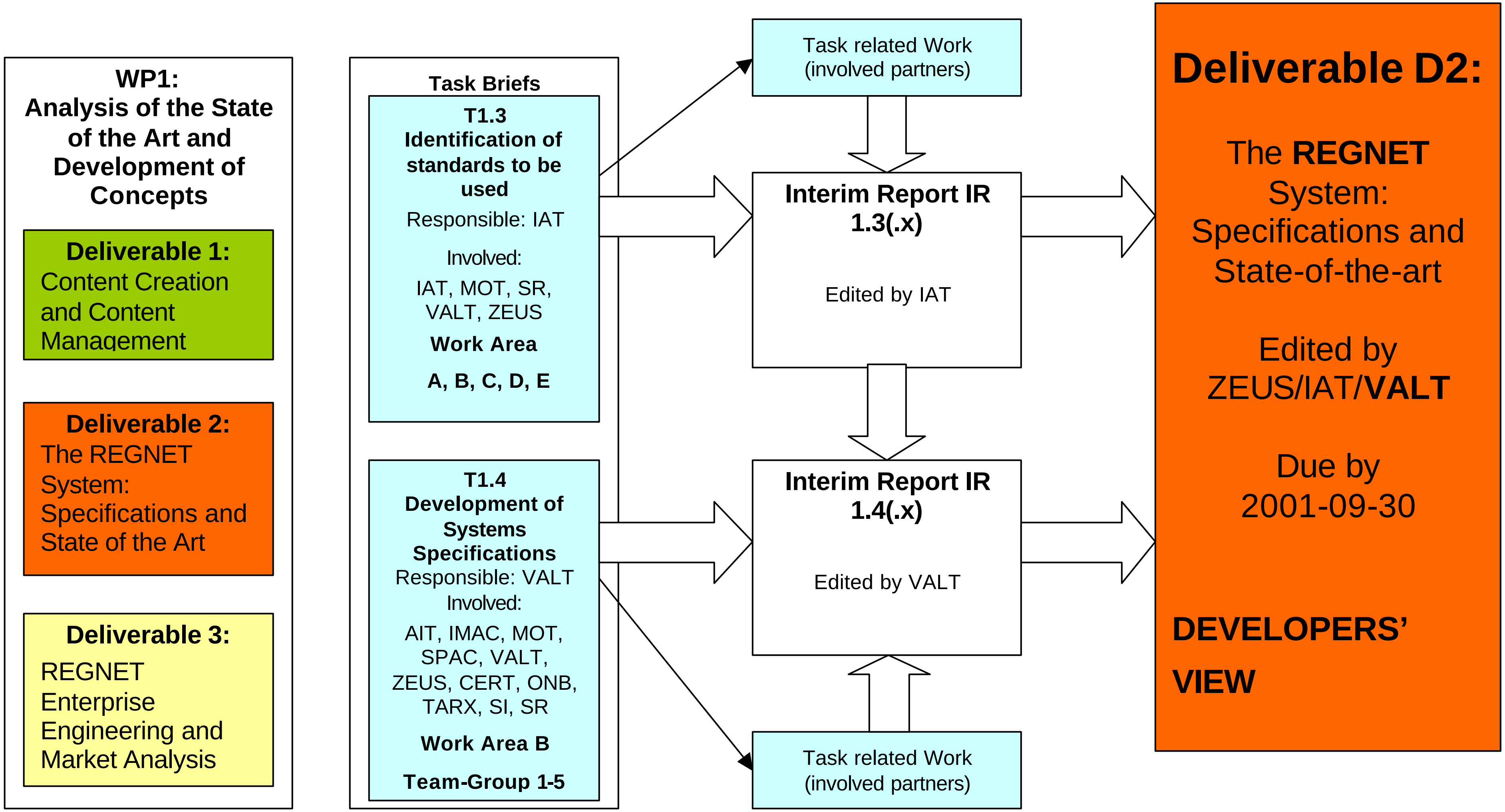
Base Systems, Parsers, Editors, etc) available in the Public Domain have mainly been chosen for the implementation process which will be carried out in Work Package 2 of the REGNET System. This should enable to run a Cultural Service Centre at low investment costs. Due to the importance of future access to Cultural Heritage, Wireless Access and Voice Access are described in dedicated chapters.

Situation

In D2 the focus is upon standards (Task 1.3: Identification of standards to be used) and technical development of concepts (Task 1.4 Development of System Specifications) to build up the REGNET System. The activities in Task 1.4 are further divided into various sub-tasks which are not shown here in detail. The next Work Package (WP 2) includes the natural follow-up to the activities in the current Work Package (WP 1) – the Implementation of the System. According to this, the next table has the aim to summarize in short, the tasks, partners, documentation and resources (person / month) for each of the involved tasks.

	Analysis of the State of the Art (WP1)				Implementation of the System and Preparation of the Demonstration (WP2)			
	Task	Leader	Document	MM	Task	Leader	Document	MM
Identification of standards to be used	1.3	IAT	IR1.3 → **D2**	7				
Development of System Specifications	1.4	VALT	IR1.4 → **D2**	65				
System Implementation (1. Version)					2.2	ZEUS	IR 2.2 → **D5**	70

PLATFORM ENGINEERING



WP1:
Analysis of the State of the Art and Development of Concepts

Deliverable 1: Content Creation and Content Management

Deliverable 2: The REGNET System: Specifications and State of the Art

Deliverable 3: REGNET Enterprise Engineering and Market Analysis

Task Briefs

T1.3 Identification of standards to be used
Responsible: IAT
Involved:
IAT, MOT, SR, VALT, ZEUS
Work Area
A, B, C, D, E

T1.4 Development of Systems Specifications
Responsible: VALT
Involved:
AIT, IMAC, MOT, SPAC, VALT, ZEUS, CERT, ONB, TARX, SI, SR
Work Area B
Team-Group 1-5

Task related Work (involved partners)

Interim Report IR 1.3(.x)
Edited by IAT

Interim Report IR 1.4(.x)
Edited by VALT

Task related Work (involved partners)

Deliverable D2:
The **REGNET** System: Specifications and State-of-the-art

Edited by ZEUS/IAT/**VALT**

Due by 2001-09-30

DEVELOPERS' VIEW

About this Document

This document is related to the *Platform Engineering* Activities of the REGNET-Project and contains the REGNET - UML specifications. According to the two-track process functional and technical specification are included.

Functional analysis is based on the functional requirements. These requirements are modelled using UML through Use Case representation. The REGNET functional architecture is then defined and shows links between modules according to the systems' functionalities.

The *technical architecture* is based on technical requirements. This architecture deals with the choice of technologies according to functionalities and technical requirements. It details the definition of hardware, network and software tools. An iterative software project plan, based on risk analysis is proposed.

The REGNET specification and solutions are split into two parts: *Requirement artefacts* which capture and present information used in defining the required capabilities of the system, and *Analysis & Design artefacts* which capture and present information related to the solution to the problems defined by the Requirements artefacts. *User requirements* (both functional and technical) reflect as much as possible the real needs so the functional requirements are based on solid ground.

The chapters are organised as follows:

- Description of the REGNET System: This section gives an comprehensive overview of the REGNET concept.
- Introduction: This section helps the reader to classify this deliverable within the overall project context.
- Methodology: This section describes the methodologies applied to produce the content of this deliverable.
- Part1 - Identification of standards to be used: In this section an overview of available standards in the content and technical area are given and their advantages and disadvantages in the context of the REGNET System are described.
- Part 2 - Development of the System Specification: In this section the Requirements, technical as well as functional, and the resulting architecture of the REGNET System are identified as well as tools for the system development.

Part 1 reviews the main metadata systems used in each of sub-domains in the Cultural Heritage field: library or bibliographic, museum and archive. In the library sub-domain, the main reference is the MARC (Machine Readable Catalogue or Cataloguing) standard. Here, we also make a review of some projects working in converting MARC records in XML specifications. In the archival sub-domain, the main references are ISAD(G), an archival description standard, and EAD, and archival communication standard, SGML-compliant. To description of sound and audio-visual archives, the main reference are the IASA Cataloguing Rules. In the museum sub-domain, harmonisation efforts are much more difficult due to the heterogeneity of the objects to be described. Some important projects in the field of arts are the AMICO project, the system CDWA and the CIDOC Information Categories.

The information search and retrieval section first presents the requirements of the cross-domain search. Next, we study the Dublin Core metadata standard, which is a metadata element set intended to facilitate discovery of electronic resources. Next, we present the main features of the Z39.50 protocol for the distributed search and retrieval. This standard specifies the structures and rules which allow a client machine to search a database on a server machine and retrieve records that are identified as a result of such search. The Z39.50 standard has been of major importance in supporting access to distributed library databases and catalogues.

The information interchange section is dedicated to study XML, RDF and EDI.

The e-Business section tries to define in detail all modern initiatives and approaches of defining the current standards and some new that are not yet completed. The basic research has been made in order to define and clarify the best solution for the eBusiness subsystem. We only describe the technologies based on the XML syntax: RosettaNet, ebXML, XML/EDI, BizTalk and the web services (UDDI, WSDL and SOAP). Conclusion of this study is that ebXML is the more accurate B2B approach according to REGNET context: it provides a business domain independent framework dedicated to smaller companies.

The publishing standards section provides relevant information about electronic publishing. Standards in the area of multimedia authoring, storyboards and multimedia document models are revised, mainly HyTime, SMIL and MPEG-7.

The specification of part 2 handles two different types of artefacts:

Requirement artefacts which captures and presents information used in defining the required capabilities of the system.

Analysis & Design artefact which captures and presents information related to the solution to the problems posed in the Requirements set.

The requirements artefacts are represented by functional and technical requirements forming the preconditions for the development of the functional and technical architecture. The requirements are further detailed using Use Cases.

After the architectural section possible set up alternatives for the system are given followed by a detailed description of the interactions of the different subsystems of the REGNET system.

1 Description of the REGNET System

The REGNET Project supports the introduction of new ways of co-operation between different cultural stakeholders ("CIO": Cultural Institutions and Organisations, Industries, Administrations, etc.) by setting up a technical infrastructure and organisational framework in the light of globalisation and world wide markets. The main activities within a support environment for (Internet) Markets comprise: Content Engineering, Platform Engineering, and Enterprise Engineering.

The REGNET - Cultural Heritage in REGional NETworks project - targets to all of the three areas. The project was introduced under the Action Line 'Access to digital collections of cultural and scientific content' of the European Union IST-Information Society Technologies Programme and 23 partners, cultural organisations (museums, libraries and archives) and IT-industry representatives, from 10 European Union states as well as Bulgaria and Russia are participating.

1.1 Envisaged Achievements

REGNET will set up a functional network of cultural service centres through Europe which will provide IT-services dedicated to Cultural Heritage organisations. A technical and legal framework, the REGNET system, for such a service infrastructure will be developed. This will offer services like data entry, search and retrieval, and e-Business. It will be based mainly on integration work using state-of-the-art components. The network will integrate multimedia industries, content providers and service centre operators. Existing cultural infrastructures will be exploited and new infrastructures should be developed where necessary.

The technical infrastructure should allow to set up even low cost service centres. The membership concept of the service centres will generate a critical mass of digital or physical goods contained in Content Provider's organisations.

1.2 The Players within the Network

The four players within the network are the content providers, the service centre operators (running a CSC: Cultural Service Centres), the system developers and end users. The content providers (museums, libraries, archives etc.) will provide access (via wired and wireless communication) to their digital contents, services and products and offer them to their clients (B2C). In return they can use the REGNET facilities for multimedia productions and data base management, or cooperage with other REGNET partners during the creation of data bases, generation of multimedia products or creation of a virtual exhibition (B2B). The service centre operators will generate income by providing the technical infrastructure (software/hardware) to content providers and other partners within the REGNET network. They offer additional IT-services and consultancies. The system developers are selling the REGNET system to other cultural service centres and content providers. They implement additional components for the REGNET software system (additional 'nodes' like an 'exhibition creator', etc), and will generate income via licence fees for the REGNET system and other IT-services. For the end user the system will offer easy and wide access to Cultural Heritage data information and the purchase of CH related goods and services at one point, with stress on the production of personalised goods (e.g. CDROM) and services.

1.3 Standards and Metadata

REGNET will provide a unique facility to meta data definitions for both CH-related data and e-Business related data via the search entry in the REGNET – Portal and the REGNET - Ontology subsystem. Here not only meta data related to objects is addressed, but also definitions of work flows, business processes, etc. are included which lead to the concept of Ontology within the Cultural Heritage domain. Within the project a data entry facility using WEB-browsers and adaptable to different needs within different domains (Museums, Library, Archives) will be established. This facility will support existing standards (ICOM/CIDOC, UNIMARC, ISAD(G), etc) and is configurable by the REGNET System user. At the e-Business level REGNET provides the generation of a customised shopping cart systems within the business to consumer (B2C) framework.

1.4 Mobile Entry

It will be examined how next generation mobile networks can be exploited to widen the potentiality of WEB services in the field of Cultural Heritage. Appropriate gateway functions and interfacing units will be envisaged to connect the REGNET architecture with the UMTS (Universal Mobile Telecommunication Systems) structure and ensure a seamless provisioning of REGNET services to mobile customers.

1.5 Electronic Catalogues

Already existing electronic catalogues (OPACS: Online Public Access Catalogues) referring to cultural & scientific objects contained in libraries, museums, archives, and galleries, as well as to goods and services will be used. Where there are no catalogues yet new catalogues will be developed in order to establish a 'virtual union' catalogue of all OPACS and product/service catalogues held locally.

1.6 Business Engineering

In addition the work includes the outlining of necessary 'supply chains' and the connected business processes and functions to deliver digital and physical goods. All the business transactions on the B2B and B2C level (containing payment features, copyright systems, authentication control, etc) will be guided by a legal framework that will be worked out accordingly. The reorganisation of existing processes and introduction of computerised functions which make transactions for low money goods worth while (e.g. small copyright fees for digital images) is an essential part of the project.

1.7 REGNET Architecture

The building blocks which constitute the REGNET System are supporting access to cultural & scientific information as well as to product & service information offered by different organisations. The building blocks are outlined in the figure below.

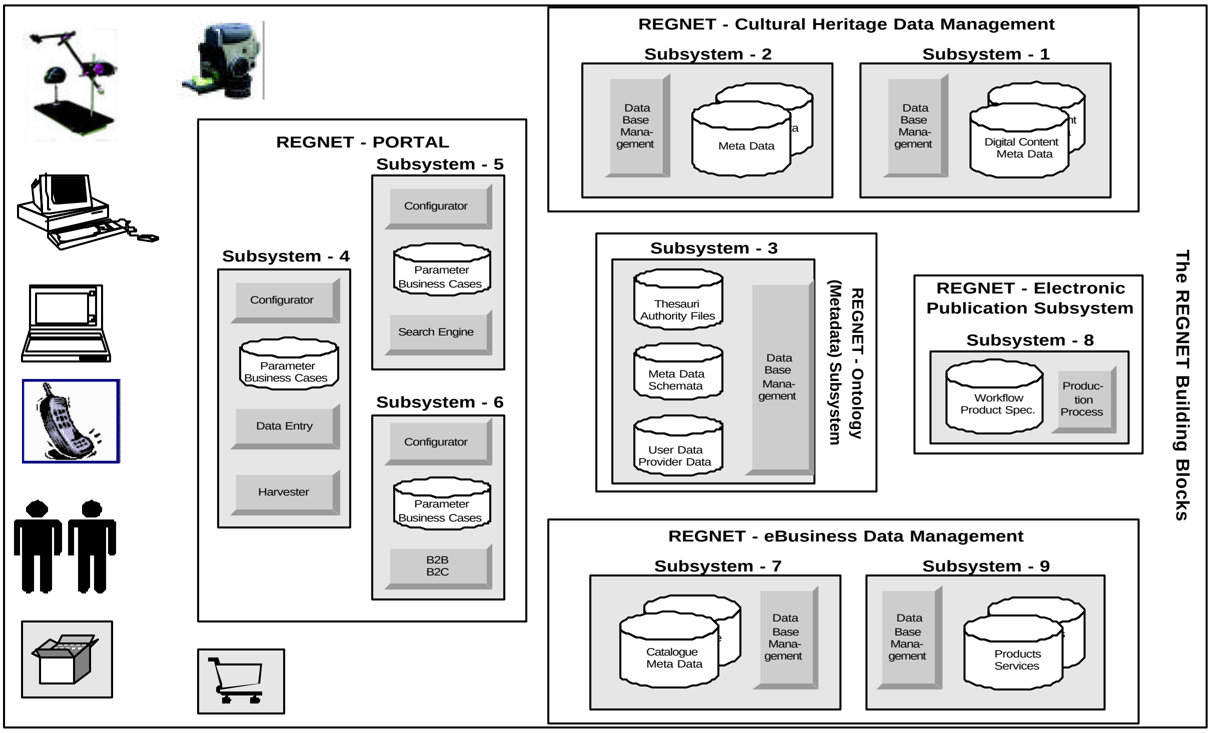


Figure 1: The REGNET Building Blocks

The REGNET-*Portal* enables entry of remote data, distributed search and e-Business functions. The *Cultural Heritage Data Management* allows the search in distributed metadata repositories connected to data of Cultural Heritage content. The *e-Business Data Management* gives access to distributed goods and services catalogues via an e-Business system, the *Ontology Checker* contains specifications of metadata, terminologies etc. which are used in the e-Business and Cultural Heritage field. The *Electronic Publisher* allows the production of personalised digital products based on standardised metadata and work flows.

The interconnection between different REGNET components are presented in the following figure:

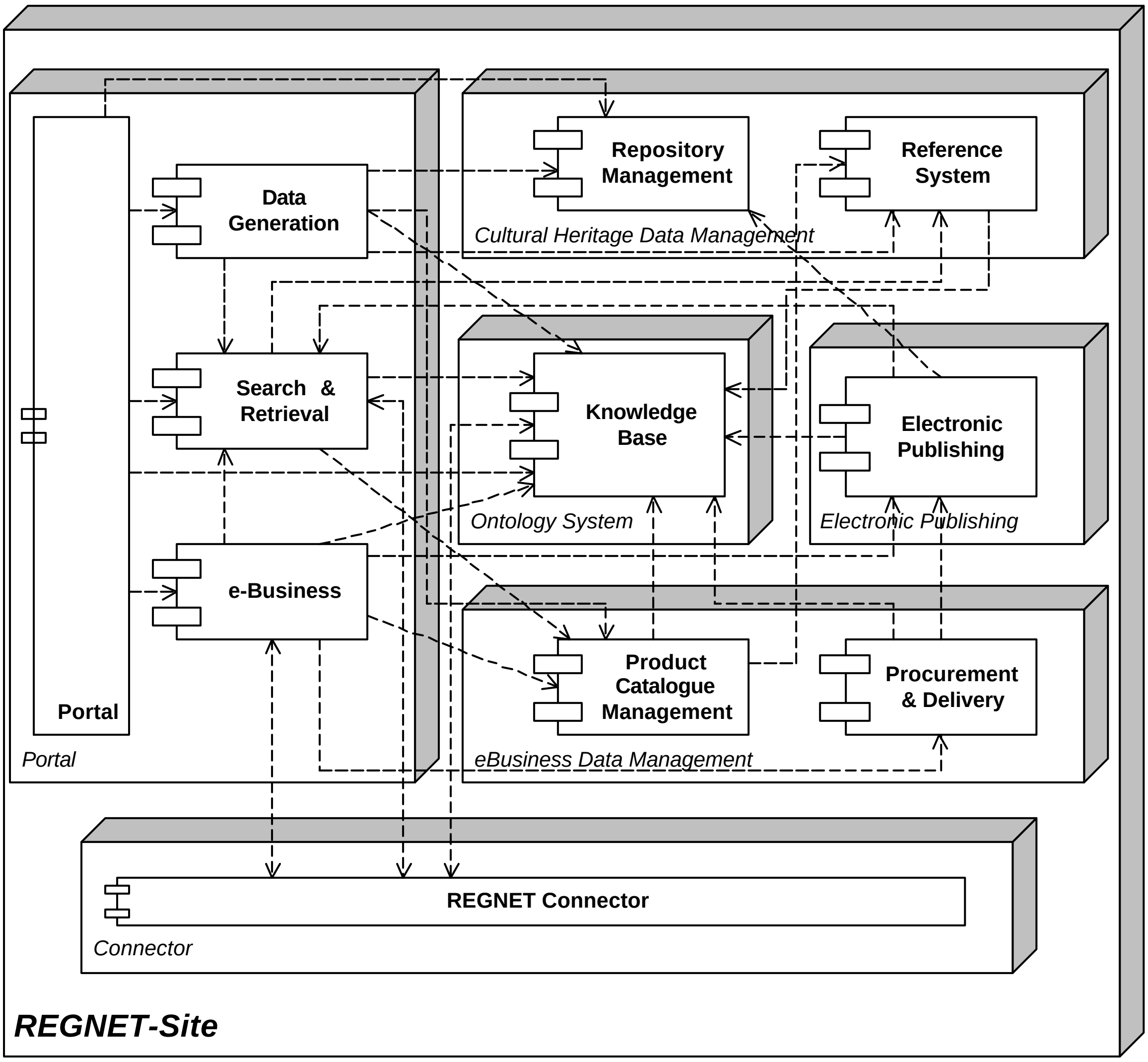
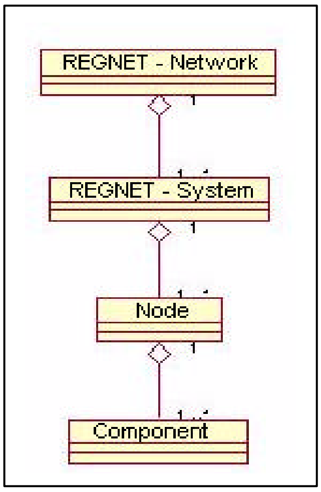


Figure 2 The REGNET-Site



A REGNET-Site can include all REGNET nodes (Portal, etc), but it is possible that a site contains only a subset of nodes, e.g. if a centre is not providing e-Business functions the e-Business related components will not constitute this site.

The conceptual framework for a REGNET Network is presented in the left figure:

- The REGNET Network consists of one or more
- REGNET Systems (Sites), which include one ore more
- REGNET Nodes. A REGNET Node can include one or more
- REGNET Components.

A REGNET Network can look like this:

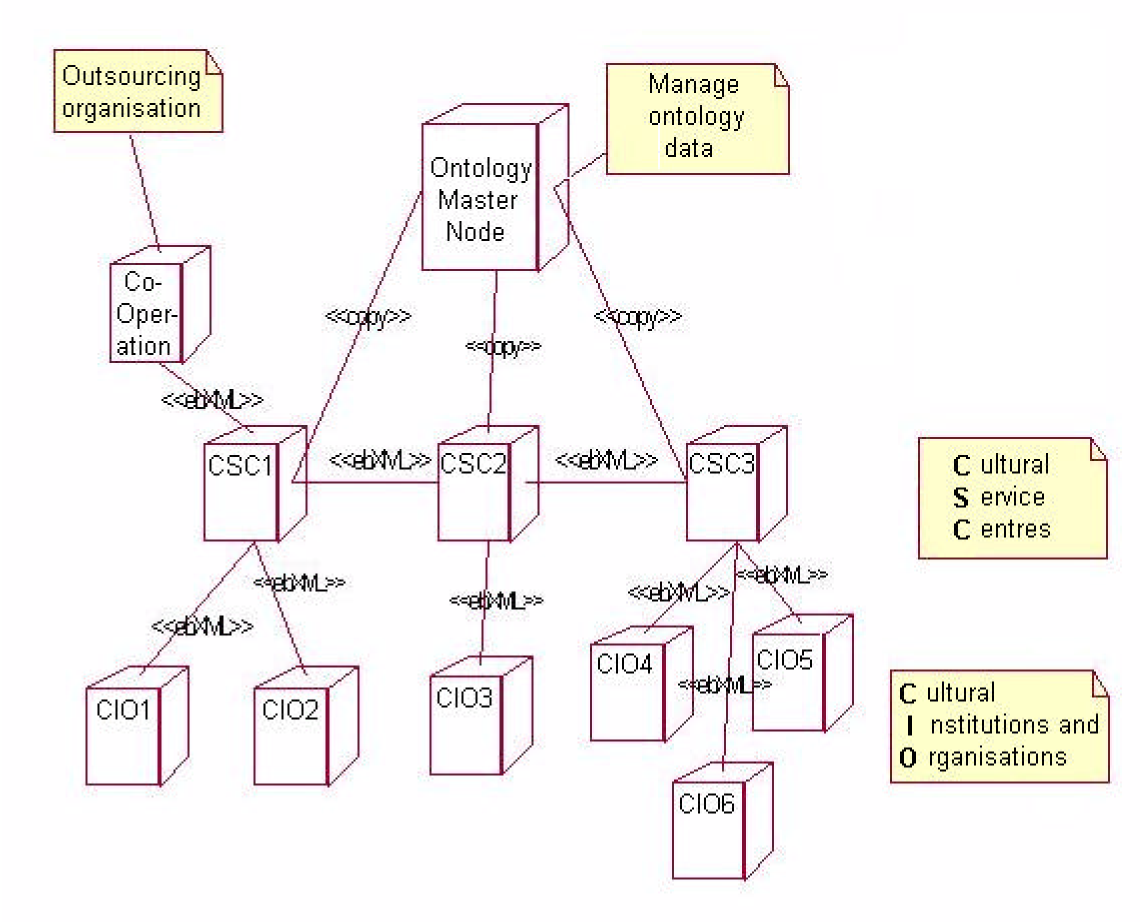


Figure 3: Example REGNET Network

Different REGNET-Sites which build the technical infrastructure of Cultural Service Centres (CSCs) are connected to each other following ebXML-Recommendations. Cultural Institutions or Organisations (CIOs) can be connected to CSCs. Special services like delivery of articles from museum shops might be outsourced to external organisations. The update of the knowledge base within all REGNET-Sites will be done via the Ontology Master Node.

REGNET extends the traditional access facilities to distributed heterogeneous catalogues by integration of product and service catalogues managed by e-Business systems. Because of the large REGNET consortium real world tests covering 4 European regions (over a dozen states) are possible.

The project will observe and take into account the ongoing and recent work of CEN/ISSS (Electronic Commerce Workshop) but will also look into recent activities like ebXML.

REGNET will evaluate a networked organisation model (comparable to the Art Museums Image Consortium/AMICO in the United States) and the integration possibilities of tools and systems for co-operative working.

1.8 Content creation, platform management, Enterprise engineering

Within the REGNET system there are three building blocks which can be considered as vertical functions:

Content creation and management is based on actual standard efforts in the field of the different organisations involved. "Application Ontology" define standardised metadata used in the Library (MARC21), Museum (AMICIO/CIMI/MDA), Archives (EAD, ISAD-G), and Artist (CDWA) domains. To enable "cross domain" searches the REGNET-search & retrieval component uses a mapping of application domain related metadata into Dublin Core metadata. A further development of Z39.50 based methodologies is using XML and provides distributed searches based on the latest technology.

This technique, to make Online Public Access Catalogues (OPACs) interoperable, is essential to REGNET, and will also be applied to the REGNET shopping system providing a virtual catalogue of products and services offered by Cultural Institutions (e.g. museum shops) connected to the REGNET Network. Regarding the collections of the REGNET-content providers it is already possible to develop “themes” which provide a path to relevant information within the REGNET System.

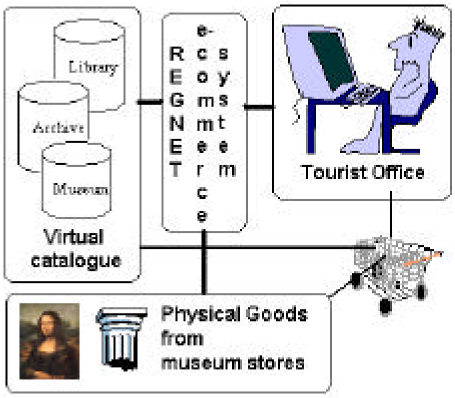
The *platform management* is based on up-to-date internet technology and is the basis for the middle ware being the agent between content and service supplier and the requester (user). It is expected that during the implementation of REGNET the first large trials of systems following the ebXML reference architecture will deliver results and might influence future REGNET-developments. The interconnection of REGNET Nodes will be based on SOAP to enable communication between different development environments (JAVA-PHP) or on direct JAVA to JAVA-communication (RMI). For data hosting XML-related tools will be preferred which does not exclude the integration of proven technologies like relational data base management etc.

The *enterprise engineering* will focus on some selected business processes and functional units: access to distributed catalogues, a shopping cart system, creation of a personalised catalogue based on retrieved data from the 'virtual catalogue' (in printed and electronic form), an internet auction system (e.g. offering duplicates of posters), and a delivery system for physical goods (e.g. goods from museum stores).

On a 'horizontal' basis the XML/XSL-technology is used to structure data semantically and physically. This affects the creation of meta data, describing real (primary) objects (artefacts, nature facts,...), media objects (photos, videos, ...) or bibliographic type objects (literature in the broadest sense). On the other hand all information within business transactions is wrapped within XML structures: Order, Invoice, Despatch, Report, ...). The recent developments in the field of XML/EDI standardisation will be used (ebXML). Another topic will be the definition of information products by appropriate document type definitions and style sheets. This should enable the 'non-media-professional' end user to easily generate catalogues or even CD-ROMs on demand. This might be the first step into the direction to create virtual exhibitions on demand by users themselves.

1.9 User Scenarios

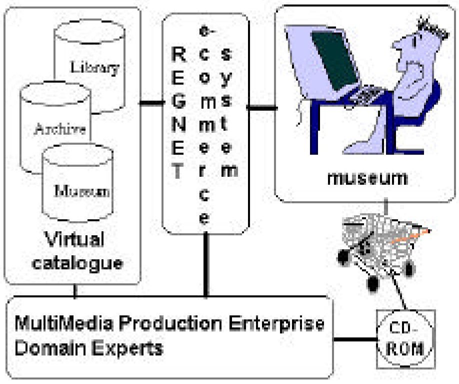
The scenarios outlined below can be considered as a first reference to e-Business processes only.



Business to Consumer (B2C):

A tourist wants to buy articles related to the Cultural Heritage of a region; he/she is interested in physical goods from one or more museum shops as well as in specific surrogate (images) of cultural objects located in museums, archives, or libraries in the region.

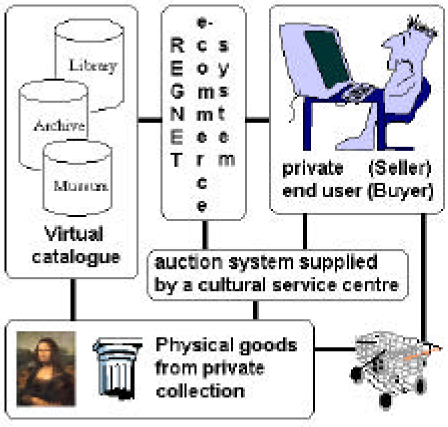
He/she can use a terminal at a tourist office where he/she searches the REGNET virtual catalogue for relevant images and information, browses for articles located in museum stores and places orders.



Business to Business (B2B):

A museum wants to produce a CDROM containing information about specific objects that he/she is interested in and which are related to a region.

The curator at the museum's side selects relevant information (text, images, films); he/she contacts a media producer, selects the basic layout/storyboard for the CDROM and works out together with the industrial partner - eventually with the help of other expert(s) - the final storyboard and the work flow necessary for the production process. He/she signs a contract for the production process and receives the master copy of the CDROM.



Consumer to Consumer (C2C):

The owner of a private collection wants to sell some of his/her assets on a maximum prize. He/She thinks about putting the offers on the internet.

He/She digitises with the help of a REGNET-Partner (e.g. Cultural Service Center/CSC) surrogates of the goods which will be sold. Additional information and expertise are gained after consultation of the Virtual Catalogue. An dedicated Internet Auction System is set up at an Cultural Service Centers (CSC) site and offers bidding to interested user communities. The marketing of this auction is supported by the CSC. This business case can be considered as combination of B2C and B2B functions and demonstrates that a stakeholder can obtain different roles in different business-cases.

1.10 Market Situation

Electronic business is increasing and world-wide eBusiness sales are expected to grow 40 times between 1998 and 2003 by which time they will become over 15% of all sales. The Internet and e-Business are also leading to a surge in new company creation. The market for REGNET systems and services can to some extent be reckoned by the number of museums, libraries and archives in Europe. A rough estimation would deliver a potential number of 3.000 (15% of 20.000 museums) museum shops throughout Europe being candidates to be partner of the REGNET network!

2 Introduction

The REGNET Project supports the introduction of new ways of co-operation between different cultural stakeholders ("CIO": Cultural Institutions and Organisations, Industries, Administrations, etc.) by setting up a technical infrastructure and organisational framework in the light of globalisation and world wide markets. The main activities within a support environment for (Internet) Markets comprise: Content Engineering, Platform Engineering, and Enterprise Engineering.

The REGNET - Cultural Heritage in REGional NETworks project - targets to all of the three areas. The project was introduced under the Action Line 'Access to digital collections of cultural and scientific content' of the European Union IST-Information Society Technologies Programme and 23 partners, cultural organisations (museums, libraries and archives) and IT-industry representatives, from 10 European Union states as well as Bulgaria and Russia are participating.

2.1 Objectives

The deliverable contains the results of Work Area B (Platform Engineering) related to Work Package 1. It refers to the design work to be done for the REGNET demonstrator and a study work related to standards, methods, and tools necessary for the REGNET implementation. The deliverable will include the basic design principles used for the design process and the baseline design as well as the detailed design for all of its components as outlined in the attachment. The tasks covered by this deliverables are:

- T 1.3: Identification of standards to be used
- T 1.4: Development of the System Specification

The aim of this document is to detail REGNET specification. These specification can be split into two parts:

- Requirement artefacts which captures and presents information used in defining the required capabilities of the system.
- Analysis & Design artefact which captures and presents information related to the solution to the problems posed in the Requirements set.

It is very important to collect user requirements (both functional and technical) which reflect as much as possible the real needs so the functional requirements are based on solid ground.

Process description:

The process uses UML notation and is based on an iterative approach.

The task T1.4 is divided in subtasks.

Each partner should use this process according to the subtask partition.

Step 1: *Use Cases and Actors identification*. According to REGNET document and to user requirements list coming from tasks T1.1, the exhaustive list of REGNET Actors and Use Cases is establish.

Step 2: *Expanding Vision document*. Vision document gives an overview of functional aspect of the software as well as marketing elements and technical architecture. Objective of this document is to have a global view of the software based mostly on user and technical requirement on use case basis. Of course, the vision document doesn't give a complete technical architecture but only main aspects of this architecture.

Remark: vision document is establish and detail in IR 1.6.1

Step 3 *Iterations definition*. Iterations are defined on the basis of Use Case List. Each iteration is based on a sub-set of the use case set.

Step 4: Analysis and design

For each iteration repeat :

Detail Use cases : that is describe with text interaction between actors and REGNET System.

Analysis : deduce business objects from use case.

End repeat

Design

Software architecture is defined in parallel from the technical requirements.

3 Methodology

For each of the two parts of this deliverable a dedicated methodology can be identified and will be described in the following sections.

3.1 Identification of standards to be used

Within this section a study on the usage of low cost hardware and reuse of available components (either hard- and software) was done. Standards related to the system development and to the storage and exchange of cultural heritage related data have been investigated. Further information on how the research has been conducted can be found at the beginning of part 1.

3.2 Process for the development of the REGNET System

The aim of this part is to explain the process that we are going to implement in order to produce the REGNET System. The detailed process is given in the Annex. The main aspects are summarised below.

3.2.1 General description

Project risks include the size and complexity of the system and the use of new technologies which are unfamiliar to most of the development teams. The provision and use of a development process to assist development and management teams can provide mitigation of this risk.

Aside: Deficiencies in the Traditional Lifecycle Model

The waterfall model is generally understood to be a serial process comprising the following steps:

- *Requirement capture (Analysis)*
- *Design*
- *Implementation*
- *Verification and Validation (V & V)*
- *Deployment*

This model has been effective in highly structured organisations where requirements change slowly, if at all and where there is sufficient budget and resources allocated to the support functions (V & V). This scenario is usually found in highly regulated applications where risks of failure of the system are well-understood and significant.

The realities of the commercial market place are that requirements change almost continuously. In addition, the risk of failure is discounted to the degree that funding for support activities is insufficient to support the rigorous V & V effort necessary.

*Recent advances in software technology, specifically the advent of modelling tools and languages now make it possible to address the limitations of the waterfall model through a new technique called **iterative development**.*

An iterative development process entails a series of short cycles through a "waterfall-like" process. Initial requirements are identified through standard methods of use case analysis, user interviews, examination of existing systems and so forth. A subset of these requirements are then expressed in a model or set of models built with one of the rapid prototyping tools now commercially available. The model(s) are then evaluated against the initial requirements. The initial model(s), analysis results and new or additional requirements are combined to serve as a baseline for another iteration of modelling. The expectation is that the next iteration of the model would cover a greater percentage of the requirements, hence more closely resemble the planned system. This process repeats until the behaviour of the system is expressed in sufficient detail in the model that implementation becomes a matter of software assembly instead of on-the-fly design.

As this process continues, the live experience with the model will cause some requirements to stabilise. Those stable requirements can then be "frozen" as baselines for development of the actual system, and actual development of the system can commence. Since the development is taking in smaller increments against clearly understood requirements and is based on a working model, the effort is better understood, easier to manage and has a greater likelihood of being completed on time with fewer defects and less rework.

The advantages of this approach are that requirements can evolve throughout the modelling process. The behaviour of the system can be realised and understood more fully early enough in the process where resulting requirements changes can be integrated as a matter of course instead of as an exception after the fact.

*Depending on the architecture of the system and requirements of the user community, it is possible to release the completed increments of the build process into production (internal or external) as standalone deliverables. This is referred to as **Incremental Delivery.***

Incremental deliveries should be between 10% and 20% of estimated overall project effort. The first incremental delivery for a project should focus more on allowing the organisation to become comfortable with the process and less on actual delivery of a workable increment.

Deficiencies in Traditional Lifecycle Models

The advances in software tool technology has given rise to new development methodologies which address the limitations of the traditional "waterfall model". The primary limitation of the waterfall model is the difficulty with timely capture and implementation of changes to requirements. This section documents the approach.

One of the most important aspects of the development process is the management of risk. Risk areas include for example, integration issues, unfamiliar technologies and resource availability. Risks should be actively sought out and addressed up front, rather than deferred.

In practical terms, a risk management approach involves identifying and solving high risk issues in earlier iterations. In other words, early iterations of the project should focus more on eliminating unknowns and less on delivering a "laundry list" of features based on possibly faulty assumptions. With the majority of unknowns resolved early in the project, the later iterations will be more predictable and can proceed according to a well-defined schedule.

A significant class of risk comes from deferring architectural analysis until after the business requirements are fully defined and modelled. We believe that **separation of technical architecture from business and application requirements** into a 'two track' approach as described below assures that architectural risks are brought forward and managed as described above. In this approach, separate tracks are defined to allow architecture to represent a specific focus within the project independent from functional concerns for an application required by the business.

The four principles of the approach to process are summarized in the table below.

Principle	Summary
Incremental and Iterative Development.	Increments should be limited to 2-3 months. This provides visibility for management, allows requirement changes along the way, and would potentially allow intermediate versions of the system to be delivered.
Risk-Oriented Planning.	Based on the 'T' approach, each increment should focus on the list of risks to be evaluated and managed, rather than on producing functionality while still leaving major unknowns. The deliverable of an increment is not just working code, but also the reduction of risks identified before planning of the increment.
Separate of technical and functional architectures.	Following the two track approach, separate tracks are defined to allow architecture to represent a specific focus within the project independent from functional concerns for an application required by the business. A common cause of failure in complex projects is the focus on business functionality, with architecture risks only managed after this functionality is fully defined and modelled. This defers these risks towards the end of the project. The separate architecture track should instead mitigate these risks as early as possible.
Continuous change management.	Since the requirements are generally continual evolving and subject to both internal and external project drivers requirements capture cannot be a one-shot activity performed at the beginning of the project. This requires a continuous requirement management approach coupled with a formal request for modification process. That is tailored to the constraints and methods that are introduced in component based development.

Table 1: Principles of the approach to process

This approach proposes a development approach, which:

- Fills the semantic gap between business analysis and software development.
- Is Business Component centric with support for standards and technical architecture based on distributed object technologies.
- Encourage the reuse of core functionality across applications.

Focuses on architecture to enable the flexible upgrade of pieces of the system, whether they were developed in-house, by third parties, or purchased off-the-shelf.

3.2.2 Project Management

The component based development approach supports project management by providing structure to assist risk-driven incremental planning avoiding high-risk approaches as described below.

The first high-risk planning approach is the "ultimate delivery" syndrome. That is every activity need to be fully completed before starting a new one. Generally this kind of strategy leads to no delivery. Furthermore it is probably not an achievable goal, because the every "finalise" activities provide some feedback that may have impact on activity itself. This can lead to an endless improvement circle. Another key point is that this strategy does not consider the changes that are likely to occur during the project life. In summary, this approach consumes a lot of time.

The following diagram illustrates this kind of planning.

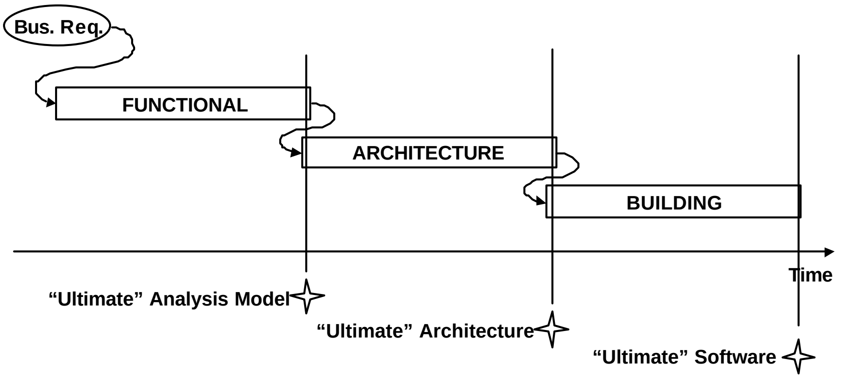


Figure 4: Waterfall "Ultimate Delivery" Syndrome

The second one is the "massive parallelism" syndrome. That is, every activity is run in parallel in order to deliver the product sooner. Generally, this kind of strategy generates a lot of deadlocks because some activities need results from others. In all cases the delivery date is unpredictable. The picture below illustrates this kind of planning.

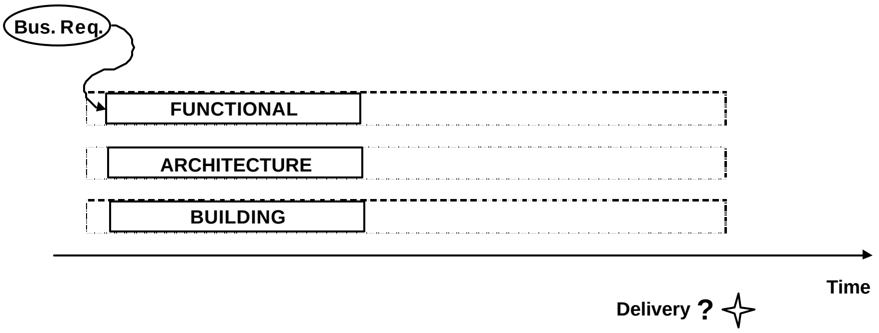


Figure 5: "Massive Parallelism" Syndrome

In order to avoid these two syndromes, actions should be taken.

Syndrome	Actions
“Ultimate delivery”	Incremental and iterative process
“Massive parallelism”	Check and co-ordinate the dependencies between work packages

Table 2: Lifecycle Risks and Actions

These considerations drive towards the iterative and incremental planning model pictured below. It is important to notice that the different activities can be conduct by different teams with corresponding skills.

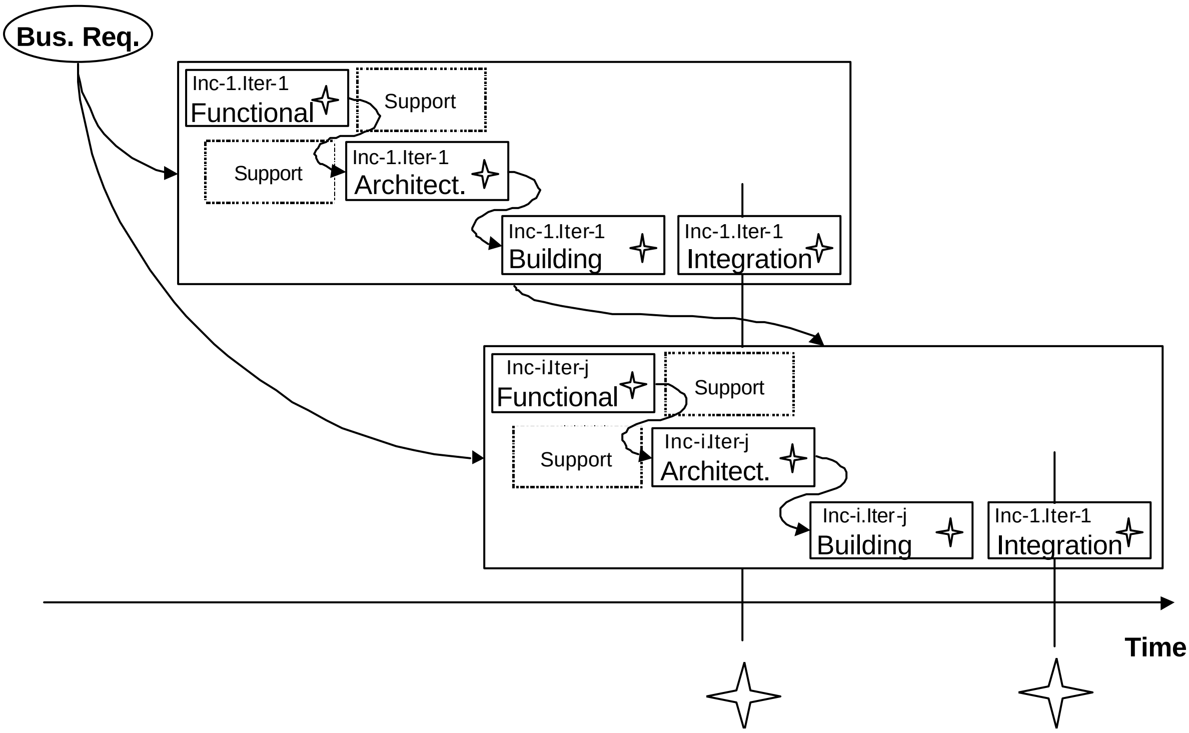
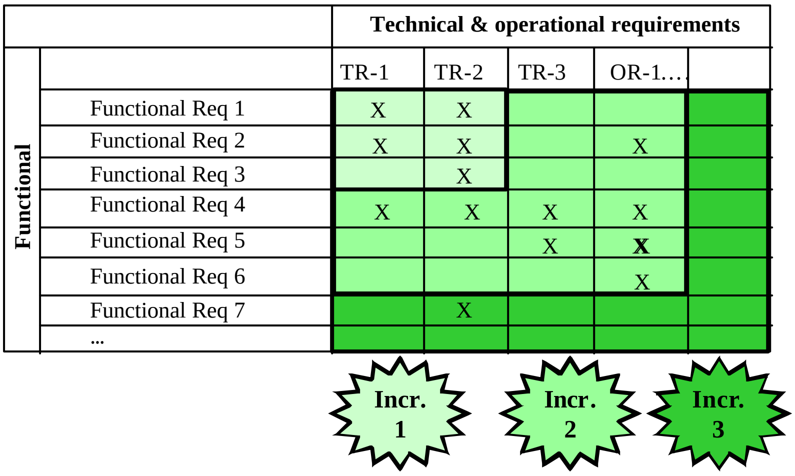


Figure 6: Incremental and iterative process

Increments definition:



		Technical & operational requirements				
Functional		TR-1	TR-2	TR-3	OR-1…	
	Functional Req 1	X	X			
	Functional Req 2	X	X		X	
	Functional Req 3		X			
	Functional Req 4	X	X	X	X	
	Functional Req 5			X	X	
	Functional Req 6				X	
	Functional Req 7		X			
	...					

Figure 7: Increment Definition Model

Part 1 Identification of standards to be used

4 Introduction to relevant standards

4.1 Publishing Standards

In order to provide relevant information about electronic publishing we used the NEDLIB report 3 *Standards for Electronic Publishing* (http://www.kb.nl/coop/nedlib/results/e-publishingstandards.pdf). NEDLIB is a project jointly funded by the European Commission's Telematics for Libraries Programme (http://www.kb.nl/coop/nedlib/). The project was completed in December 2000. The report *Standards for Electronic Publishing* was commissioned by NEDLIB in June 2000, as a supplement to a Process Model for a Deposit System for Electronic Publications. Its aim was to overview of the extent to which publishers are using common standards in their electronic publications. The report was developed through a series of structured interviews with 14 of the largest European publishers.

4.1.1 Offline Publications

Among the offline electronic publications, by which we mean those issued on discrete physical digital media such as tapes, diskettes or, more commonly, optical disks of some kind, are CD-ROM publications the most considered and for the REGNET Project the most relevant.

We have to mention here, that many publishers did not expect CD-ROMs to be the publishing product of the future. Most publications previously published on CD-ROM seem likely to migrate primarily to online publication. Some publishers use CD-ROM as an adjunct to online publication (hybrid publications).

CD-ROM products (unless they are web-browser based) deliver their content through proprietary user interfaces. Many of them are specific to a particular product (only *FolioViews* and *DynaText* were mentioned more than once) and most of them are Windows products.

4.2 Content and Content formats

4.2.1 Content types

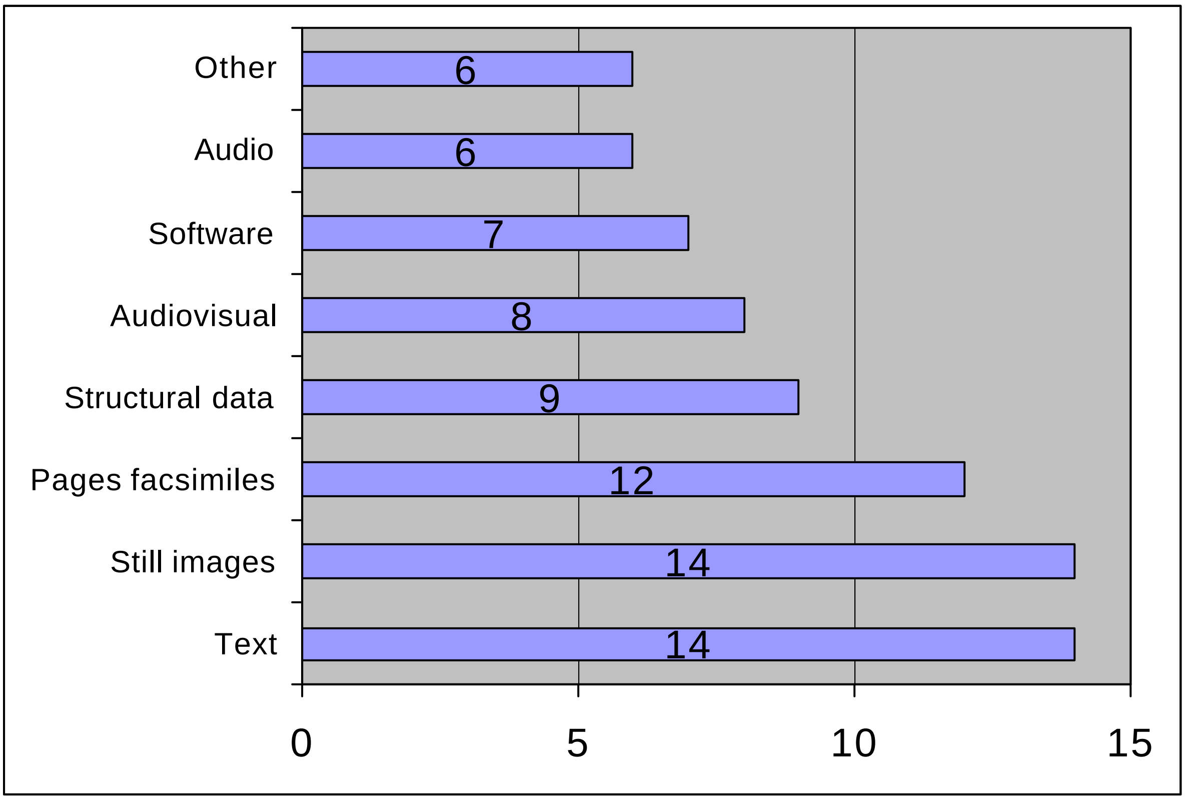


Figure 8: Content types

Text and still images are the most common content type, but also structured data is used very often.

4.2.2 Content formats – text

All of the publishers delivered HTML text to the end-user, but most of them are generating HTML “on-the-fly” from SGML and XML coded text , which also will likely be the case for REGNET.

4.2.3 Content formats – graphics

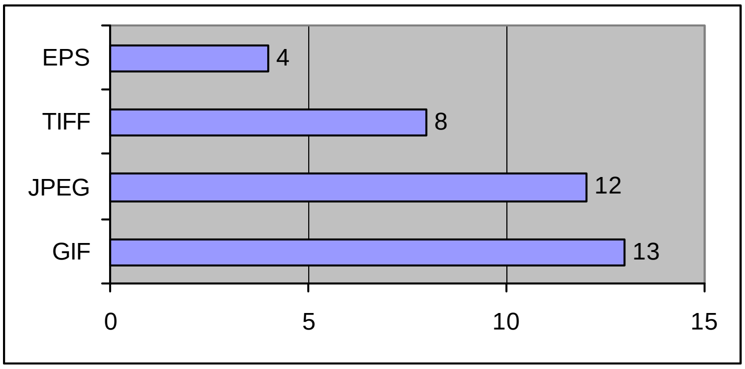


Figure 9: Content formats

Most of the publishers were using more than one format.

4.2.4 Content formats – audio, video, and multimedia

For audio formats WAV, MP3 and Real Audio were the most mentioned ones. For video, respondents are using Quick Time, Real Video and Lotus QuickCam. There was only little consistency from the publishers in terms of theirs approach to formatting multimedia content.

These content formats will also be not very relevant to REGNET, because in REGNET is planned to produce CDROMS as kind of catalogues and therefore text and graphics will be of more interest.

All publishers are also using PDF for delivering page facsimiles to end-users.

A large number of products are dependent on middleware of one sort or another. Most commonly this provides SGML/XML to HTML conversion. If some offline publication is produced in REGNET, a third party product will also be used. This situation will become a little easier as XML-capable browsers become standard, and publishers can use the full capabilities of XML. There is a very rapid trend in the direction of full SGML/XML mark up of text, particularly among journal and professional publishers. It seems that SGML/XML mark up come closest to providing a format that is susceptible to long-term preservation of content. Standardisation of format of the content will also depend on the standardisation of the applications used by the authors.

4.3 Online Publications

The increasing publishing activity in the world Wide Web has been built on the rapid and universal adoption of a standard, and proves the power of the network effect on the adoption of standards.

There is a very high proportion of Web pages that are simply HTML tagged text. File types that are also used are GIF and JPEG image file and a not significant part uses other file types. To access a broad volume of users the public web is relatively simply structured.

The primary issue of preservation of content on the web are not related to file formats. There are other issues to be considered, like identifying the boundaries of a publication in a hyperlinked environment. Can you only usefully preserve a document if you preserve all the documents to which it links?

4.4 Metadata Systems

Metadata can serve a variety of purposes, from identifying a resource that meets a particular information need, to evaluating their suitability for use, to tracking the characteristics of resources for maintenance or usage over time.

The more simplistic definition of metadata is *data about data*. A more detailed definition of metadata is:

metadata is all physical data (contained in software and other media) and knowledge (contained in employees and various media) from inside and outside an organisation, including information about the physical data, technical and business processes, rules and constraints of the data, and structures of the data used by a corporation.

The rise of the World Wide Web has created an urgent need to define standard methods and vocabularies for describing its contents in a consistent and orderly manner. Although the concept of metadata predates the Internet and the Web, world-wide interest in metadata standards and practices has exploded with the increase in electronic publishing and digital libraries. Anyone who has ever tried to find information online using a web search engine will no doubt have experienced the frustration of retrieving a large number of “hits” but then finding themselves unable to narrow the search down more precisely.

Metadata are needed in order to be able to organise the content of electronic resources on the web. Several factors have triggered the need for metadata in businesses today. These include the following:

- Current systems are inflexible and non integrated.
- Users needs potent systems for searching and retrieval information.
- Current systems need interoperation with other systems.

- Existent data warehouses and data marts need to grow.
- Metadata facilitates the interchange of information among systems.

In order to allow a common platform for applying metadata schemas ensuring interoperability between all information resources, it is important to get consensus among all related initiatives. This process is culminating in the emergence of standards on metadata, and in the development of specifications to be used freely by any person, company or institution.

Metadata are applied to different systems and sectors. For example, the SCHEMAS project (http://www.schemas-forum.org) in the last metadata watch report identifies the following domains:

- Industry
- Publishing sector
- Audio-visual sector
- Cultural Heritage sector
- Educational sector
- Academic sector
- Geographical information sector

The REGNET project is only interested on proposals and standards on Cultural Heritage sector, i.e. the library, archive, and museum sub-domains.

Another hand, librarians and digital library users desire integrated access to distributed resources, often in conjunction with resource discovery where searches are across many types of information resources. There is a requirement for effective cross-domain searching of diverse resources including digital library catalogues, government information, museum systems, and archives.

Therefore, it is important to analyse and describe in detail the techniques and the tools that will be used in order to support the cross-domain search efficient. The basic prerequisites, which are necessary for the proper integration of the cross-domain search is the use of Dublin Core metadata and the use of the Z39.50 protocol (see 4.5.2 Cross-Domain Search for details).

4.4.1 Bibliographic Information Objects (Library Sub-domain)

In the library sub-domain, the main reference is the MARC (*Ma*chine *R*eadable *C*atalogue or *C*ataloguing, http://www.loc.gov/marc) standard, developed at the Library of Congress of US (http://www.loc.gov) in 1965-6. However, MARC is neither a kind of catalogue nor a method of cataloguing. MARC is a short and convenient term for assigning labels to each part of a catalogue record so that it can be handled by computers. While the MARC format was primarily designed to serve the needs of libraries, the concept has since been embraced by the wider information community as a convenient way of storing and exchanging bibliographic data.

The MARC communication format is intended to be:

- Hospitable to all kinds of library materials
- Sufficiently flexible for a variety of applications in addition to catalogue production
- Usable in a range of automated systems

A MARC record is composed of three elements: the record structure, the content designation, and the data content of the record. The **record structure** is an implementation of the *American National Standard for Information Interchange* (ANSI/NISO Z39.2) and its ISO equivalent ISO 2907. The **content designation** are the tags, codes, and conventions established explicitly to identify and further characterise the data elements within a record and to support the manipulation of that data are defined by each of the MARC formats. The **content** of the data elements that comprise a MARC record is usually defined by standards outside the formats, such as the *International Standard Bibliographic Description* (ISBD), or other conventions used by the organisation that creates a record.

Since the early 1970s an extended family of more than 20 MARC formats has grown up, e.g. UKMARC, INTERMARC and USMARC, whose paths diverged owing to different national cataloguing practices and requirements. Differences in data content means that editing is required before records can be exchanged.

4.4.1.1 UNIMARC

One solution to the problem of incompatibility was to create an international MARC format which would accept records created in any MARC format. This format was called UNIMARC (http://www.ifla.org/VI/3/p1996-1/unimarc.htm). So records in one MARC format could be converted into UNIMARC and then be converted into another MARC format. The intention was that each national agency would need to write only two programs - one to convert into UNIMARC and one to convert from UNIMARC - instead of one program for each other MARC format, e.g. INTERMARC to UKMARC, USMARC to UKMARC, etc.

So in 1977 the International Federation of Library Associations and Institutes (IFLA, http://www.ifla.org) published *UNIMARC : Universal MARC format*, stating that "The primary purpose of UNIMARC is to facilitate the international exchange of data in machine-readable form between national bibliographic agencies".

In the mid-1980s it was seen necessary to expand UNIMARC to cover documents other than monographs and serials. So a new description of the format was produced in 1987. By this time UNIMARC had been adopted by several bibliographic agencies as their in-house format. So the statement of purpose was amended to include "UNIMARC may also be used as a model for the development of new machine-readable bibliographic formats".

Developments did not stop there. Increasingly a new kind of format - an authorities format - was used. Previously agencies had entered an author's name into the bibliographic format as many times as there were documents associated with him or her. With the new system they created a single authoritative form of the name (with references) in the authorities file; the record control number for this name was the only item included in the bibliographic file. The user would still see the name in the bibliographic record, however, as the computer could import it from the authorities file at a convenient time.

The latest development in the format has come about because of the requirement of European Community countries to produce unified specialised catalogues of their records. In order to produce such unified catalogues they had to adopt a common format for them - UNIMARC.

Bibliographic records in the UNIMARC format are designed for use in automated library systems. Depending on the versatility of the system a range of related functions can be supported by manipulating the data. Two such functions are information retrieval and displaying citations:

- **Information retrieval.** In the UNIMARC format each data element is identified for the purposes of information retrieval. Using computer software, it is possible to search on most of the MARC fields and sub-fields in the record.
- **Displaying citations.** UNIMARC offers a choice of formats for displaying records. Naturally, readers will not want to consult the full MARC record simply because the format is intended not for human perusal but for processing by computer.

4.4.1.2 MARC 21

MARC 21 (http://www.loc.gov/marc/bibliographic/) is not a new format. It is the result of harmonised USMARC (from the Library of Congress of US) and CAN/MARC (from the National Library of Canada) formats. Compatibility had been a feature of the development processes for both formats for many years. In 1997 and early 1998, updates to the formats were issued. These made the format specifications identical. The name both points to the future as we move into the 21st century and suggests the international character of the format, which is appropriate and important given its expanding world-wide use.

The MARC 21 formats are widely used standards for the representation and exchange of authority, bibliographic, classification, community information, and holdings data in machine-readable form. They consist of a family of five co-ordinated formats: *MARC 21 Format for Authority Data*; *MARC 21 Format for Bibliographic Data*; *MARC 21 Format for Classification Data*; *MARC 21 Format for Community Information*; and *MARC 21 Format for Holdings Data*. Each of these MARC formats is published separately to provide detailed field descriptions, guidelines for applying the defined content designation (with examples), and identification of conventions to be used to insure input consistency.

The *MARC 21 Concise Formats* provides in a single publication a quick reference guide to the content designators defined in each MARC format. It provides a concise description of each field, each character position of the fixed-length data element fields, and of the defined indicators in the variable data fields. Descriptions of sub-field codes and coded values are given only when their names may not be sufficiently descriptive.

4.4.1.3 From MARC to XML

There have been discussions on encoding bibliographic records in other than the MARC standard, so that machine-readable bibliographic data can become more open and interchangeable in the Internet environment. In 1995, the Library of Congress began to look into the feasibility of using the SGML standard to encode MARC 21 format. Subsequently, the MARC DTDs (Document Type Definitions, http://www.loc.gov/marc/marcsgml.html) that define the MARC 21 data in SGML format were released.

SGML is the standard for document markup. Within this framework, there is a multitude of specific standards for the markup of particular types of documents, such as books, journal articles, law reports, theses, or manuscript finding aids. Each of these standards is known as a Document Type Definition.

The primary purpose of the MARC DTD project was to create standard SGML Document Type Definitions to support the conversion of cataloguing data from the MARC data structure to SGML (and back) without loss of data. Early 1998, it was announced a software to convert between MARC 21 and SGML.

In 1997, the World Wide Web Consortium published XML as a simplified standard of SGML. It promises to made the Web smarter by allowing web pages to carry not just the layout, but the semantic structure of its content. Since then, many software companies have raced to apply it in various areas. Obviously, bibliographic data is one of the potential areas. Using XML standard and the XSL (eXtensible Stylesheet Language) stylesheets,

- we can create bibliographic records once and publish them in different formats;
- bibliographic records can (will) be directly viewed by the Web browsers, search engines, and potentially library systems without the need of further conversion;
- bibliographic records can be interchanged between XML and MARC without any data loss;
- many of the problems that were inherited with MARC format become insignificant, including those related to romanisation and authority control.

Some relevant implementations and projects of MARC records to XML are the following:

- MARC-XML Conversion Utilities developed by the Library of Congress (ftp://ftp.loc.gov/pub/xmldtd/marcconv_xml.zip).
- XMLMARC DTD (http://xmlmarc.stanford.edu) was developed in the frame of the Medlane project by the Stanford University Medical Centre. They have created a JAVA client/server program for converting MARC records into XML files conforming to an XMLMARC DTD. Software can be downloaded from the XMLMARC web site for the non commercial use.
- MARC to XML to MARC converter (http://www.logos.com/marc/marcxml.asp), developed by the Logos Library System. This program converts a MARC record into a very simple, well-formed XML document. It can also convert the XML document (as is or modified) back into a valid MARC record.
- BiblioML (http://www.culture.fr/BiblioML/en/) is an XML application for UNIMARC records. It is expected that BiblioML conformant records may be created by automatic exporting records form UNIMARC databases and transcoding them to XML.

A lot of link about other implementations and projects of MARC records to XML can be obtained from http://www.oasis-open.org/cover/marc.html.

4.4.2 Archival Information Object (Archival Sub-domain)

Archives exist for the preservation and continuation of the Cultural Heritage and that heritage is made from a variety of cultures, past and current civilisations, artefacts, manuscripts and printed materials and the more recent phenomena of audio-visual materials and electronic documents.

Archives share with libraries the responsibility to remember on behalf of others. Archives differ from libraries in the nature of the things remembered. Libraries collect individual published books and serials, or bounded sets of individual items. The books and journals libraries collect are not unique, at least not in ways that are of general interest. Multiple copies of one publication exist, and any given copy will generally satisfy as well as any other copy. The materials in archives and manuscript libraries are the unique records of corporate bodies and the papers of individuals and families. The records and papers are the unselfconscious by-products of corporate bodies carrying out their functions and responsibilities, and of individuals or families living their lives. All records or documents generated by one corporate body or individual or family are referred to as a collection, or *fonds*.

The standardisation of archival description equires several interrelated standards. First, there needs to be a standardisation of the essential components or categories of description, and the interrelation of these categories. This constitutes the intellectual semantics and syntax for archival description. This is essentially a structural framework which is comprehensive rather than prescriptive. *ISAD(G)* is the International Council of Archives' structural standard for archival description. Second, there needs to be a content standard, with specifications on required and optional categories, how to compose, and what to include in each category. Third, standard rules and authorities are needed for highly controlled information such as geographic, country, and language codes; personal, corporate, and family names; and subjects. Finally, there must be a standard communication format or syntax representing the structural standard. The communication standard enables information sharing between computers and between people. Encoded Archival Description (EAD), based on *ISAD(G),* is an archival description communication standard.

4.4.2.1 ISAD(G)

ISAD(G) (General International Standard Archival Description, http://www.ica.org/ISAD(G)E-pub.pdf) has been developed by the Commission on Descriptive Standards of the International Congress on Archives (ICA, http://www.ica.org). After several review, the last version was submitted for publication for the XIVth International Congress on Archives in Seville, in September 2000.

ISAD(G) provides general guidance for the preparation of archival descriptions. The purpose of archival description is to identify and explain the context and content of archival material in order to promote its accessibility. This is achieved by creating accurate and appropriate representations and by organising them in accordance with predetermined models.

ISAD(G) contains general rules for archival description that may be applied irrespective of the form o medium of the archival material. This set of general rules is part of a process that will:

- ensure the creation of consistent, appropriate, and self explanatory descriptions;
- facilitate the retrieval and exchange of information about archival material;
- enable the sharing of authority data; and
- make possible the integration of descriptions form different locations into a unified information system.

ISAD(G) is intended to be broadly applicable to descriptions of archives regardless of the nature or extent of the unit of description. There are 26 elements that potentially combine to constitute the description of an archival entity and the rules guide the formulation of each of the 26 elements. There is a preferred structure for any given description incorporating elements governed by the rules. Within the structure the elements are grouped in five information areas:

- **Identity Statement:** identifies what is being described and says some significant things about what it is called.
- **Context:** which provides information about the origin and custody of the materials; background, context and provenance.

- **Content and Structure:** which provides information about the subject matter held within the materials, its form, and the way it is arranged.
- **Condition of Access and Use:** informs users about availability.
- **Allied Materials:** tells users about other materials that are significant to the ones being described.

There is a sixth area, **Notes**, in which anything else of interest not otherwise catered for is placed.

Although there are 26 elements of archival description in ISAD(G) only a subset is required to be used in any given description and only 5 are considered essential for the international exchange of descriptive information and all 5 are to be found within the 'identity statement' information area.

The structure and content of the information in each of the elements should be formulated in accordance with applicable national rules. As general rules, these are intended to be broadly applicable to descriptions or archives regardless of the nature or extent of the unit of description. However, the standard does not define output formats, or the ways in which these elements are presented, for example, in inventories, catalogues, lists, etc.

Each rule consist of:

- the name of the element of description governed by the rule;
- a statement of the purpose of incorporating the element in a description;
- a statement of the general rule (or rules) applicable to the elements; and
- where applicable, examples illustrating implementations of the rules.

4.4.2.2 EAD

EAD (Encoded Archival Description, http://www.loc.gov/ead) is a set of rules for designating the intellectual and physical parts of archival finding aids (inventories, registers, indexes, and other documents created by archives, libraries, museums to describe collections) so that the information contained therein may be searched, retrieved, displayed, and exchanged in a predictable platform-independent manner. The EAD rules are written in the form of a SGML Document Type Definition (DTD), which uses coded representations of elements for efficient machine-processing by SGML authoring and viewing software.

EAD was initiated in 1993 as a project of the University of Berkeley and it has had different developments. It is considered an emerging descriptive standard and it is expected changes in the next future following new experiences in its application and the evolution of web browsers and protocols. The number of project based in EAD has being growing constantly (http://www.loc.gov/ead/eadsites.html) and also the availability of vendors assisting with the conversion to EAD of older paper-based finding aids and providers of EAD software products. Currently, EAD is maintained by the Network Development and MARC Standards Office of the Library of Congress in partnership with the Society of American Archivist.

EAD has been designed to preserve and enhance the current functionality of existing registers and inventories: description, control, navigation, indexing, and online or print presentation, both for original materials and digital surrogates. The standard is intended to facilitate interchange of finding aids across institutions, permit the sharing of identical data in two or more finding aids, assist in the creation of union databases, and permit the reuse of the encoded data for multiple output purposes.

The encoding standard consists of three parts: an SGML-compliant DTD, the tag library containing definitions of the elements and attributes and the application guidelines. It is important remark the relationships with other standards. EAD identifies the essential data element within finding aids, it does not define or prescribe intellectual content and is intended to be used with data content standards as ISAD(G). EAD has the potential of achieving the same status for archives as MARC has for libraries all over the world.

EAD is a necessary complementary to ISAD(G). Reflecting ISAD(G), the EAD DTD emphasises the hierarchical nature of archival description and inheritance of description. A diverse set of descriptive elements is available for describing the whole of a collection or fonds. Following the description of the whole, the same elements are available for describing components of the whole, components of the components, and so on. At each level of description, only that description which applies to the entire level is given. Each lower level inherits the description of the containing or superior level. For example, the name of the repository would only be given in the description of the whole, and not repeated in the description of sub-components.

The EAD DTD contains three high-level elements:

- The **<eadheader>** is used to document the archival description or finding aid
- The **<frontmatter>** is used to supply publishing information such as a title page, and other prefatory text.
- The **<archdesc>** contains the archival description itself, and thus constitutes the core of the EAD.

On 1998 was found the EAD Roundtable. The principal function of the Roundtable is to promote the use of EAD by helping implementers find useful information, through the EAD Help Pages web site (http://jefferson.village.virginia.edu/ead/). Some basic practical questions about implementing EAD in XML is addressed in the page (http://jefferson.village.virginia.edu/ead/xml.html).

The MALVINE project (http://www.malvine.org/) has developed two perl scripts that can convert MARC records to EAD. The first of them, available at http://helmer.hit.uib.no/malvine/1marccon.txt, converts a MARC file encoded in MARC into a "readable" MARC file. The second, available at http://helmer.hit.uib.no/malvine/2malvine.txt, which will convert the file from the "readable" MARC into EAD.

EAD tools and resources can be founded in the page of the NCSU Libraries Special Collections website (http://www.lib.ncsu.edu/archives/tech_serv/eadtools.html). It includes macros (for WordPerfect and MS-WORD), templates, and scripts which can be used to minimise the staff, time, and resources needed to produce EAD documents.

4.4.3 Audio-visual Information Object (Sound recording and audio-visual archives)

The purpose of descriptive information in the cataloguing of sound recordings is the need to define the unique and evanescent recording which captures a slice of the space-time continuum in sound. Recordings, once made, might remain unpublished or be published and re-published on a variety of different media both serially and in parallel. Media of distribution can include physical carriers such as discs, tapes and, compact discs, sometimes referred to generically as "products". However, publication in a broad sense might also take the form of a radio or television broadcast or an internet transaction.

An audio-visual archive is different from a conventional archive. It may have the same policies and philosophy and similar aims in the preservation and collection of a particular slice of human activity. This slice may be the large one of an era, century or decade, reflecting the cultural and social life of the times, or it may be a smaller slice which records on one or more materials a particular aspect of a special place or a restricted time.

But the collection policies - the principles of arrangement, organisation, access, security, conservation and preservation of audio-visual materials, are different, or at least require something of a rethink for the archivist as conventionally seen especially if the material is to be included in an audio-visual archive. Technical considerations in particular will have a profound effect upon the audio-visual archives - it is not just a question of preservation of materials, it has to be a question of continual transfer, copying and restoration of the originals.

To describe sound recordings and related audio-visual media, the more important rules are the developed by the International Association of Sound and Audio-visual Archives (IASA), called the IASA Cataloguing Rules.

4.4.3.1 IASA Cataloguing Rules

The IASA Cataloguing Rules (http://www.llgc.org.uk/iasa/icat) is a manual for description of sound recordings and related audio-visual media, developed by IASA (International Association of Sound Archives, http://www.llgc.org.uk/iasa/). The primary purpose of these rules is to establish a norm in audio-visual archives for describing sound recordings conformant with other schemes for bibliographic description. The intention behind this primary purpose is to ensure that the cataloguing of sound recordings can be easily and efficiently incorporated into mainstream cataloguing activity.

IASA is a non-governmental organisation, which has over 380 members (representing archives of music, history, literature, drama and folk-life recordings; radio and television sound archives; collections of oral history, natural history, bio-acoustic and medical sounds; recorded linguistic and dialect studies) from almost 50 countries. It was established in 1969 in Amsterdam to function as a medium for international co-operation between archives which preserve recorded sound and audio-visual documents.

The association supports the professional exchange of information and fosters international co-operation between audio-visual archives in all fields, especially in the areas of:

- acquisition and exchange
- documentation
- access and exploitation
- copyright
- conservation and preservation

The *IASA Cataloguing Rules*' primary emphasis is specify requirements for the description and identification of sound recordings and related audio-visual media, assign an order to the elements of the description and specify a system of punctuation for that description. They are designed for use by sound and audio-visual archives as a guide in the preparation of cataloguing records and as a standard for the exchange of bibliographic information concerning sound and related audio-visual materials.

Special emphasis is given to information that is appropriate to include for different types of content on sound recordings and related audio-visual materials as identified above. Options and alternatives are presented to assist archives and libraries in deciding on the most suitable approach to cataloguing their collections, in order to meet the requirements of public service and archival imperatives.

The IASA Cataloguing Rules are designed to harmonise with the Anglo-American Cataloguing Rules. - 2nd ed., and the International Standard Bibliographic Description (Non-Book Materials) and to be able to be used in MARC or other cataloguing systems.

These rules also cover the concept of multilevel description (division of descriptive information into two or more levels). Multilevel description has traditionally been used in archives and cataloguing agencies which apply the technique of fonds and collection level cataloguing. Depending on the information retrieval requirements and cataloguing policies and resources of the particular archive or cataloguing agency concerned, multilevel description may or may not extend to the level of the individual recording.

4.4.4 Museum Information Object (Museum Sub-domain)

In the museum sub-domain harmonisation efforts are much more difficult due to the heterogeneity of the objects (things) to be described. A coin needs quite different metadata elements compared to a painting. Specific attributes (e.g. Genre in the arts environment) have been introduced to develop site specific domains.

According to the metadata watch report delivered by the SCHEMAS project, the main proposals in this field of arts are the AMICO Data Dictionary (from the Art Museums Image Consortium, http://www.amico.org/), the Categories for the Description of Works of Art (from the Getty Research Institute, http://www.getty.edu/research/institute/standards/cdwa/) and the CIDOC Information Categories (from the International Council Of Museums, http://www.cidoc.icom.org).

4.4.4.1 AMICO Data Dictionary

AMICO (Art Museum Image Consortium, http://www.amico.org/) is a non-profit association of institutions with collections of art, collaborating to enable educational use of museum multimedia, the AMICO Library, which offers access to 65.000 works of art.

To contribute to this digital library, AMICO has developed a set of data specifications (http://www.amico.org/docs/dataspec.html), composed by Text Record Specification, Related Image and Multimedia File Specification and AMICO Data Dictionary.

Each work of art is documented in the AMICO Data Specifications by :

- A catalogue record.
- Associated multimedia files, including at least one image file showing a full view of the work, and any other number of other files.
- A metadata record documenting each multimedia file.

Some works include further multimedia documentation such as audio files and textual documents.

The AMICO DTD is one of the outcomes of the AMICO project. In the context of this project, a set of DTDs has been specially constructed for digital library infrastructure, where multimedia content is an important part of the information held:

- **amico-objects.dtd** (http://www.npaci.edu/DICE/AMICO/Demo/amico-objects-long-DTD.txt).
- **amico-media.dtd** (http://www.npaci.edu/DICE/AMICO/Demo/amico-media-long-DTD.txt)
- **amico-2in1.DTD** (http://www.npaci.edu/DICE/AMICO/Demo/amico-2in1-DTD.txt)

In the COVAX Project they decided to select the last one, because the previous two conceptual levels are covered. There is a great flexibility offered by the media DTD design. A variety of related multimedia files can be linked to the museum object. Furthermore, when proprietary museum information does not map well to the AMICO standard, the media DTD provides a solution for referencing and linking to an XML document that describe the museum object using “local” data. While this practice is not recommended, it provides a temporary solution for reducing “information loss” when mapping to a particular DTD version and when the content owner wishes to make the information available for presentation. Later when conditions are acceptable, the local data represented in XML could be transformed and integrated into the standard repository based on a new version of the DTD.

4.4.4.2 Categories for the Description of Works of Art

Categories for the Description of Works of Art (CDWA, http://www.getty.edu/research/institute/standards/cdwa/) is a product of the Art Information Task Force (AITF) from Getty Research Institute (http://www.getty.edu/research/institute/standards.html). CDWA articulates an intellectual structure for the content of object and image descriptions. The Categories are intended to enhance compatibility between diverse systems wishing to share art information.

The AITF brings together representatives of communities that use and provide art information: art historians, museum curators and registrars, visual resource professionals, art librarians, information managers, and technical specialists. The AITF works to define information about works of art from the researcher's perspective to create a standard for the description of objects and images, a standard that will facilitate the electronic exchange of this information.

CDWA was created to be able to address art works and material culture from all periods and geographic areas and their visual surrogates (slides, photographs, digital images.) The categories range from general to specific with a small number of core elements required. Object information is entered into a consistent template that groups categories and subcategories. There are 31 broad categories with nine essential areas and 225 subcategories that allow information to be specifically recorded.

Accompanying the *Categories* and furthering the exchange of art information, is a technical protocol based on SGML, recommended by the consortium for the Computer Interchange of Museum Information (CIMI, http://www.cimi.org) which promotes an open, standards-based approach to the creation and international interchange of information by museums and Cultural Heritage organisations.

The main objectives of the *Categories* are:

- provide a framework to which existing art information systems can be mapped and upon which new systems can be developed, making easiest the data migration;

- help to give end-users consistent, reliable access to information, regardless of the system in which it resides;
- provide a common ground for reaching agreement on what information should be included in art information systems, and what information will be shared or exchanged with other institutions or systems.

The *Categories* make distinction between information intrinsic to the work (art object, architecture, or group) and information extrinsic to the work. Extrinsic information about persons, places, and concepts related to the work may be important for retrieval, but is more efficiently recorded in separate *authorities* than in records about the work itself. The advantage of storing ancillary information in an authority is that this information needs only be recorded once, and it may then be linked to all appropriate work records.

The *Categories* often deal with differences between information intended for display and information intended for retrieval:

- **Information for display:** information must be in a format that is easily read and understood by users. Such texts can contain all the nuances of language necessary to relay the uncertainty and ambiguity that are common in art information.
- **Information for retrieval:** key elements of information must be formatted to allow for retrieval, often referred to as *indexing* in the CDWA.

4.4.4.3 The CIDOC Information Categories

The *International Guidelines for Museum Object Information: The CIDOC Information Categories* (http://www.cidoc.icom.org/guide/guide.htm), was published in June 1995 by the International Committee for Documentation of the International Council of Museums (CIDOC, http://www.cidoc.icom.org).

CIDOC is the international focus for the documentation interests of museums and similar organisations. It has over 750 members from 60 countries, including documentation specialists, registrars, computer managers, system designers, advisors and trainers.

CIDOC has been committed to the development of museum documentation standards for over 25 years. It has provided a forum for the discussion of standards issues and a focus for practical initiatives by a series of Working Groups. Its members include most of the national museum documentation standards organisations and the leading specialists in this field.

The *Guidelines* are a description of the Information Categories that can be used when developing records about the objects in museum collections. It can be adopted by an individual museum, national documentation organisation, or system developer, as the basis for a working museum documentation system.

The *Guidelines* are designed to support the needs of all disciplines represented in museums, including archaeology, cultural history, art, science and technology, and natural science. For convenience, the text uses the term "object," but this should be taken to cover both objects and specimens.

The *Guidelines* incorporate the following elements:

- a definition of the *Information Categories* that should be used when recording details about objects;
- an outline of the *format rules and conventions* governing how information is entered in these categories;
- comments on the *terminology* that can be used in these categories.

The key objectives of museum documentation supported by the CIDOC Information Categories are the following:

a) *ensure accountability for objects*: they can be used to define the objects that are owned by a museum, identify the objects, and record their location;

b) *aid the security of objects*: they can be used to maintain information about the status of objects and provide descriptions and evidence of ownership in the event of theft;

c) *provide an historic archive about objects*: they can be used to maintain information about the production, collection, ownership, and use of objects and as a means of protecting the long term value of data;

d) *support physical and intellectual access to objects*: they can be used to support access to objects themselves and information about the objects.

The *Guidelines* have a number of main roles:

- as the basis for an *international museum information standard*. This work will be undertaken in close collaboration with other initiatives and CIDOC members;
- as a *model for a practical documentation system*. These *Guidelines* and the related standards can be used as a model by individual museums, national organisations, and system developers when designing systems. These systems can be paper based or computer based, with the Information Categories being comparable to the spaces on recording forms or the fields in a computer system;
- as a basis for *sharing information* within a museum and among museums. The consistent use of these *Guidelines* and the related standards will make it easier to share information;
- as a means of *protecting the long term value of data*. The widespread adoption of these *Guidelines* and the related standards will support the development of high quality records;

The *Guidelines* can be used as the basis of an inventory of the collections or a full catalogue. An *inventory* consists of the basic collections management information about each object in a collection, including the details that are essential for accountability and security. A *catalogue* is a fuller record with additional details about the historic significance of the objects.

The CIDOC Data Model Working Group has created a relational data model (http://www.cidoc.icom.org/model/relational.model/), as a prerequisite to recommending a relational data structure for the interchange of museum information world-wide. This model defines relations between basic 'entities' like person, event, and object. The latest development followed an object oriented approach and has led to the Conceptual Reference Model (CRM, http://cidoc.ics.forth.gr). This model represents an 'ontology' for Cultural Heritage information, i.e. it describes in a formal language the explicit and implicit concepts and relations relevant to the documentation of Cultural Heritage. The primary role of the CRM is to serve as a basis for mediation of Cultural Heritage information and thereby provide the semantic 'glue' needed to transform today's disparate, localised information sources into a coherent and valuable global resource.

4.5 Protocol

4.5.1 Distribution Search and Retrieval (Information Search and Retrieval)

The pursuit of knowledge by scholars, scientists, government agencies, and ordinary citizens requires that the seeker be familiar with the diverse information resources available. They must be able to identify those information resources that relate to the goals of their inquiry, and must have the knowledge and skills required to navigate those resources, once identified, and extract the salient data that are relevant to their inquiry. The widespread distribution of recorded knowledge across the emerging networked landscape is only the beginning of the problem. The reality is that the repositories of recorded knowledge are only a small part of an environment with a bewildering variety of search engines, metadata, and protocols of very different kinds and of varying degrees of completeness and incompatibility. The challenge is to not only to decide how to mix, match, and combine one or more search engines with one or more knowledge repositories for any given inquiry, but also to have detailed understanding of the endless complexities of largely incompatible metadata, transfer protocols, and so on.

This section first identifies the functional requirements of a search and retrieval process, and then review the Dublin Core standard and the Z39.50 protocol.

4.5.2 Cross-Domain Search

This section identifies the functional requirements of a search and retrieval process. These requirements focus on cross-domain search and retrieval for resource discovery. The general requirements comprise three Functional Areas:

- ***Functional Area A*** for Basic Bibliographic Search & Retrieval, with Primary Focus on Library Catalogues
- ***Functional Area B*** for Bibliographic Holdings Search & Retrieval
- ***Functional Area C*** for Cross-Domain Search & Retrieval.

The functional area that it is the main interest of this report is the third one, functional area C, which is based on the previous two areas.

Librarians and digital library users desire integrated access to distributed resources, often in conjunction with resource discovery where searches are across many types of information resources. There is a requirement for effective cross-domain searching of diverse resources including digital library catalogues, government information, museum systems, and archives. A user may desire to send a single search to one or more of these resources. For example, a user within a library might desire to search the local catalogue plus one or more museum systems and an archive to find information related to a specific artist. A library-client configured to do cross-domain searching could send out queries to Z39.50 accessible museum and archive systems configured to support cross-domain searching. Similarly, a museum curator could use a museum-client configured to support cross-domain searching to search the local museum system, or even one or more other museum systems, one or more library catalogues, and government resources that are Z39.50 accessible and configured to support cross-domain searching.

Interoperability in the retrieval of such resources requires standard record syntax. This requirement can be accommodated through the Z39.50 Simple Unstructured Text Record Syntax (SUTRS) and the XML.

Librarians and digital library users engage in a wide range of searching behaviours. Agreements on a core set of digital library searches have evolved:

- Author searches that include searching for an established name heading, searching for names not under any authority control, and searching where only part of a name is used as a search term.
- Title searches that include searching for the entire title, the first part of a title, and searching using one or more words from a title.
- Subject searches that include searching for a complete subject heading, the first part of a subject heading, and searching using one or more subject words as search terms.
- Keyword searches that include high recall searches using one or more words from author, title, subject, and other common access points.
- Boolean searches that include combining search terms with the Boolean operators of AND, OR, NOT.
- Truncation searches where the user wants the system to truncate a single word on a character boundary or to truncate on word boundaries.

Even in this case there is the distinction of a two level search:

- Level 0 can be considered a set of core searches with a general focus on recall rather than precision. Level 0 searches provide basic functionality for common author, title, and subject searches. Level 0 searches are likely to be available in existing implementations.
- Level 1 inherits all Level 0 searches and defines additional searches to provide for more precision in search and retrieval. Implementers are encouraged to provide Level 1 searches. For each of the searches defined in Level 0 or Level 1, a description of expected server behaviour is provided.

4.5.3 Dublin Core Metadata

The Dublin Core Metadata Initiative (DCMI, http://dublincore.org/) is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models. DCMI's activities include consensus-driven working groups, global workshops, conferences, standards liaison, and educational efforts to promote widespread acceptance of metadata standards and practices.

The DCMI is a cross-disciplinary international effort to develop mechanisms for the discovery-oriented description of diverse resources in networked environments such as the Internet.

Dublin Core metadata provides card catalogue-like definitions for defining the properties of objects for Web-based resource discovery systems, and is used to supplement existing methods for searching and indexing Web-based metadata, regardless of whether the corresponding resource is an electronic document or a "real" physical object.

The Dublin Core Metadata Element Set (DCMES) was the first metadata standard deliverable out of the DCMI was an IETF RFC 2413. The DCMES is a set of 15 descriptive semantic definitions. It represents a core set of elements likely to be useful across a broad range of vertical industries and disciplines of study:

Content	Intellectual Property	Instantiation
Coverage Description Type Relation Source Subject Title	Contributor Creator Publisher Rights	Date Format Identifier Language

Table 3: Dublin Core Metadata Element Set

These elements can be used together to create a metadata record to describe networked material in much the same way than a catalogue record describes traditional text resources in a library. The Dublin Core is not intended as a replacement for more complex metadata schemes such as MARC or EAD, but can rather be seen as a means of describing the essence - or 'core' - of digital and non-digital resources.

Each element is optional and may be repeated. Each element also has a limited set of qualifiers, attributes that may be used to further refine (not extend) the meaning of the element. The DCMI has defined standard ways to "qualify" elements with various types of qualifiers. A set of recommended qualifiers conforming to DCMI "best practice" is available in http://dublincore.org/documents/dcmes-qualifiers/.

Another way to look at Dublin Core is as a "small language for making a particular class of statements about resources". In this language, there are two classes of terms--elements (nouns) and qualifiers (adjectives)--which can be arranged into a simple pattern of statements. The resources themselves are the implied subjects in this language. In the diverse world of the Internet, Dublin Core can be seen as a "metadata pidgin for digital tourists": easily grasped, but not necessarily up to the task of expressing complex relationships or concepts.

Several projects using Dublin Core Metadata Set can be founded in http://dublincore.org/projects/.

Some works expressing the Dublin Core within XML/RDF can be founded in:

- http://dublincore.org/documents/dcmes-xml/
- http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/
- http://www.cimi.org/wg/xml_spectrum/XML_for_DC_testbed_rev.doc

Among the tools available at the DC web site are the DC to MARC Converter from the Nordic Metadata Project (http://www.bibsys.no/meta/d2m/) and DC-dot that extracts and validates metadata from HTML resources and MS Office files (http://www.ukoln.ac.uk/metadata/dcdot). Also CIMI is promoting the use of DC as common access points for its projects and they have created a XML Dublin Core DTD.

4.5.4 Z39.50

The development of library, archival and museum networks over the next years will be based on the interconnection of distributed systems, and the use of client/server technology. The implementation of certain key technical standards will allow particular applications such as searching and retrieval to be managed co-operatively between several computer systems. The main standard here is the Z39.50 protocol.

Z39.50 (http://www.loc.gov/z3950/agency/agency.html) is a standard which specifies a client/server based protocol for information retrieval over a network. It specifies procedures and structures for a client to search a database provided by a server, retrieve database records identified by a search, scan a term list, and sort a result set. The protocol addresses communication between corresponding information retrieval applications, the client and server, but it does not address interaction between the client and the end-user. One of the major advantages of using Z39.50 is that it enables uniform access to a large number of diverse and heterogeneous information sources. A client can thus provide users with a unified interface to both the query and the returned search results, regardless of where the information originated.

The protocol was originally proposed in 1984 for use with bibliographic information. As interest in Z39.50 broadened, the Z39.50 Implementers Group (ZIG) was established, in 1990. Members include manufacturers, vendors, consultants, information providers, and universities, who wish to access or provide access to various types of information, including bibliographic, text, image, financial, public utility, chemical, and news.

The latest edition of Z39.50 was approved in 1995 by the National Information Standards Organisation (NISO), the only organisation accredited by the American National Standards Institute (ANSI) to approve and maintain standards for information services, libraries and publishers. Z39.50 is also recognised world-wide and will soon become an international standard replacing the Search and Retrieve (SR) Standard approved by the International Organisation for Standardisation (ISO) in 1991. The new ISO standard will be known as ISO 23950.

Z39.50 has been adopted widely to provide access to many classes of information, including but not limited to:

- Bibliographic Data
- Museum Information
- Government Information Resources (both nationally and internationally)
- Scientific and Technical Data
- Geospatial Data

4.5.4.1 Functionalities

The Z39.50 standard is rich in functionality and provides many optional features, although it also allows for very simple implementations as well. Because Z39.50 is designed modularly, most Z39.50 features can be developed and added incrementally, and still maintain backward compatibility with older implementations.

Some of the features provided by Z39.50 include:

- ***Initialising***: When a session is first established between a client and a server, Z39.50 provides the means for negotiating options that are to be used throughout the remainder of the session. This includes the supported Z39.50 features, the default character set, the default language, and the protocol version. It also provides a mean to authenticate the user.

- ***Searching***: Z39.50 provides the means of searching one or more databases using a structured query in a well-known search format. The query may contain Boolean operators, fielded search terms, proximity searching, weighted search terms, truncation specification, relation specifiers (e.g. less than, equal), etc. Because the query is formulated in a well-known structure, there is no need to lexically parse a query received by the server. The Z39.50 protocol has also demonstrated extensibility to support search based on generalised pattern-matching techniques. These techniques will be increasingly important for finding abstract information such as chemical configurations, gene sequences, fingerprints, faces, video imagery, and numeric trend data.
- ***Presenting Records***: An extensive mean of accessing information from a set of search results is provided through the protocol. This includes requesting specific ranges of search results (e.g., records 10 through 20), specific elements in a record (e.g., title and author), specific variants of the record (e.g., MS-Word and HTML, or English and French), search term highlighting, etc. The server may also include other metadata information (scores, word frequency counts, document lengths, etc.) to enable proper merging of results obtained from multiple distributed databases.
- ***Maintaining Multiple Search Results***: Z39.50 provides the capability for creating, naming, storing, and retrieving from one or more search result sets. This capability also permits the client to apply new query criteria to a previously created result set (e.g., refining a search).
- ***Browsing***: Z39.50 provides the ability to browse a window of index terms or specific fields within a database (e.g., title or author).
- ***Sorting of Results***: Z39.50 offers the means to sort a set of search results based on any given sort criteria.
- ***Controlling Access***: Not only does Z39.50 enable authentication on a per-session basis, but it also allows authentication on a per-operation basis for cases where access to specific databases or records is controlled.
- ***Explaining Server Capabilities***: Z39.50 provides an extensive mechanism for a client to learn the capabilities provided by the server. This includes the databases available for searching, the search access points, etc.
- ***Controlling Resources***: Z39.50 provides the means for clients to cancel a search or presentation request in the middle of an operation, while continuing to maintain an open session with the server. It also permits clients to request resource reports that include accounting information on the number of searches, retrievals, etc. performed by the user. This functionality is particularly important in fee-based online services.
- ***Extended Services***: A number of additional services are also available through Z39.50. This includes the ability for a client to set up a persistent or periodic queries performed by the server on behalf of the client, and the ability for users to order a document. Z39.50 also provides the ability to perform database maintenance operations, such as database updates, record insertion, deletions, etc.

4.5.4.2 Profiles

Various groups have been developing Z39.50 profiles. Profiles are basically customisations of the standard to particular communities of implementers with common applications requirements. A profile may include a whole range of agreements: for example, agreements to use or not to use specific optional features; agreements on particular attribute sets and record syntax to be used (including perhaps the definition and registry of new attribute sets and/or record syntax to support the community in question); and even agreements on what extended services will be used (including, again, definitions of new extended services that the profiles community may want to use). Often it is doubtful how much meaningful interoperability will be possible between one Z39.50 implementation that is built according to a given profile and another which is not aware of the specific profile. Examples of profile work include GILS, the Government Information Locator System; the Museum Interchange Profile being developed by the Computer Interchange of Museum Information (CIMI) group; the Digital Collections profile under development by the Library of Congress; the (revised)

WAIS profile; profiles for applications involving remote sensing and geospatial data, and a cataloguing profile under development by the National Library of Australia.

We can view profile development within the Z39.50 community as a response to the lack of other well-defined processes for establishing standards for attribute sets and record interchange syntax to support various semantic classes of information resources (such as museum information); these are developed as Z39.50 profiles rather than separate parallel standards that are used in conjunction with Z39.50.

4.5.4.3 Tools

There are a lot of (free) tools implementing the Z39.50 information retrieval protocol. For example:

- Perl client-side API (http://www.miketaylor.org.uk/tech/nz/index.html), a module providing a Perl interface to Z39.50.
- JZKit (http://www.k-int.com/jzkit/), a pure JAVA toolkit for building distributed information retrieval systems, with particular focus on the Z39.50 standard. The aim is to provide a comprehensive toolkit that anyone can use to leverage the huge potential of the JAVA programming environment into information retrieval applications, be they retrieval clients or servers providing access to a resource.
- ZAP! (http://www.indexdata.dk/zap/), an Apache module which allows you to build simple WWW interfaces to Z39.50 servers, requiring only a minimum of knowledge about the Z39.50 standard.
- PHP/YAZ (http://www.indexdata.dk/phpyaz/), an extension to the popular web server language PHP that implements Z39.50 origin (client) functionality.

4.5.5 Dublin Core and Z39.50

Recognising the value of querying distributed Dublin Core–based databases via Z39.50, a number of organisations within the Dublin Core and Z39.50 communities are now exploring the feasibility of creating a specific Dublin Core profile.

Z39.50 offers exciting possibilities for the exchange of Dublin Core metadata (http://dublincore.org/documents/dc-z3950/). What we need is an attribute set with not too many, but not too few Use attributes. Their semantics should be well enough defined that they are clear, but not so tightly defined that they apply to only a few subject domains. The Dublin Core elements seem to satisfy these requirements and have the additional benefit of already being accepted as being applicable to many domains. That last point makes an attribute set based on Dublin Core superior to any other arbitrary list of attributes.

One of the points of discussion/development in the Dublin Core community is "qualification". Qualification allows the document developer to say more things about a Dublin Core element than just the type of the element. An example of a qualifier is Scheme, which can be used to qualify the source of a subject heading. Such qualification is intrinsic to Z39.50 attribute sets and is defined in the Dublin Core attribute set.

Because qualification is native to Z39.50 attribute sets and not a topic of debate in the Z39.50 community, we are going to unilaterally resolve one of the Dublin Core issues. We will aggregate Creator, Contributor and Publisher into a single Abstract attribute of Name and provide Semantic Qualifiers to specify the original semantic intent of those elements.

One of the clear strengths of the BIB-1 attribute (attribute set of Z39.50 protocol, which is used for queries against bibliographic data, http://lcweb.loc.gov/z3950/agency/defns/bib1.html) is the number of Use attributes available. The semantics of many of those Use attributes can be made available in the Dublin Core attribute set through judicious use of Semantic Qualifier and Content Authority attributes.

4.5.5.1 Related Initiatives and Projects

In this section some related projects are presented, which follow the same guidelines with our proposed approach. All of them use the Dublin Core metadata and the Z39.50 Protocol for the implementation of the projects and more specifically the main interest is the integration of cross-domain search with the use of the above standards.

An important consortium working to bring the capabilities of Z39.50's computer-to-computer protocol to museums is the Consortium for Computer Interchange of Museum Information (CIMI, http://www.cimi.org). CIMI was founded in 1990 by the Museum Computer Network (MCN) to develop, test, and disseminate standards to support management of museum information.

In 1998 the *CIMI Profile, Release 1.0H: A Z39.50 Profile for Cultural Heritage Information* provided a set of technical specifications to instruct the interaction between clients and servers for information retrieval from remote sources. The profile gives specifications for searching databases, selecting desired information, and formatting for transfer from the server to the client.

What makes this unique to Cultural Heritage information as opposed to library information? CIMI has created an attribute set which specifies access points that are relevant to Cultural Heritage information, these attributes are created to work within resources such as museum object record databases or image databases. In the Profile, CIMI also provided a schema that identifies possible units of information that may be found in a Cultural Heritage information database. The schema does not dictate the naming of fields (which reflect the needs of a local organisation) but provides a standard way of referencing these fields. The schema also provides specifications for retrieval of images or preformatted data (such as SGML documents.) Additionally, to support interoperability between libraries and Cultural Heritage institutions, the Profile gives guidance for transferring information in USMARC record syntax.

Beginning in March 1998 and concluding in January 2000, CIMI conducted a test of Dublin Core (DC) for describing museum information. With 15 repeatable, qualified elements, it was thought that DC would be a practical standard to implement for recording the wide variety of museum information. From this database of about 200,000 records, CIMI was able to review the kinds of information used to manage a wide range of collections from around the world. Using the findings from this examination an XML-DTD for museum information and best practices guidelines for museums planning to use DC were created. CIMI found that DC could work well as a location tool for museum collections, but that problems with DC semantic refinements adversely affected the integrity of detailed object records.

Currently, CIMI is working on Phase II of their DC project. One of the first goals will be to complete and publish the best practices guide. The guide will provide interpretation of the DC element set specific to the museum environment. In response to concern that DC is unable to adequately describe museum objects with enough specificity, qualifier elements are being identified and recommended to enhance description. Also, Resource Description Framework (RDF) is being examined as a possibility to aid the exchange of metadata. This Framework has the ability to handle transfer of various metadata standards. The test will involve applying RDF to the existing test-bed record set from the previous phase of the DC experiment.

4.5.5.1.1 Arts & Humanities Data Service

The Arts & Humanities Data Service (AHDS, http://ahds.ac.uk/) is a federal organisation, consisting of a central executive and five service providers encompassing archaeology, history, textual studies and the performing and visual arts. The goal of this organisation is to build an integrated system capable of providing a seamless whole to the user of the electronic resources available from each service provider.

AHDS gateway (http://ahds.ac.uk/public/metadata/discovery.html) is a Z3950 gateway to provide integrated access to the distributed holdings of five service providers.

4.5.5.1.2 EULER

The aim of the EULER (http://www.emis.de/projects/EULER) project is to provide strictly user-oriented, integrated network based access to mathematical publications. The EULER service intends to offer a "one-stop shopping site" for users interested in Mathematics. Therefore, an integration of all types of relevant resources is necessary:

- Bibliographic databases

- Library online public access catalogues
- Electronic journals from academic publishers
- Online archives of pre-prints and grey literature
- Indexes of mathematical Internet resources

A common user interface - the EULER Engine - will assist the user in searching for relevant topics in different sources in a single effort. The EULER system will be designed as an open, scaleable and extensible information system.

Library users and librarians from mathematics in research, education, and industry will actively participate. EULER is an initiative of the European Mathematical Society, and especially focuses on real user needs.

Standard, widely used and non-proprietary technologies such as HTTP, SR/Z39.50, and Dublin Core will be used. Common resource descriptions of document-like objects will enable interoperability of heterogeneous resources.

4.5.5.1.3 Electronic Library Image Service for Europe

The Electronic **Library Image Service for Europe** (ELISE, http://nile.dmu.ac.uk/elise/e2_intro.html) service will operate on a client/server model, making use of Z39.50 and Dublin Core. In the ELISE II prototype, the catalogue data supplied by participating institutions is mapped to DC and displayed alongside thumbnail images.

4.6 Data Formats

Information interchange standards define the technical framework for exchanging information, whether between systems in a single institution or among systems in multiple institutions.

One of the broadly diffused standards is SGML. SGML (Standard Generalised Markup Language), which attempts to define a universal standard for electronically exchanging data, although SGML predates the Internet and the web.

XML (eXtensible Markup Language, http://www.w3.org/XML/) was developed to exchange of information in the web. XML is a subset of SGML, which maintains the important architectural aspects of contextual separation while removing nonessential features of SGML. Furthermore, XML is an open technology standard of the World Wide Web Consortium (W3C, http://www.w3.org/), the standards group responsible for maintaining and advancing HTML and other web-related standards.

There is a common agreement in REGNET to use XML.

4.6.1 Extended Markup Language (Information Interchange)

Information interchange standards define the technical framework for exchanging information, whether between systems in a single institution or among systems in multiple institutions.

This section is first dedicated to study XML and related technologies, and then present the main features of RDF. Finally, we will see the classical EDI approach.

4.6.1.1 XML

XML (eXtensible Markup Language, http://www.w3.org/XML/) is an internet standard defined by the World Wide Web Consortium (W3C, http://www.w3.org) to provides mechanisms for content specification and constraint. XML is a markup language for documents containing structured information.

XML defines the physical and logical structure of an XML *document* containing an arbitrary graph of entities with associated attributes, relationships and associated constraints. XML is expressed as an Extended Backus-Naur Form (EBNF) grammar which specifies the rules which govern the well forming i.e. validity of an XML document.

XML was created so that richly structured documents could be used over the web. The only viable alternatives, HTML and SGML, are not practical for this purpose. HTML comes bound with a set of semantics and does not provide arbitrary structure. SGML provides arbitrary structure, but is too difficult to implement just for a web browser. Full SGML systems solve large, complex problems that justify their expense. Viewing structured documents sent over the web rarely carries such justification.

A document is well-formed if it obeys the syntax of XML. A well-formed document is valid only if it contains a proper Document Type Definition (DTD) and if the document obeys the constraints of that declaration (element sequence and nesting is valid, required attributes are provided, attribute values are of the correct type, etc.).

An important consortium related with XML is OASIS (Organisation for the Advancement of Structured Information Standards, http://www.oasis-open.org). OASIS is the international, not-for-profit consortium that advances electronic business by promoting open, collaborative development of interoperability specifications. OASIS operates XML.ORG, the non-commercial portal that delivers information on the use of XML in industry. The XML.ORG Registry provides as an open community clearinghouse for distributing and locating XML application schemas, vocabularies and related documents. OASIS serves as the home for industry groups and organisations interested in developing XML specifications.

More information on XML can be obtained from http://www.xmlinfo.com.

4.6.1.2 Advantages of XML

XML offers many advantages as a format for data exchange. These include:

- XML is already established as an open, platform-independent and vendor-independent standard by an international organisation.
- XML does not rely on any programming language or proprietary API, and a range of XML APIs are available to create, view, and integrate XML information. Leading XML API presently include DOM, SAX, etc.
- The cost of entry for XML information providers is low. XML documents can even be created by hand using any text editor. There are also XML-based WYSIWYG editors with support for XSL rendering that allow creation of XML documents.
- XML's tag structure and textual syntax are easy to read and are clearly superior to HTML for conveying structured information.
- The cost of entry for automatic XML documents producers and consumers is low, with the set of available development tools already growing. Major computer vendors currently offers complete, free, commercially unrestricted XML parsers written in JAVA. A variety of other XML support tools, including implementations of the XML APIs, are available on the Internet.
- XML supports the international character set standards of extended ISO Unicode.

4.6.1.3 Specifications Related to XML

The W3C also defines other specifications closely related to XML; for XML document content manipulation and for content transformation into a suitable presentation:

- **Document Object Model (DOM, http://www.w3.org/DOM/)** – Defines an interface model for the traversal and manipulation of XML documents. DOM level 2 is a W3C recommendation of November 2000 and DOM level 3 is a January 2001 working draft from W3C. DOM is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page.
- **eXtensible Style Language (XSL, http://www.w3.org/Style/XSL/)** – XSL Version 1.0, released in November 2000, is a recommendation of the W3C. It defines transformation rules for application to XML documents for translation to any required presentation format. Successor to other more established stylesheet languages, notably CSS version 1.0, which was recommended for adoption by the W3C in December 1996 or CSS version 2.0 which was recommended in May 1998. A great number of tools are listed at

http://www.w3.org/Style/XSL/. More information on XSL can be obtained from http://www.xslinfo.com. XSL consists of three parts:

- an XSL Transformations (XSLT, http://www.w3.org/TR/xslt): a language for transforming XML documents;
- an XML Path Language (XPath, http://www.w3.org/TR/xpath): an expression language used by XSLT to access or refer to parts of an XML document (XPath is also used by the XML Linking specification, http://www.w3.org/TR/xlink/)
- an XML vocabulary for specifying formatting semantics (XSL Formatting Objects).

- **XML Schema** (http://www.w3.org/XML/Schema)- Is a recommendation from W3C since 2001, May 2. XML Schemas define shared mark-up vocabularies, the structure of XML documents which use those vocabularies, and provide hooks to associate semantics with them.. Some tools to works with XML Schema are:
  - Free Web-form access to XSV (http://www.w3.org/2001/03/webdata/xsv), an XML Schema Validator from University of Edinburgh/W3C;
  - Free Web-form access to XSU (http://www.w3.org/2001/03/webdata/xsu), an upgrade transform from the 20001024 to the 20010330 version, from University of Edinburgh/W3C;
  - Free download of self-installing version of XSV (ftp://ftp.cogsci.ed.ac.uk/pub/XSV/XSV12.EXE) for WIN32.
  - IBM XML Schema Quality Checker (http://www.alphaworks.ibm.com/tech/xmlsqc). This tool, written in JAVA, reads Schemas conforming (or alleging to conform via their namespace declaration) with the latest specs, and attempts to determine if they are 100% valid under all the various constraints that apply to schemas. If it determines they are not valid, it attempts to explain the problem in language that a schema novice could probably understand.

As well as being a standard recommended by the W3C, XML has been adopted by the Object Management Group (OMG, http://www.omg.org/) as the standard for exchange of data between modelling tools (XMI: XML Metadata Interchange). XMI is a new standard combining UML and XML, which specifies an open information interchange model that is intended to give developers working with object technology the ability to exchange programming data over the Internet in a standardised way, thus bringing consistency and compatibility to applications created in collaborative environments.

4.6.1.4 XML Tools

There are a lot of tools and software related with XML (http://www.xmlsoftware.com), such as:

- XML parsers, both *validating* and *non-validating*. Non-validating parsers check XML documents for syntactical correctness to determine if they are well-formed. In addition, validating parsers check that documents conform to meta-data contained in DTDs which describe constraints on structure and content. There are many examples of XML parsers and source code available.
- XML editors, stand-alone products for creating and editing XML documents. Many XML editors are sensitive to DTDs and so can enable you to easily produce valid XML documents.
- XML browsers, tools that perform client-side processing of XML content with an associated XSL style sheet.
- XML application servers or web publishing systems with an application development framework to build applications that deliver XML documents to any browser client or device, format or media type.
- Other XML applications, such as:
  - Software for indexing XML documents and search and retrieval of content in XML documents.
  - Tools for creating, editing, and managing XML-based web services.

- Utilities and tools sit on top of XML processors (sometimes included) and provide additional processing and services.
- General and specific tools for converting to and/or from XML.

Another directory where we can found XML resources is http://www.xmldir.com.

4.6.1.5 RDF

RDF (Resource Description Framework, http://www.w3.org/RDF/) is a W3C recommendation from February 1999. It is a metadata model for the WEB.

RDF is an framework for encoding, modelling, and exchanging metadata, which uses XML as its encoding syntax. RDF provides interoperability between applications that exchange machine-understandable information on the Web. RDF metadata can be used in a variety of application areas; for example: in *resource discovery* to provide better search engine capabilities; in *cataloguing* for describing the content and content relationships available at a particular Web site, page, or digital library; by *intelligent software agents* to facilitate knowledge sharing and exchange; in *content rating*; in describing *collections* of pages that represent a single logical "document" and in many others.

At the heart of RDF is a very simple three part model: metadata is about a ***resource***, the resource has one or more ***properties***, and each property has a ***value*** as shown in Figure 10. This mechanism allows us to build *labelled directed graphs*.

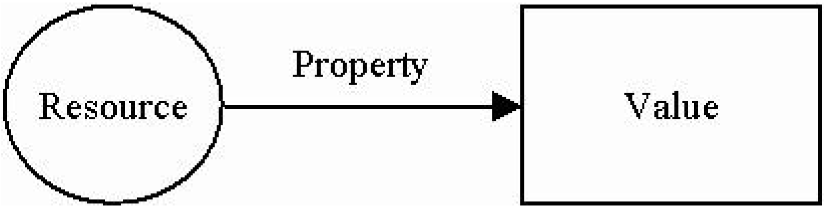


Figure 10: The basic RDF model.

- A resource is anything that can have a URI; this includes all the Web's pages, as well as individual elements of an XML document.
- A property is a resource that has a name and can be used as a property, for example Author or Title. In many cases, all we really care about is the name; but a property needs to be a resource so that it can have its own properties.
- A statement consists of the combination of a resource, a property, and a value. These parts are known as the 'subject', 'predicate' and 'object' of a statement.

A number of tools have been created by developers working with RDF. For an in-depth treatment of these, consult the W3C RDF home page (http://www.w3.org/RDF/). A number of other listings are available, including XML.com (http://www.xml.com/pub/rg/97), XMLhack (http://www.xmlhack.com/list.php?cat=28) and Dave Beckett's RDF Resource Guide (http://ilrt.org/discovery/rdf/resources/).

RDF is an attempt to empower effective creation, exchange and use of metadata on the World Wide Web, and therefore addresses many of the same issues as the Dublin Core. Unlike the Dublin Core, however, RDF makes few assumptions about semantics whilst instead defining a coherent structure (for the expression of semantics) and recommending a powerful transport syntax in the form of XML. As such, the combination of structure and syntax offered by RDF in XML is a suitable complement to the semantically rich Dublin Core element set (http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/).

4.6.1.6 Making Resources Available to Internet Search Engines and Browsers

The common trend for institutions is to make their resources available via the Internet. Presenting web resources files on the Internet using Dublin Core Metadata Element Set (or extending Dublin Core) in RDF/XML format should be considered because it facilitates resources discovery across the Internet by search engines or by specialised web search interfaces. One method to enhance resource discovery is to build an RDF/XML metadata file (or database link) that describes the resources accessible from a particular site and how to connect to that resource. This file is stored in the web root directory and exists as a source for any search engine or a RDF supported web browser, such Mozilla and Netscape.

An interesting project here is the Mozilla RDF/Z39.50 Integration Project (http://www.mozilla.org/rdf/doc/z3950.html). The aims of this project are make Z39.50 data sources accessible for searching from within Mozilla and find an RDF representation of Z39.50 attribute sets. There are thousands of networked Z39.50 servers in existence already. It should be possible to identify some mechanism whereby the Mozilla user interface allows people to send queries to these servers and have the resulting records appear within the standard Mozilla bookmarks/site maps interface.

4.6.1.7 EDI/EDIFACT

EDI (Electronic Data Interchange) is a standard format for exchanging business data. The standard is American National Standards Institute X12 and it was developed by the Data Interchange Standards Association. ANSI X12 is either closely co-ordinated with or is being merged with a European standard, EDIFACT (http://www.edifact-wg.org).

An EDI message contains a string of data elements, each of which represents a singular fact, such as a price, product model number, and so forth, separated by delimiter. The entire string is called a data segment. One or more data segments framed by a header and trailer form a transaction set, which is the EDI unit of transmission (equivalent to a message). A transaction set often consists of what would usually be contained in a typical business document or form. The parties who exchange EDI transmissions are referred to as trading partners.

UN/EDIFACT model is well adapted for collaboration in a private community with same economic interest. This community define its own rules, specialise its semantics and define exchange modes (application rules). Moreover this community defines its participant directory. Such a community is quite closed and facilitate security aspects but discourage new participants.

Internet e-Business model is more open. It specifies interoperability based on technical choices (TCP/IP, domain names, SET[1], etc.) and some common universal process (RosettaNet, OBI, Biztalk, etc.). In such a system every publisher push offer and client are free to choose for each transaction. This model is called market-place model and is used as a media between publishers and clients. These market-place are light to settle comparing to an EDI community.

An important consortium related with UN/EDIFACT is UN/CEFACT (www.unece.org/cefact). UN/CEFACT is the United Nations body whose mandate covers world-wide policy and technical development in the area of trade facilitation and electronic business. Headquartered in Geneva, it has developed and promoted many tools for the facilitation of global business processes including UN/EDIFACT, the international EDI standard. Its current work programme includes such topics as Simple-EDI and Object Oriented EDI and it strongly supports the development and implementation of open interoperable, global standards and specifications for electronic business.

4.7 e-Business

Achieving full benefits from the Internet economy requires the transformation of business processes across the entire commerce chain – the chain of business interactions that define how a company operates, regardless of industry. Integrating business-to-business (B2B) applications with internal systems represents the cutting edge of this transformation process.

But locking in e-Business benefits means being able to respond quickly to change, including ongoing changes to applications, business models and the technologies of the new economy. Transforming the commerce chain requires an adaptive integration architecture, one that is built from the ground up to deliver the agility needed for what will be an extended journey.

Integrating B2B initiatives with existing systems is not easy. The typical IT landscapes is a mosaic of applications, databases and data warehouses of every variety, including hundreds of legacy systems.

Multiple systems are also used for communications and data transport, including software for message queuing, publish-and-subscribe, straight-through processing, file transfer and email. And

[1] Secure Electronic Transaction is a system for ensuring the security of financial transactions on the Internet.

there is still heavy reliance on traditional eBusiness, including EDI over value-added networks and direct connections between partners using proprietary formats and protocols.

Diversity is not the only problem. Creating B2B integration solutions that can be easily deployed, scaled, and dynamically managed is a critical issue, as is the ability to manage partner relationships. Without tools to manage and control the B2B environment, the best integration technology in the world cannot deliver sustainable benefits.

Since REGNET will use XML as data format language, we will only describes the e-Business technologies based on XML.

4.7.1 Virtual Business

The main problem of the REGNET e-Business subsystem is to define in detail the infrastructure that it will be used in a way that will reflect the best solution for the project.

The best solution for the REGNET project is to integrate in the final system a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet.

This section tries to define in detail all modern initiatives and approaches of defining the current standards and some new that are not yet completed. The basic research has been made in order to define and clarify the best solution for the e-Business subsystem.

We describes based technologies necessary to eBusiness. These technologies are based on the XML syntax. Three layers can be distinguish:

- The first layer contain approaches dedicated to a business domain (example: RosettaNet).
- The second layer contain frameworks which are business domain independent (ex: ebXML).
- The last layer contain technologies relative the Web services: UDDI, WSDL and SOAP.

We have to notice that these technologies evolve currently very quickly.

This section is organised according to the following subsections. First, we give a global description which introduce the problem. Later, we present XML based B2B approach. Finally, we make a synthesis according to REGNET context. Conclusion of this study is that ebXML is the more accurate B2B approach according to REGNET context: it provides a business domain independent framework dedicated to smaller companies.

4.7.1.1 Global Description

Achieving full benefits from the Internet economy requires the transformation of business processes across the entire commerce chain – the chain of business interactions that define how a company operates, regardless of industry. Integrating business-to-business (B2B) applications with internal systems represents the cutting edge of this transformation process.

Integrating B2B initiatives with existing systems is not easy. The typical IT landscapes is a mosaic of applications, databases and data warehouses of every variety, including hundreds of legacy systems.

These challenges are commonly address by two technologies families :

- IAI (Internet Application Integration) / B2B: set of processes and technologies dealing with the structural integration of software applications **between** organisations.
- EAI (Enterprise Application Integration) / A2A (Application To Application): set of processes and technologies dealing with the structural integration of software applications **inside** an organisation.

These concepts are illustrated by the following schema:

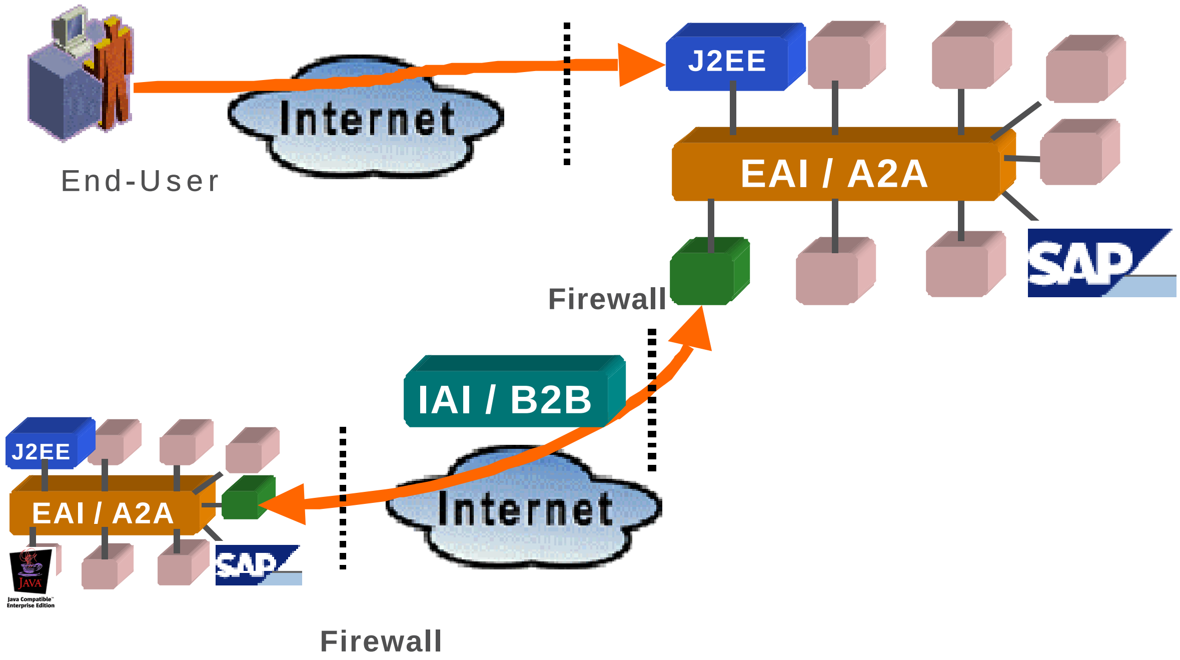


Figure 11: Commonly used technologies

This document address the IAI/B2B problem according to REGNET context in order to build an Internet media based collaborative environment, that is an environment which allows distributed people working together by using Internet as communication tool. The use of Internet imply TCP/IP as transport protocol but it doesn't gives more indication concerning higher level protocol (WEB/HTTP, CORBA/IIOP, DCOM/RPC, JAVA/RMI, etc.). Moreover collaboration means exchange of information between software applications, this information is related to business objects and so there is a need for a standardisation of these objects. Standardisation deal with data formats to represent object but also to a representation of objects' behaviour. This approach has been initiated by EDI (Electronic Data Interchange) with standard like UN/EDIFACT (United Nations Electronic Data Interchange For Administration, Commerce and Transport). New standards emerge due to the increase of Internet based plat-forms : e-Business technologies.

There is a great number of works concerning e-Business standardisation. Most relevant to REGNET are:

- Virtual Business (http://www.ontology.org/)
- RosettaNet (http://www.rosettanet.org)
- CommerceNet (http://www.commercenet.com)
- BizTalk (http://www.biztalk.org)
- ebXML (http://www.ebxml.com)

Web Services is a new model, essentially object-oriented programming for web-based objects. The SOAP 1.1 (http://www.w3.org/TR/SOAP/) specification was a step in this direction in that it described how to use XML formatted messages for requests and responses. UDDI is the piece of the puzzle that will enable businesses to find these services. Web Services Description Language (WSDL) is the XML vocabulary that will describe services and service providers.

According to Sun :

XML	UDDI	SOAP or XP	ebXML
syntax	list services: yellow, white, green pages	access to services	electronic services; message exchange

Table 4: Web Services

4.8 XML based communities

In this chapter there is a presentation of the most known approaches for designing and developing e-Business. Special care will be given in the specific way that all these approaches satisfy international standards and how these approaches define new and innovative standards.

4.8.1 RosettaNet

RosettaNet (http://www.rosettanet.org) is an independent, non-profit consortium dedicated to the collaborative development and rapid deployment of open Internet-based business standards that align processes within the global high-technology trading network.

RosettaNet, which represents more than 400 companies and over 1 trillion USD in annual information technology, electronic components and semiconductor manufacturing revenues, provides RNIF (RosettaNet's Implementation Framework) as a framework for electronic business process development and implementation.

Developed by means of an industry-wide partnership, RosettaNet standards address the Information Technology (IT), Electronic Components (EC) and Semiconductor Manufacturing (SM) supply chain, including manufacturers, distributors, resellers, shippers and end users.

RosettaNet model is not limited to the standardisation of exchanged data but takes into account business process. Some other business domain (oil, car, etc.) are looking to this model in order to generalised it to their business.

4.8.1.1 Model

PIPs: RosettaNet Partner Interface Processes™ (PIPs™) define business processes between trading partners. RosettaNet PIPs are specialised system-to-system XML-based dialogs that define business processes between trading partners. Each PIP specification includes a business document with the vocabulary, and a business process with the choreography of the message dialog. PIPs apply to the following core processes: Administration; Partner, Product and Service Review; Product Introduction; Order Management; Inventory Management; Marketing Information Management; Service and Support; and Manufacturing.

Dictionaries: RosettaNet dictionaries provide a common set of properties for PIPs™. The RosettaNet Business Dictionary designates the properties used in basic business activities. RosettaNet Technical Dictionaries provide properties for defining products. RosettaNet dictionaries reduce confusion in the procurement process due to each company's uniquely defined terminology. The RosettaNet Business Dictionary designates the properties for defining business transactions between trading partners, and RosettaNet Technical Dictionaries provide properties for defining products and services.

RosettaNet Implementation Framework: The RosettaNet Implementation Framework (RNIF) provides exchange protocols for quick and efficient implementation of PIPs. The RNIF Core Specification provides exchange protocols for quick and efficient implementation of RosettaNet standards. The RNIF specifies information exchange between trading-partner servers using XML, covering the transport, routing and packaging; security; signals; and trading partner agreement.

Product and partner codes : Product and partner codes in RosettaNet standards expedite the alignment of business processes between trading partners.

RosettaNet's standards programs provide a benchmark for quality in RosettaNet solutions.

4.8.1.2 Standards

RosettaNet architecture is based on recognise standards from EDI, Web (HTTP, SSL, XML). For the layer architecture model, RosettaNet is based on OBI (Open Buying on Internet) consortium. The Open Buying on the Internet Consortium is a non-profit organisation dedicated to developing open standards for business-to-business Internet commerce. The OBI Consortium is an independent organisation managed by CommerceNet (http://www.commerce.net/).

RosettaNet has announced (27 April 2001) plans to integrate support for the UN/CEFACT (www.unece.org/cefact) and OASIS (www.oasis-open.org) backed ebXML Messaging Services Specification in future releases of RosettaNet's Implementation Framework (RNIF).

4.8.1.3 Tools

There is a great number of tools supporting the RosettaNet model:

- BEA WebLogic Collaborate business-to-business (B2B) platform.
- WebMethods B2B for RosettaNet.
- Mercator e-Business Integration Broker.
- PTC, Active Software, Netfish, etc.

4.8.1.4 RosettaNet vs Oasis

RosettaNet is working on standards for supply chain : members of RosettaNet are mainly from electronic devices and IT industry. They began with business modelling in order to standardised process then they define common XML semantic for each product and at the end they define technical architecture.

Oasis is working on technical architecture : in this field they define ebXML in order to standardise electronic commerce platforms.

The best solution for the REGNET project is to integrate in the final system a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet.

The final decision was finalised after of great concern and reference to several thematic and important areas. The requirements that supported the final decision and proposition constitute the basic prerequisites for the proper operation of an e-Business system. Topics like cost, compatibility, standardisation, integration, open source software, support of SMEs, support of common message structure, enhancement of competitiveness and several more constituted the basic prerequisites and the metrics for the final proposition that satisfies all the above.

4.8.2 BizTalk

BizTalk (http://www.biztalk.org) is a Microsoft software that provides the infrastructure and tools for building e-Business communities. The core of BizTalk Server offers business document routing, transformation, and a rules-based tracking infrastructure. BizTalk Server offers the features, outlined below, with which you quickly build dynamic business processes—easily integrating applications and business partners and using public standards to ensure interoperability.

BizTalk Server 2000 enables a rapidly building and deployment of integrated business processes within the organisation and with the trading partners. It can get the solutions to market more quickly, using fewer resources, which allows to move swiftly to respond to the customer needs and competitive pressures. BizTalk Server 2000 offers a suite of tools and services that make building business processes and integrating applications fundamentally faster. Secure, reliable trading partner relationships can be quickly implemented independent of operating system, programming model, or programming language.

4.8.2.1 Build Dynamic Business Processes

The BizTalk Server infrastructure helps the quick integration, management, and automation of dynamic business processes by exchanging business documents among applications, within or across organisational boundaries. With all the tools that companies need for business process orchestration, BizTalk Server helps to build processes that span not only applications, but also businesses, over the Internet. Graphical tools make it easy for business analysts and application developers to model and implement solutions for business.

4.8.2.2 Easily Integrate Applications and Business Partners

BizTalk Server 2000 makes it easy for developers to integrate applications and businesses together. Business analysts and application developers benefit from a host of rich graphical tools for building XML schema, performing schema transformation, establishing trading partner relationships over the Internet, and tracking and analysing data and documents that are exchanged. With support for XML and standard Internet technologies, BizTalk Server 2000 extends the features of traditional e-Business and electronic data interchange (EDI) to entire e-Business communities.

4.8.2.3 Ensure Interoperability Using Public Standards

With extensive support for public standards and specifications, such as XML, EDI, Hypertext Transfer Protocol (HTTP), and security standards like public key encryption, digital signatures, and encryption, BizTalk Server 2000 ensures interoperability and security with several applications and business partners.

4.8.3 ebXML

ebXML (http://www.ebxml.org) is an exchange architecture based on standard business scenarios (select-buy-pay-deliver) and not on precise data formats. Its aim is to allow application witch are based on this model to easy enable collaboration.

The ebXML initiative (http://www.ebXML.org) is developing specifications to enable a single global electronic marketplace based on an open public XML-based infrastructure. The goal is to enable the global use of electronic business information in an interoperable, secure and consistent manner by all parties. A primary objective of ebXML is to lower the barrier of entry to electronic business in order to facilitate trade, particularly with respect to small- and medium-sized enterprises (SMEs) and within developing nations. The ebXML initiative is sponsored since June 1999 by UN/CEFACT and OASIS and is an open public initiative with now approaching two thousand participants.

EbXML participants are split into work-groups : technical infrastructure, directory and register mechanism, transport and routing, business process modelling and based components specification. The whole proposition is a public standard in the spirit of free software. EbXML gets functions and semantic of current standards (Edifact, X. 12, HL7).

4.8.3.1 Description

The ebXML specifications provide a framework where SMEs, business analysts, software engineers, and other organisations can create consistent, robust and interoperable e-Business services and components seamlessly within an integrated global e-Business market.

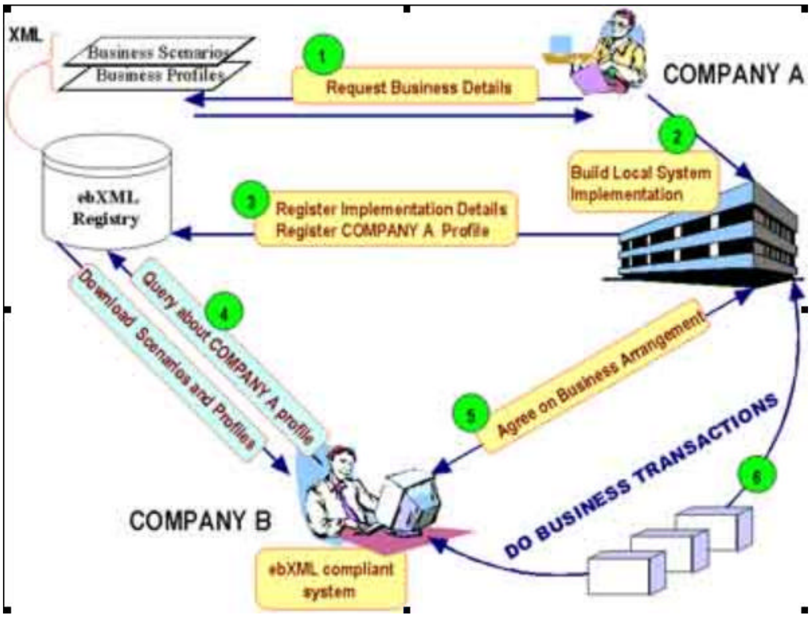

XML
Business Scenarios
Business Profiles
1
Request Business Details
COMPANY A
2
Build Local System Implementation
ebXML Registry
3
Register Implementation Details
Register COMPANY A Profile
4
Query about COMPANY A profile
Download Scenarios and Profiles
5
Agree on Business Arrangement
DO BUSINESS TRANSACTIONS
6
COMPANY B
ebXML compliant system

Figure 12: The ebXML Approach; Automating Business-to-business Interactions

The actual architectural model of ebXML uses two views to describe the relevant aspects of all business interactions (see Technical Architecture specifications http://www.ebxml.org/specdrafts/approved_specs.htm). These two views stem from early work on OpenEDI by UN/CEFACT, and are part of the UN/CEFACT Modelling Methodology (UMM). The first view is the Business Operational View (BOV), which addresses the semantics of business data transactions, and associated data interchanges (see Figure 13). The architecture for business transactions includes operational conventions, agreements and mutual obligations and requirements. These specifically apply to the business needs of ebXML trading partners.

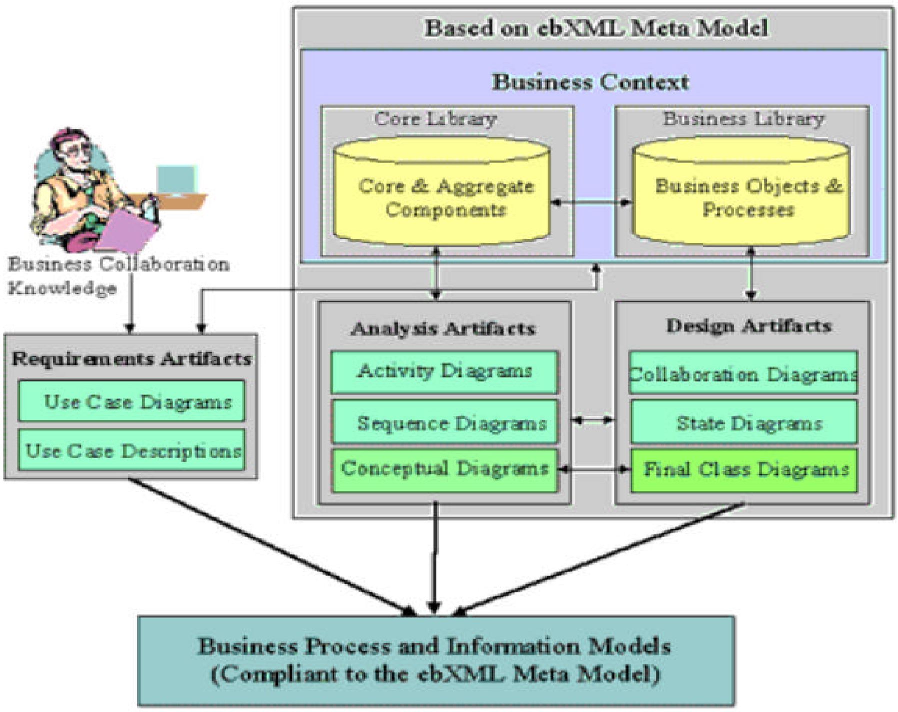


Figure 13: The Business Operational View

Second is the Functional Service View (FSV), which addresses the supporting services and meeting the deployment needs of ebXML (see Figure 14). The implementation of the FSV of ebXML has three major phases; implementation, discovery and deployment and then the runtime phase. The implementation phase deals specifically with procedures for creating an application of the ebXML infrastructure. Then the discovery and deployment phase that covers all aspects of the actual discovery of ebXML related resources and self-enabled into the ebXML infrastructure. And after that, the run time phase that addresses the execution of an ebXML scenario with the actual associated ebXML transactions.

FSV focuses on the information technology aspects of functional capabilities, service interfaces and protocols including the following:

- Capabilities for implementation, discovery, deployment and run time scenarios;
- User application interfaces;
- Data transfer infrastructure interfaces;
- Protocols for interoperation of XML vocabulary deployments from different organisations.

In order to deliver on the BOV and FSV, integral to the ebXML architecture is the Registry System. An ebXML Registry provides a set of distributed services that enable the sharing of information between interested parties for the purpose of enabling business process integration between such parties by utilising the ebXML specifications.

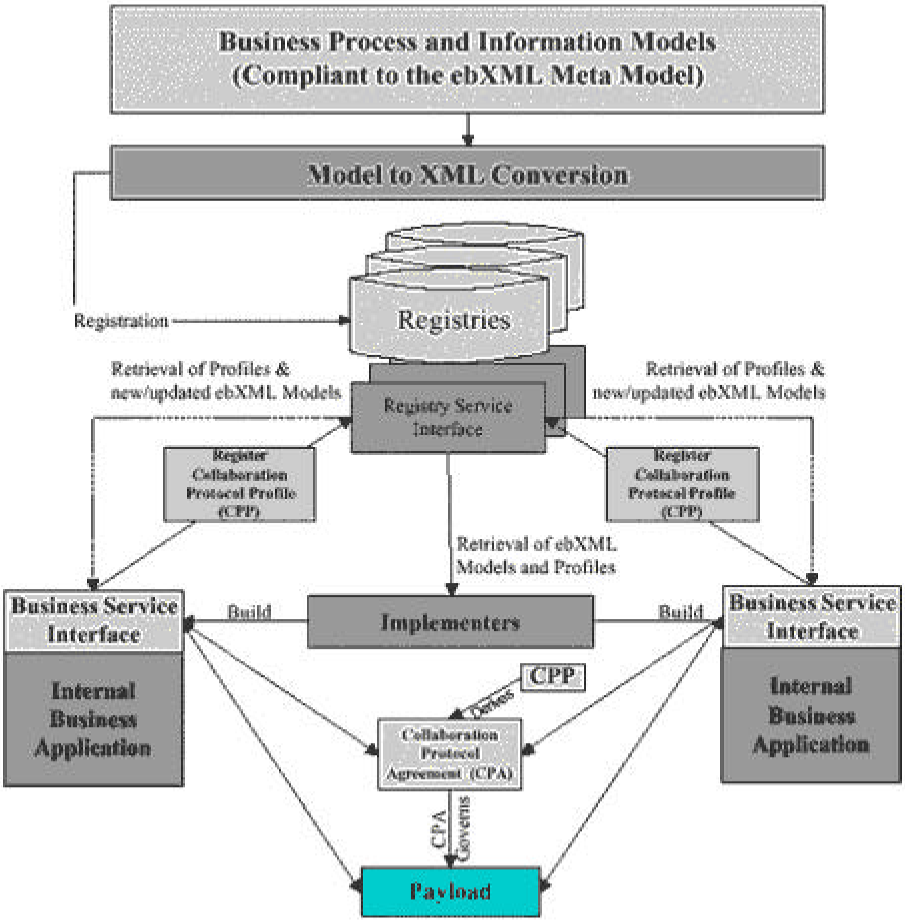

Business Process and Information Models
(Compliant to the ebXML Meta Model)
Model to XML Conversion
Registration
Registries
Retrieval of Profiles & new/updated ebXML Models
Retrieval of Profiles & new/updated ebXML Models
Registry Service Interface
Register Collaboration Protocol Profile (CPP)
Register Collaboration Protocol Profile (CPP)
Retrieval of ebXML Models and Profiles
Business Service Interface
Internal Business Application
Build
Implementers
Build
Business Service Interface
Internal Business Application
CPP
Defines
Collaboration Protocol Agreement (CPA)
CPA Governs
Payload

Figure 14: The Functional Service View

The shared information is maintained as objects in an ebXML Registry that is managed by ebXML Registry Services. Access to an ebXML Registry is provided by the interfaces (APIs) exposed by Registry Services. The Registry provides the access services interfacing, the information model and reference system implementation and the physical backend information store. For example, an ebXML Registry may provide a Collaboration Protocol Profile (CPP) in response to a query; or an ebXML Registry may contain reference DTDs or Schemas that are retrieved by the Registry as a result of searching a metadata classification of the DTDs or Schemas. Figure 15 provides an overview of this configuration.

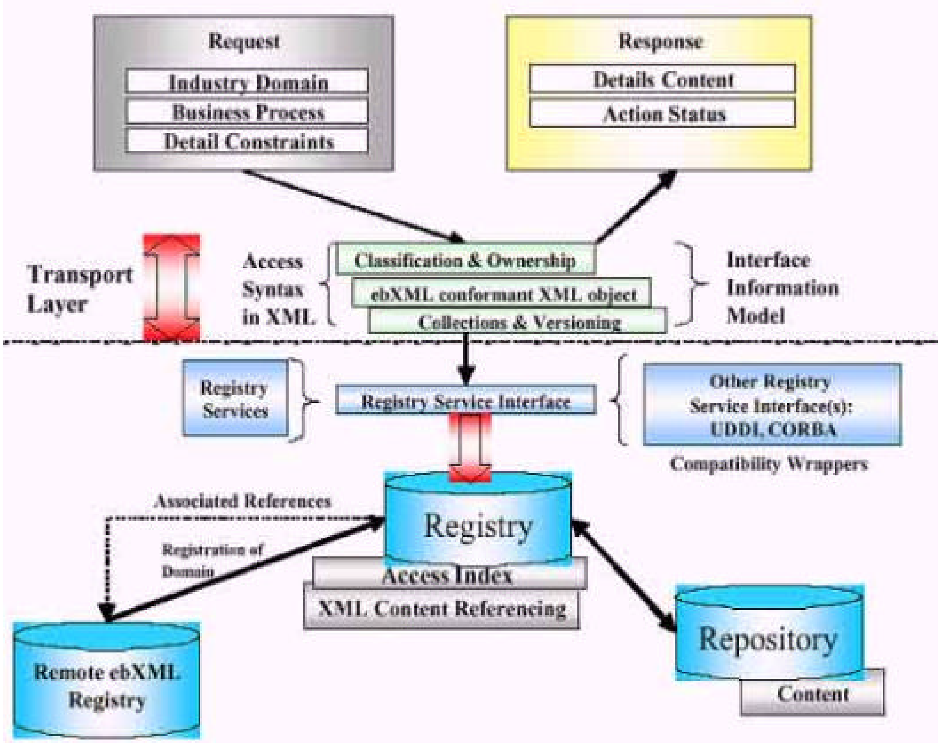


Figure 15: Registry Interaction Overview.

The ebXML, Transport Routing and Packaging (TR&P) is working on transport level for XML documents. Specification are likely to remain agnostic with respect to underlying wire protocol. Discussed protocols include HTTP, FTP, SMTP and SOAP. SOAP is the best candidate.

On Friday 11 May ebXML approved _ALL_ required Specifications, Technical Reports and White Papers resulting in proving the ebXML e-Business Frameworks as set out 18 months ago.

4.8.3.2 Tools

Currently available tools seems to provide framework in order to build ebXML compliant services.

IBM Web Services Development Environment (free from IBM alphaworks): SOAP + UDDI available at: http://www.alphaworks.ibm.com/tech/wsde

Component-X: http://www.enterprise-component.com/CxPosition.htm

Two software providers to ebXML demonstration :

- Cisco: http://www.cisco.com
- Interwoven: http://www.interwoven.com/

IONA/NETFISH (http://www.iona.com) announce support of the standard as soon as available.

Apache has announced that it will develop an ebXML reference implementation in JAVA code.

4.8.4 ebXML the best solution for REGNET

Using ebXML, companies now have a standard method to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes.

ebXML is a formal standard:

ebXML offers a framework that will become an international standard, most likely under the auspices of UN/CEFACT, one of the four de jure standards bodies in the world. Even before ebXML becomes a formal standard, it has already become a de facto standard as industry groups, individual trading partners and e-Business solution providers adopt it.

The Business need for ebXML integration:

The business need for integrating ebXML is the fact that until now, the technology available for most businesses to exchange data was electronic data interchange or EDI, which made significant contributions to productivity and inventory control. Many companies, however, find EDI expensive and difficult to implement. ebXML, using the economies of scale presented by the Internet, breaks through these obstacles. Therefore, it is easy to combine the current state-of-the-art with

The conventions established by ebXML are available publicly. These conventions encourage software developers to build packaged applications based on the common structure and syntax of ebXML messages and dramatically lower the cost of exchanging business data.

XML as technical foundation:

One of the technical foundations of ebXML is XML, that allows parties to exchange structured data, like the information kept in databases, over the Internet. XML is an open and freely available document from the World Wide Web Consortium and has the support of the world's leading technology companies. XML also supports Unicode that enables the display and exchange of most of the world's written languages.

ebXML supports B2B and B2C applications:

ebXML supports messages and services among businesses as well as between businesses and consumers. For business-to-consumer exchanges, however, the specifications define only the services and architecture on the business end, not customer screens or interactions.

ebXML address the needs of the small-medium size enterprises :

ebXML's requirements begin with the objective to promote the use of shrink-wrapped, plug-and-play software to support its messages. By keeping that focus paramount, as well as taking advantage of the economies of scale presented by the Internet, ebXML's design and technical architecture remain within the reach of smaller businesses.

ebXML and other XML initiatives:

Few if any other XML-based initiatives have tried to accomplish what ebXML does. Other e Business specifications address single industries or a specific set of business functions. Many of these initiatives now support ebXML and integrate the specifications into their own work.

RosettaNet, a consortium of more than 400 companies in information technology, electronic components and semiconductor manufacturing, plans to integrate support for the ebXML Messaging Services Specification in future releases of RosettaNet's Implementation Framework (RNIF). The Global Commerce Initiative, which represents manufacturers and retailers of consumer goods, chose to base their new Internet protocol standard for trading exchanges and B2B communications on ebXML.

Other industry organisations, such as the Automotive Industry Action Group, Health Level Seven, Open Applications Group, Open Travel Alliance, SWIFT and formal international and North American EDI standards bodies, have also been active participants in the ebXML initiative

4.8.4.1 The e-Business Advantage

ebXML offers several advantages in businesses of any kind or size:

Common Message Structure:

ebXML offers businesses of all sizes a common message structure and syntax for exchanging business data over data networks like the Internet using XML. Without ebXML, companies face the prospect of interacting with multiple vocabularies, most focusing on specific industries or functions that cannot talk to each other.

ebXML reduce costs

ebXML enables businesses to exchange XML-based messages and offer data services over networks with any other businesses. Companies that use EDI now, will likely find ebXML software much less expensive and easier to implement. For companies that use paper-based forms, the staff time saved through using business data exchange will be even greater.

ebXML enhances the competitive advantage

Companies that implement ebXML will find it easier to use networks for exchanging data with current and potential trading partners. They will be able to add new trading partners much more easily and open up new markets with less effort than before.

How will ebXML affect relationships with trading partners

Those suppliers and customers with whom companies now use EDI will likely see little change at first, since systems based on EDI will continue operating successfully. For those trading partners not using standards-based data exchanges, however, ebXML offers a chance to begin taking advantage of the improvements in business processes and productivity that these exchanges offer.

ebXML extends electronic business to new and existing trading partners

ebXML includes specifications for public repositories of industry business processes, messages, and common data objects that companies need to get started exchanging data, as well as to register their capabilities to engage in electronic business. Companies can use these registries to access the stored data objects and find new suppliers or customers with the ability to provide electronic messages or services.

For existing trading partners - for example, those using EDI - ebXML offers a way to increase the level of support or service while maintaining compatibility with your existing EDI investment.

4.8.4.2 Implementing ebXML

The ebXML impact on current EDI investments:

Companies with systems set up for business data exchange will probably have fewer changes in business processes than those starting from scratch. ebXML builds on the lessons learned from EDI, particularly the need to identify trading partners and messages and account for all message traffic. The best practices established for effective EDI apply to ebXML. ebXML also identifies common data objects, called core components, that allow companies to interchange standard EDI data with XML vocabularies compliant with the ebXML specifications.

ebXML facilitates convergence of different XML-based implementation frameworks:

The common message structure and syntax of ebXML encourages industries with XML vocabularies to adjust their efforts to meet ebXML requirements. Companies in these industries gain interoperability with other industries as a result of this effort. No business communicates solely within its supply chain. All companies need to exchange messages with those outside their industry boundaries as well as within them.

The ebXML affect on an existing IT infrastructure

If a company does not yet exchange electronic business data, ebXML means making the connections to send and receive these messages, authenticating other parties, editing the contents of the messages, and mapping the data to internal systems. If a company already uses EDI or other business data exchange protocols, it may have already established these facilities but may still need to write new routines for ebXML messages. We expect packaged software to make these functions transparent to the end-users, but they will still need to get done.

The development environments that ebXML supports

ebXML was designed to be independent of equipment, software platforms or communication networks. As long as a system supports standard Internet transport protocols and XML, it should also support ebXML.

EbXML is free of charge

UN/CEFACT and OASIS provide ebXML specifications free of charge. There are no royalties or fees associated with the use of the ebXML specifications. Openness of the ebXML specifications is a requirement in order to encourage adoption.

4.9 Multimedia Document Models

With respect to the publishing component, standards in the area multimedia document models are of importance for REGNET. They are the underlying models for authoring and presentation tools. Today, a variety of proprietary and standard *multimedia document models* for the specification of interactive multimedia content exist. They are employed to model the relationships between the media elements participating in a multimedia presentation. They must cope with basic requirements to multimedia documents, i.e., the modelling of the temporal and spatial course of a multimedia presentation and also the modelling of user interaction. To meet the requirements of REGNET the model should allow easy and wide access of users to cultural (multimedia) data and the production of personalised/customised cultural (multimedia) content. Therefore, the document model must provide access to its *structure* and allow the fine-grained *reuse* of parts of previously composed multimedia content. Additionally, the multimedia document model must provide the foundation *adaptation* of the content to allow and adaptation/personalisation of the documents *to the users system infrastructure and interest*.

In compliance with REGNET's aim to base content creation and management on actual standards the selected models are the ISO and W3C standards HTML, DHTML, HTML+TIME to HyTime, MHEG-5, SMIL 1.0, and SMIL 2.0. Open multimedia document standards have now achieved a mature state and offer powerful means for the standard conformant specification of multimedia presentations. To provide a suitable basis to decide on a suitable multimedia document model, in the following we present standards along the features that mainly characterise such model with regard to REGNET's general requirements. To meet REGNET's requirements, a multimedia document model should model the temporal course, the spatial arrangement, the interaction capabilities, and adaptation capabilities. With QuickTime, ShockWave we find special (multi) media formats that are proprietary. However, as these are quite often employed in the context of multimedia and form a kind of de-facto standard, these models are briefly introduced as well.

4.9.1 HTML family

4.9.1.1 Introduction into the HTML family with regard to multimedia features

With HTML it is possible to include various kinds of objects like media elements (e.g., images, videos and audio tracks), JAVA applets, ActiveX components, and scripts. In addition to that, HTML allows for the definition of *hyperlinks* between documents to define interactions. Scripts, Applets, and ActiveX components included with a document are executed at presentation time by the HTML browser software, e.g., the Netscape Communicator . However, the HTML standard does neither define syntax or semantics of the scripting languages, so presentation behaviour of an HTML page that includes scripts depends on the employed browser software.

There were efforts of the large HTML browser software vendors Netscape and Microsoft to allow for the manipulation of the structure, layout, and content of an HTML document with scripting languages. Under the notion of **Dynamic HTML** (DHTML), in 1997, Netscape (http://developer.netscape.com/tech/dynhtml/index.html) and Microsoft (http://msdn.microsoft.com/workshop/author/dhtml/dhtmlovw.asp) developed different, non-compatible HTML object models and scripting languages that allow to dynamically manipulate HTML documents. With the Document Object Model (DOM) these developments were brought together in the W3C recommendation of a standardised object oriented programming model for HTML and XML documents. With DOM a standardised internal representation offers a platform and language independent interface for programs and scripts for the dynamic manipulation of content structure and style of XML/HTML documents. Very often one finds DOM employed for dynamic manipulation of HTML pages with the scripting language JavaScript. JavaScript is a scripting language developed by Netscape (http://www.netscape.com) that allows Web developers to create dynamic, interactive pages that respond to user input. In this response, JavaScript can produce pop-up windows, display date and time information, perform form verification and calculations. With JavaScript some multimedia functionality can basically be implemented using events and timeout, however, it does not at all contribute to the HTML document model.

With **HTML+TIME** (http://www.w3.org/TR/1998/NOTE-HTMLplusTIME-19980918), we find a Microsoft (http://www.microsoft.com) specific extension of HTML. HTML+TIME extends HTML by own tags for the integration of audio and video into the HTML page as well as for temporal synchronisation of media elements. HTML+TIME enhanced HTML Pages can be presented from the Internet Explorer Version 5.5 on. The temporal model of HTML+TIME provides a combination of time line based and event based temporal model. With a specific tag, the model supports limited adaptation of the document to a very limited number of context attributes like the system bit-rate, and the system language. With a specific tag, the model supports limited adaptation of the document to a very limited number of context attributes like the system bit-rate, and the system language. The SMIL 2.0 standard (cf. Section 0) that is currently in its final negotiation phase defines so-called language binding. For example in http://www.w3.org/TR/2000/WD-smil-boston-20000622/html-smil-profile.html an XHTML+SMIL language binding has been proposed. With support of the HTML+SMIL Language Profile in SMIL 2.0 in their browser Microsoft wants to bridge the gap between HTML and SMIL. Microsoft claims to support the SMIL modules Timing and Synchronisation, Animation module, Media Object and the Integration module with its Internet Explorer 5.5.

With the document models HTML, DHTML, and HTML+TIME we presented so far different variations of HTML based (multimedia) document models. With the following standards we leave the notion of a standard document model towards *embedding specific proprietary formats and applications in HTML* that provide more or less multimedia functionality. However, as they are often mentioned in the context of HTML and multimedia and for the sake of completeness we shortly introduce these and their features with regard to multimedia capabilities.

With JAVA (http://java.sun.com) *applets* platform independent programs can be embedded into an HTML page and run by the Web browser's JAVA runtime environment. With the JAVA Media Framework API (http://java.sun.com/products/java-media/jmf/index.html) arbitrary multimedia applications can be programmed and embedded in the HTML page, if the respective APIs are installed on the local platform. However, the multimedia application is not part of the HTML document and has nothing to do with a multimedia document model. The same applies for *Shockwave* files that carry interactive multimedia animations and interactive movies in the *QuickTime* format. These files can, like JAVA applets, be embedded in an HTML page and are presented with the respective Shockwave and QuickTime plugin in an Internet browser.

4.9.1.2 Relevance for REGNET

The cultural (multimedia) content is potentially of different media types and entering, retrieving and publishing this content will address also *multi*media content. Multimedia compositions, however, can not be sufficiently represented with HTML, as not even temporal synchronisation is possible with this model. Consequently, the publishing model of REGNET should not be based on pure HTML. The aim of REGNET to be compliant with standards also contradicts to the usage of HTML with DHTML as then the multimedia functionality is purely programmed. However, with HTML+TIME, especially with the HTML+SMIL Language Binding, supported by Microsoft's Internet Explorer 5.5, we find a document format that allows describing synchronised multimedia presentations. The temporal and spatial course of a presentation can be modelled with HTML+TIME as well as the adaptation to a limited set of system parameters. Reuse of previously authored parts is not supported as the models do not provide modelling primitives for explicitly reusing and recomposing existing parts - it is still just copy and paste. With the SMIL 2.0 standardisation it can be expected that HTML+TIME will be compliant to the SMIL language profile specification, however, will not provide support for the content control module that defines comprehensive adaptability.

The table below gives an overview of the HTML related models (+ = support o = partial support - = no support):

	Temporal Synchronisation	Spatial Modelling	Interaction	Reuse	Adaptation
HTML	–	o	+	–	–
DHTML	–	o	+	–	–
HTML+TIME	+	o	+	–	o

Table 5: HTML related models

For HTML, DHTML and HTML+TIME presentation engines by means of Web browsers exist. Authoring tools for HTML are widely available. Authoring support for DHTML is given only with regard to the HTML part. HTML+TIME is supported only by the Internet Explorer from version 5.5. However, for HTML+TIME so far no authoring support is given that goes beyond the capabilities of HTML editing tools. Authoring the multimedia part of HTML+TIME document is not supported.

The following table illustrates the authoring and presentation support (+ = support; o = partial support; - = no support):

	HTML	DHTML	HTML +TIME
Authoring Tool support	+	o	o
Tool support Presentation	+	+	+ Internet Explorer 5.5

Table 6: Authoring and presentation support

4.9.2 MHEG-5

4.9.2.1 Introduction into the MHEG-5

MHEG-5: MHEG-5 (ISO/IEC 13522-5) is the fifth part of MHEG (http://www.km.giti.waseda.ac.jp/WG12/). This part defines the MHEG object classes for interchange and use in base-level applications intended to be run on limited resource terminals such as set-top-boxes in such contexts as interactive broadband services. MHEG-5 is an adaptation of the MHEG-1 Standard (ISO/IEC IS 13522-1) to the needs of video-on-demand and kiosk applications for minimal resource systems such as set-top-boxes and low-end PCs. MHEG-5 encodes applications in their final form and aims at an efficient realisation of MHEG-1 attracting the interest of telecommunication and Entertainment industry in this standard.

MHEG-5 provides an object-oriented data model for multimedia documents. The standard defines a hierarchy of *MHEG-5 classes*. This hierarchy comprises classes for various uses. For example, there are classes that represent media elements like videos and audios, classes that represent interaction elements like buttons, and even classes that provide variable functionality of programming languages. Classes possess attributes, can perform actions (which closely resemble methods in object-oriented programming languages), and fire events. An MHEG-5 document is a composition of instances of these classes organised in *scenes* which are the main structural primitives. A scene corresponds to a ``page'' on a screen and, hence, only one scene can be presented at a time. In addition to that, each MHEG-5 document features one instance of the class `Application` defining the entry point for document's presentation. Moreover, this application object can contain objects which are global to every scene. The presentation behaviour of an MHEG-5 document is defined by the means of `links` which resemble event-condition-action rules. With these an event based temporal model and user interaction modelling is realised. Interaction is specified by a set of interaction elements - slider, text input feld, hotspots, and buttons. Media objects are included in the document by `content` objects, that contain or refer to the media data. The representation of media data uses other standards like JPEG and MPEG. `Procedure` objects are used to interface to platform specific devices. They can also be used to implement a remote procedure call to the server system.

There are several areas in which study and discussion is taking place with regard to extensions for MHEG-5:

1. Support for 3D objects: during the Seoul meeting a new project AMD1 "Additional Class for MHEG-5 was submitted to SC 29 and approved;
2. Object and parameter passing (two-way) between MHEG-5 and other external entities, including web technologies, databases, etc.;
3. An additional connection class to support enhanced functionality for controlling and managing connections. The DSMCC does not now support the required functionality;

4. Convergence issues for MHEG-5 and WWW technologies.

MHEG-6 (ISO/IEC IS 13522-6) is an extension of MHEG-5 that introduces an interface between an MHEG-5 engine and a JAVA Virtual Machine (http://citeseer.nj.nec.com/hofmann96mheg.html). With MHEG-6, it is possible to include JAVA Bytecode into an MHEG-5 document and have this code executed from within the MHEG-5 application. MHEG-6 defines an architecture consisting of an MHEG-5 engine and an MHEG-6 virtual machine using SUN's JAVA technology. Interfaces between the MHEG-Engine and the JAVA Virtual Machine are a JAVA Class Mapper, an MHEG-5-API and a Procedure Call Interface. The JAVA classes to be executed at the presentation device are transported via the MHEG-5 `Procedure` objects. With the Procedure Call Interface the MHEG-5 application can call JAVA Methods and the MHEG-5 API allows JAVA methods to call actions on MHEG-5 objects. With MHEG-6 interactivity is gained for the loss of declarative document structure in favour of programmed manipulation of MHEG-5 documents.

MHEG-7 (ISO/IEC 13522-7) defines a test suite that can be used to test an MHEG-5 engine's interoperability and conformance to a specific application domain. It also defines a format for test cases that can be used to extend the test suite, either for more detailed testing or for extensions defined by the application domain.

MHEG-8 will provide XML encodings for MHEG-5 and was approved in March 1999. At present the XML DTD is being developed and this work has been progressed to the Committee Draft level during the Vancouver meetings.

4.9.2.2 Relevance for REGNET

Although MHEG-5 offers high multimedia functionality, it mainly describes the presentation and not the structure of a multimedia document and, therefore, cannot be employed for presentation independent modelling of multimedia document content for the easy and wide (re) use of the content in REGNET. The supported reuse of existing document fragments is severely hampered due to the inflexible scene-based document structure.

The table below gives an overview of multimedia capabilities of MHEG-5 (+ = support; o = partial support; - = no support):

	Temporal Synchronisation	Spatial Modelling	Interaction	Reuse	Adaptation
MHEG-5	+	+	+	–	–

Table 7: MHEG-5 multimedia capabilities

There are only a few tools available for MHEG-5 such that authoring and presentation is only rarely supported as illustrated in the following table 8 (+ = support; o = partial support; - = no support):

	MHEG-5
Authoring Tool support	o
Tool support Presentation	o

Table 8: MHEG-5 authoring and tool support

4.9.3 SMIL

4.9.3.1 Introduction into SMIL

SMIL 1.0: The Synchronised Multimedia Integration Language (SMIL, http://www.w3.org/TR/1998/REC-smil-19980615) aims at synchronised multimedia presentations on the web. It was developed by the SYMM (Synchronised Multimedia) Working Group of the W3C (http://www.w3c.org/AudioVideo) and its first Version 1.0 is a recommendation of the W3C since 1998. SMIL is defined by an XML DTD (http://www.w3.org/TR/1998/REC-xml-19980210) and, hence, the language can be understood as a set of element definitions specified in terms of XML. A SMIL document provides synchronisation of continuous media elements, spatial layout and specifies interaction and adaptation. SMIL defines the synchronisation elements *par* and *seq* to describe temporal synchronisation between media elements. Furthermore, the spatial layout of the whole presentation and that of single media elements can be defined. SMIL also allows to specify links between documents or parts of documents which are equivalent to HTML links. An interesting feature of SMIL is the *switch* element which is a simple means for modelling alternatives in the course and quality of a presentation. With the help of switch elements, an author can specify different presentation alternatives among which, one is chosen at presentation time due to external parameters. For the SMIL 1.0 standard there are already many different free players available like GRiNS (http://www.oratrix.com/GriNS), SOJA (http://www.helio.org/products) Schmunzel (http://www.salzburgresearch.at/suntrec/schmunzel/) and (http://www.real.com/products/player) - RealPlayer G2

SMIL 2.0: After the completion of the SMIL 1.0 version it was clear that an advanced and extended version was needed -- hence SMIL was subject to ongoing research. The steps of advancement to SMIL 1.0 have been aggregated in the SMIL Boston activities. The results of these efforts are now on their way to become the SMIL 2.0 standard: in September 2000 the last call for the Public Working Draft of SMIL 2.0 has been published by the W3C and the latest version is of June 2001 (http://www.w3.org/TR/2001/PR-smil20-20010605). Compared to the SMIL 1.0 in SMIL 2.0 existing features have been extended and refined and additional features have been added to the standard: With regard to the organisation of the standard a new aspect is the modularisation into sets of semantically-related elements. Special emphasis has been put into the extension and refinement of the timing and synchronisation. The standard extends the temporal model --- the timeline based model (temporal aggregation) is extended by an event-based timing which allows to specify a temporal course that is dependent on user interactions and other system events and not only follow a strict timeline. With regard to adaptation the content control module provides additional features for the specification of presentation alternatives. So called `customAttributes` elements allow to specify user defined test attributes for selecting the presented content. Further extension concern the support of animation, spline animation, alignment of visual media, pre-fetch control of media elements, smooth transitions between media elements are included in the standard, modelling of metadata. The SMIL 2.0 standard defines a so-called HTML language profile (http://www.w3.org/TR/2000/WD-smil-boston-20000622/html-smil-profile.html). With support of the HTML+SMIL Language Profile in SMIL 2.0 in their browser Microsoft wants to bridge the gap between their proprietary HTML+TIME format and SMIL. Microsoft claims to support the SMIL modules Timing and Synchronisation, Animation module, Media Object and the Integration module with its Internet Explorer 5.5.

4.9.3.2 Relevance for REGNET

SMIL is going to be *the* standard for multimedia on the WWW. From its multimedia capabilities as presented it can be favoured as can be seen from the following table (+ = support; o = partial support; - = no support):

	Temporal Synchronisation	Spatial Modelling	Interaction	Reuse	Adaptation
SMIL 1.0	+	+	+	–	o
SMIL 2.0	+	+	+	–	+

Table 9: SMIL multimedia capabilities

Tool support for authoring and presentation is given with SMIL 1.0. And also for the upcoming SMIL 2.0 standard tools will be soon available that support the main features of the new standard. (+ = support; o = partial support; - = no support):

	SMIL 1.0	SMIL 2.0
Authoring Tool support	+	+
Tool support Presentation	+ free players	+ free players

Table 10: SMIL authoring and tool support

4.9.4 HyTime

HyTime (Hypermedia/Time-based Structuring Language, http://xml.coverpages.org/hytime.html) is a standard (ISO 10744:1997, http://www.ornl.gov/sgml/wg8/docs/n1920/html/n1920.html) which allows for the description of the structure of multimedia documents. HyTime provides a well-defined set of primitives which allows for the *addressing* of portions of media data, the *interlinking* of media objects, and the *alignment* that places portions of media in co-ordinate systems which can represent space, time, or any quantifiable dimension.

The primitives provided by HyTime are offered by means of *architectural forms* and are organised in terms of *modules*. Architectural Forms (AF) are HyTime elements with pre-defined multimedia semantics and attributes. An AF can be used in any SGML DTD by extending an SGML element type by an attribute *HyTime* bearing the name of the AF to be used. In that way, the element type inherits the semantics and attributes of the AF. In compliance with the SGML philosophy HyTime excludes presentation format modelling.

The modules of HyTime are the Base-Module, which defines the basic concepts of HyTime, the Location-Address-Module, which implements the powerful construct of *Locators* providing an abstract mechanism for addressing external document objects, the Hyperlink-Module, which implements the concept of links, the Finite-Co-ordinate-Space-Module, which provides means for the synchronised presentation of media objects based on n-dimensional *co-ordinate spaces*, the Event-Projection-Module, which allows to transform event schedules defining the temporal execution of a presentation, and the Object-Modification-Module, which allows to transform presentation objects, e.g., fading.

Though HyTime is a very sophisticated model, it does not provide a model for interaction and, in accordance with the SGML philosophy, includes no presentation format modelling. HyTime is not wide spread, as, among others, there are no commercial tools available for authoring and presentation. The only published presentation environment was the research prototype HyOctane.

4.9.4.1 Relevance for REGNET

Powerful support for reuse is the strength of HyTime. Moreover, HyTime describes document content at a very high semantically level and, thus, is perfectly suited for presentation-neutral modelling of document content and hence the reuse in different user environments. However, the lacking capability of interaction modelling and modelling of adaptation make it irrelevant to REGNET.

HyTime features with respect to REGNET are summarised in the table below (+ = support; o = partial support; - = no support):

	Temporal Synchronisation	Spatial Modelling	Interaction	Reuse	Adaptation
HyTime	+	+	–	+	–

Table 11: HyTime features

The even stronger drawback is the complete lacking of both authoring and presentation tools as illustrated in the table below (+ = support; o = partial support; - = no support):

	HyTime
Authoring Tool support	–
Tool support Presentation	–

Table 12: HyTime authoring and tool support

4.9.5 Commercial Approaches

4.9.5.1 QuickTime

QuickTime (http://developer.apple.com/quicktime/) originates from Apple Computer Inc. (http://www.apple.com), and offers a proprietary format with its corresponding tools and APIs for creating and presenting multimedia content. The underlying QuickTime Movie File (QMF) is a published file format for storing multimedia content for QuickTime presentation. QMF uses a track model for organising the temporally related data of a movie. A QuickTime movie can contain one or more tracks. A track is a time ordered sequence of a media type; the media addressed using an edit list, that is a list of the endpoints of digital media clips and segments. With the respective server software, media elements in QuickTime movies can be streamed to the client. A QuickTime movie also allows to specify different quality versions; so-called reference movies contain pointers to the alternate data rate movies. QuickTime Players are available for most platforms, Plug-ins for QuickTime are available for most browsers.

4.9.5.2 Shockwave

Shockwave (http://www.macromedia.com/shockwave/) is a Macromedia Inc. file format (http://www.macromedia.com) for multimedia animations produced with their tool Flash. These files can be presented for example with the Shockwave Plugin in an Internet browser. The animations can contain comprehensive graphics, spatial and temporal arrangement of objects, and can include audios and videos. Macromedia announced an export to SMIL from Flash version 5.0 on.

4.9.5.3 Relevance for REGNET

Though both with QuickTime and with Shockwave sufficient multimedia capabilities are provided they form de-facto standards which are not in line with REGNET's aim for supporting standards and the related tools. With Shockwave one is limited to Macromedia's authoring tools and the result are "programmed" non-reusable multimedia presentations. Let aside that adaptation is not or only partly supported.

4.9.6 Comparison of relevant multimedia document models for REGNET

The decision for a multimedia document standard model on the authoring and presentation level allows to use and integrate existing tools that are and will be available on the market for these standards into the system. In the following, we shortly recap the document models, HTML, DHTML, HTML+TIME, SMIL and MHEG with regard to the tool support and the respective consequences for the REGNET in case of a decision for each of this model.

Summarising we can say that "pure" HTML can hardly be characterised as a multimedia document model because it lacks support for even the most basic multimedia requirement, a temporal model. Though HTML can become a quite powerful multimedia document model by the extension to DHTML, it still lacks support for reuse at all levels of granularity and suffers from a low semantic level of content description, which leaves DHTML unsuitable for presentation-neutral description of multimedia content. HTML+TIME, however, offers a solution which provides suitable multimedia support with currently available presentation environment. The future multimedia "language" of the Web will be SMIL and with SMIL 2.0 a quite mature is going to be available soon. MHEG-5 describes the multimedia composition only on a very low semantic level and there are not many authoring environments around as well as players. HyTime, being a multimedia standard available for a long time does not offer interaction, lacks authoring and presentation environments and has not achieved wide acceptance.

The following table summarises the different standard models presented so far (+ = support; o = partial support; - = no support):

	Temporal Synchronisation	Spatial Modelling	Interaction	Reuse	Adaptation
HTML	–	o	+	–	–
DHTML	–	o	+	–	–
HTML+TIME	+	o	+	–	o
SMIL 1.0	+	+	+	–	o
SMIL 2.0	+	+	+	–	+
MHEG-5	+	+	+	–	–
HyTime	+	+	–	+	–

Table 13: Summary of multimedia document models

Along with the document model a set of commercial tools are available on the market for authoring the documents, a fact which should not be unconsidered when deciding for a document model for authoring and presentation. Though the syntax of HTML+TIME is not difficult to understand tool support for authoring is lacking. HTML+TIME can be presented by the available Internet browser IE 5.5. Many presentation tools for presenting SMIL documents are available so far. The Real Player already combines their streaming approach with the SMIL (2.0) capabilities. Also, many authoring tools are available; our non-empirical trials have shown their usage can still be tedious. Even though MHEG-5 is intended for small resource systems such as set-top-boxes the lack of widely available authoring and presentation environments. The following summarises the different models with regard to their tool support (+ = support; o = partial support; - = no support):

	HTML	DHTML	HTML +TIME	SMIL 1.0	SMIL 2.0	MHEG-5	HyTime
Authoring Tool support	+	o	o	+	+	o	–
Tool support Presentation	+	+	+ Internet Explorer 5.5	+ free players	+ free players	o	–

Table 14: Summary of authoring and tool support

Hence, HTML+TIME and SMIL 2.0 should be the models of choice. With HTML+TIME one restrict oneself to an HTML based solution and the Internet Explorer as presentation environment. However, if the intended authors are rather used to create HTML based presentations this solution could be more convenient. SMIL 2.0 is expected to be a standard soon and comprises a wide set of multimedia capabilities. Along with this, authoring tools being available that support major parts of the standard and they will provide authors with the necessary authoring support.

4.9.7 MPEG-7

4.9.7.1 Introduction

MPEG-7 (http://www.cselt.it/mpeg/standards/mpeg-7/mpeg-7.htm) is an ISO/IEC standard (ISO/IEC WD 15938-1) developed by MPEG (Moving Picture Experts Group, http://www.cselt.it/mpeg/), the committee that also developed the Emmy Award winning standards known as MPEG-1 and MPEG-2, and the 1999 MPEG-4 standard. MPEG-1 and MPEG-2 standards made interactive video on CD-ROM and Digital Television possible. MPEG-4 provides the standardised technological elements enabling the integration of the production, distribution and content access paradigms of the fields of digital television, interactive graphics and interactive multimedia.

MPEG-7, formally named “Multimedia Content Description Interface”, aims at the creation of a flexible exchange format for the description of media content on both a semantic level (e.g., the persons depicted on a photograph) and a technical level (e.g., the colour distribution of an image). MPEG-7 is not aimed at any one application in particular; rather, the elements that MPEG-7 standardises shall support as broad a range of applications as possible. The following details on MPEG-7 are based on the *original* ISO/IEC (ISO/IEC JTC 1/SC 29/WG 11) documents for MPEG-7.

Audio-visual data content that has MPEG-7 data associated with it, may include: still pictures, graphics, 3D models, audio, speech, video, and composition information about how these elements are combined in a multimedia presentation (scenarios). Special cases of these general data types may include facial expressions and personal characteristics.

MPEG-7 description tools do, however, not depend on the ways the described content is coded or stored. It is possible to create an MPEG-7 description of an analogue movie or of a picture that is printed on paper, in the same way as of digitised content.

MPEG-7, like the other members of the MPEG family, is a standard representation of audio-visual information satisfying particular requirements. The MPEG-7 standard builds on other (standard) representations such as analogue, PCM, MPEG-1, -2 and -4. One functionality of the MPEG-7 standard is to provide references to suitable portions of them. For example, perhaps a shape descriptor used in MPEG-4 is useful in an MPEG-7 context as well, and the same may apply to motion vector fields used in MPEG-1 and MPEG-2.

Even though the MPEG-7 description does not depend on the (coded) representation of the material, the standard in a way builds on MPEG-4, which provides the means to encode audio-visual material as objects having certain relations in time (synchronisation) and space (on the screen for video, or in the room for audio). If the material is encoded using MPEG-4, it will be possible to attach descriptions to elements (objects) *within* the scene, such as audio and visual objects. MPEG-7 will allow different granularity in its descriptions, offering the possibility to have different levels of discrimination.

Because the descriptive features must be meaningful in the context of the application, they will be different for different user domains and different applications. This implies that the same material can be described using different types of features, tuned to the area of application. To take the example of visual material: a lower abstraction level would be a description of e.g. shape, size, texture, colour, movement (trajectory) and position (‘where in the scene can the object be found?). And for audio: key, mood, tempo, tempo changes, position in sound space. The highest level would give semantic information: ‘This is a scene with a barking brown dog on the left and a blue ball that falls down on the right, with the sound of passing cars in the background.’ Intermediate levels of abstraction may also exist.

The level of abstraction is related to the way the features can be extracted: many low-level features can be extracted in fully automatic ways, whereas high level features need (much) more human interaction.

Next to having a description of the content, it is also required to include other types of information about the multimedia data:

- *The form* - An example of the form is the coding scheme used (e.g. JPEG, MPEG-2), or the overall data size. This information helps determining whether the material can be ‘read’ by the user.
- *Conditions for accessing the material* - This includes links to a registry with intellectual property rights information, and price;
- *Classification* - This includes parental rating, and content classification into a number of pre-defined categories;
- *Links to other relevant material* - The information may help the user speeding up the search.
- *The context* - In the case of recorded non-fiction content, it is very important to know the occasion of the recording (e.g. Olympic Games 1996, final of 200 meter hurdles, men)

In many cases, it will be desirable to use textual information for the descriptions. Care will be taken, however, that the usefulness of the descriptions is as independent from the language area as possible. A very clear example where text comes in handy is in giving names of authors, film, places.

Therefore, MPEG-7 Description tools will allow to create descriptions (i.e., a set of instantiated Description Schemes and their corresponding Descriptors at the users will) of content that may include:

- Information describing the creation and production processes of the content (director, title, short feature movie)
- Information related to the usage of the content (copyright pointers, usage history, broadcast schedule)
- Information of the storage features of the content (storage format, encoding)
- Structural information on spatial, temporal or spatio-temporal components of the content (scene cuts, segmentation in regions, region motion tracking)
- Information about low level features in the content (colours, textures, sound timbres, melody description)
- Conceptual information of the reality captured by the content (objects and events, interactions among objects)

All these descriptions are of course coded in an efficient way for searching, filtering, etc.

To accommodate this variety of complementary content descriptions, MPEG-7 approaches the description of content from several viewpoints. Currently five viewpoints are defined: Creation & Production, Media, Usage, Structural aspects and Conceptual aspects. The five sets of description elements developed on those viewpoints are presented here as separate entities. However, they are interrelated and can be combined in many ways. Depending on the application, some will present and others can be absent or only partly present.

A description generated using MPEG-7 description tools will be associated with the content itself, to allow fast and efficient searching for, and filtering of material that is of interest to the user.

MPEG-7 data may be physically located with the associated AV material, in the same data stream or on the same storage system, but the descriptions could also live somewhere else on the globe. When the content and its descriptions are not co-located, mechanisms that link AV material and their MPEG-7 descriptions are needed; these links will have to work in both directions.

The type of content and the query do not have to be the same; for example, visual material may be queried using visual content, music, speech, etc. It is the responsibility of the search engine and filter agent to match the query data to the MPEG-7 description.

MPEG-7 addresses many different applications in many different environments, which means that it needs to provide a flexible and extensible framework for describing audio-visual data. Therefore, MPEG-7 does not define a monolithic system for content description but rather a set of methods and tools for the different viewpoints of the description of audio-visual content. Having this in mind, MPEG-7 is designed to take into account all the viewpoints under consideration by other leading standards such as, among others, SMPTE Metadata Dictionary, Dublin Core, EBU P/Meta, and TV Anytime. These standardisation activities are focused to more specific applications or application domains, whilst MPEG-7 tries to be as generic as possible. MPEG-7 uses also XML Schema as the language of choice for the textual representation of content description and for allowing extensibility of description tools. Considering the popularity of XML, usage of it will facilitate interoperability in the future.

Basically, MPEG-7 comprises two parts: the definition of a *Description Definition Language (DDL,* http://archive.dstc.edu.au/mpeg7-ddl/) that can be used to define application-specific media description schemes, and a set of standard description schemes for common application domains (http://www.darmstadt.gmd.de/mobile/MPEG7/Documents.html). The main elements of the MPEG-7's standard are:

- Descriptors (D): representations of Features, that define the syntax and the semantics of each feature representation,
- Description Schemes (DS), that specify the structure and semantics of the relationships between their components. These components may be both Descriptors and Description Schemes,

- A Description Definition Language (DDL) to allow the creation of new Description Schemes and, possibly, Descriptors and to allows the extension and modification of existing Description Schemes,
- System tools, to support multiplexing of descriptions, synchronisation of descriptions with content, transmission mechanisms, coded representations (both textual and binary formats) for efficient storage and transmission, management and protection of intellectual property in MPEG-7 descriptions, etc.

4.9.7.2 MPEG-7 parts

The MPEG-7 Standard consists of the following 7 parts:

Part 1 MPEG-7 Systems (http://www.darmstadt.gmd.de/mobile/MPEG7/Documents/w4001.htm) - the tools that are needed to prepare MPEG-7 Descriptions for efficient transport and storage, and to allow synchronisation between content en descriptions. Tools related to managing and protecting intellectual property. It defines the terminal architecture and the normative interfaces.

Part 2 MPEG-7 Description Definition Language (DDL, http://www.darmstadt.gmd.de/mobile/MPEG7/Documents/w4002.htm) - the language for defining new Description Schemes and perhaps eventually also for new Descriptors.

XML Schema Language has been selected to provide the basis for the DDL. As a consequence of this decision, the DDL can be broken down into the following logical normative components:

- The XML Schema structural language components;
- The XML Schema datatype language components;
- The MPEG-7 specific extensions.

Part 3 MPEG-7 Visual (http://www.darmstadt.gmd.de/mobile/MPEG7/Documents/w4062.htm) – the Descriptors and Description Schemes dealing with (only) Visual descriptions

MPEG-7 Visual description tools consist of basic structures and descriptors that cover following basic visual features: Colour, Texture, Shape, Motion, Localisation, and Others. Each category consists of elementary and sophisticated descriptors.

Part 4 MPEG-7 Audio (http://www.darmstadt.gmd.de/mobile/MPEG7/Documents/w4004.htm) – the Descriptors and Description Schemes dealing with (only) Audio descriptions

MPEG-7 Audio Committee Draft comprises six technologies: the audio description framework (which includes the scale tree and low-level descriptors), sound effect description tools, instrumental timbre description tools, spoken content description, the uniform silence segment, and melodic descriptors to facilitate query-by-humming.

Part 5 MPEG-7 Multimedia Description Schemes (http://www.darmstadt.gmd.de/mobile/MPEG7/Documents/w3966.htm) - the Descriptors and Description Schemes dealing with generic features and multimedia entities.

Generic entities are features, which are used in audio, visual, and text descriptions, and therefore “generic” to all media. These are, for instance, “vector”, “histogram”, “time”, etc.

Apart from this set of generic description tools, more complex description tools are standardised. They are used whenever more than one medium needs to be described (e.g. audio and video.) These description tools can be grouped into 5 different classes according to their functionality:

a) Content description: representation of perceivable information

b) Content management: information about the media features, the creation and the usage of the AV content;

c) Content organisation: representation the analysis and classification of several AV contents;

d) Navigation and access: specification of summaries and variations of the AV content;

e) User interaction: description of user preferences pertaining to the consumption of the multimedia material.

Part 6 MPEG-7 Reference Software (http://www.darmstadt.gmd.de/mobile/MPEG7/Documents/w4006.htm) - a software implementation of relevant parts of the MPEG-7 Standard

The eXperimentation Model (XM) software is the simulation platform for the MPEG-7 descriptors (Ds), description schemes (DSs), coding schemes (CSs), and description definition language (DDL). The XM applications are divided in two types: the server applications and the client applications.

Part 7 MPEG-7 Conformance - guidelines and procedures for testing conformance of MPEG-7 implementations.

Part 2 Development of System Specification

5 Overview and document structure of part 2

5.1 System Specification structure

The structure of that section is the outflow of the two-tracks process described below which consists mainly of the functional and technical analysis of the system and the architectures derived from this analysis:

Functional analysis is based on the functional requirements. These requirements are modelled using UML through Use Case representation. The REGNET functional architecture is then defined and shows links between modules according to the systems' functionalities.

The *technical architecture* is based on technical requirements. This architecture deals with the choice of technologies according to functionalities and technical requirements. It details the definition of hardware, network and software tools. An iterative software project plan, based on risks analysis is proposed.

The REGNET specification and solutions are split into two parts: *Requirement artefacts* which capture and present information used in defining the required capabilities of the system, and *Analysis & Design artefacts* which capture and present information related to the solution to the problems defined by the Requirements artefacts. *User requirements* (both functional and technical) reflect as much as possible the real needs so the functional requirements are based on solid ground.

According to this approach the REGNET System Specification is split as follows:

- The first part describes the REGNET building blocks.
- The second part contains both functional and technical requirements that drive the architectural elaboration.
- The third part contains the REGNET Actors and Use cases.
- The fourth part contains the REGNET Functional Architecture describing the System organisation according to the identified functionalities.
- The fifth part contains the REGNET Technical Architecture which describes the REGNET organisation according to the technical requirements.
- The sixth part contains the REGNET System Architecture from the viewpoint of tool support.
- The seventh part contains the REGNET Deployment possibilities.
- The last part is dedicated to technical interfaces between REGNET components that will guide the system development.

5.2 REGNET Building Blocks

The REGNET system is an Internet based collaborative network of Cultural Heritage Organisations and Institutions. The organisations / institutions are organised in a geographic way: each one is managed by a cultural service centre (CSC) which supports the local Cultural Heritage Institutions / Organisations (CIO). Each CIO provides different sets of functionalities according to specificities. All REGNET entities share common knowledge which is called Ontology. This knowledge deals with user profiles, data formats and presentation of information.

REGNET functionality has been split up into subsystems. The aim of this part is to describe main functionalities of these subsystems. Detail functionalities are defined in the part dedicated to functional requirements, links between subsystems are detailed in the part dedicated to architecture and interfaces.

The REGNET System consists of different building blocks (called ***nodes***) which can be located on different Hardware/Software-Platforms. These nodes are:

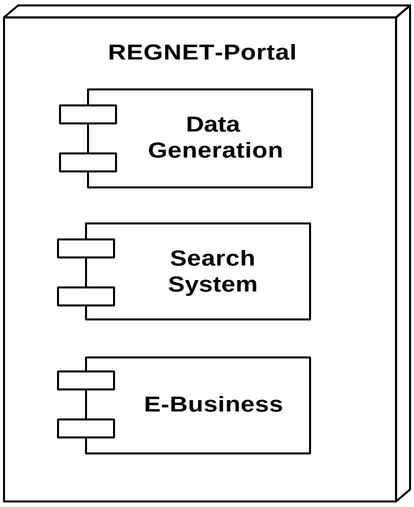
- REGNET-Portal

- REGNET-Cultural Heritage Data Management
- REGNET e-Business Data Management
- REGNET-Ontology Checker
- REGNET-Electronic Publisher

These building blocks host **REGNET** subsystems (called ***components***) which provide following functionalities:

- Repository Management (subsystem-1)
- Reference System (subsystem-2)
- Knowledge Base Access (subsystem-3)
- Data Generation (subsystem-4)
- Search and Retrieval (subsystem-5)
- e-Business (subsystem-6)
- Product Catalogue Management (subsystem-7)
- Electronic Publishing (subsystem-8)
- Procurement and Delivery (subsystem-9)

5.2.1 REGNET-Portal



The REGNET-Portal consists of three components (subsystems), enabling access to following system functions:

- Data Generation (subsystem-4);
- Search and Retrieval (subsystem-5);
- e-Business (subsystem-6).

5.2.1.1 Data Generation (subsystem-4)

This subsystem enables the generation of meta data either via a (configurable) data entry facility (loaded into the user's browser) or by sending a harvester to repositories included in Subsystem-1. Meta data is either stored in Subsystem-1 or -2. The data entry and harvester processes can be triggered by document type definitions (residing in Subsystem-3) as needed by the end user (librarian, archivist, curator). The client connected to this subsystem might even support multimedia and 2/3D data input of digital content.

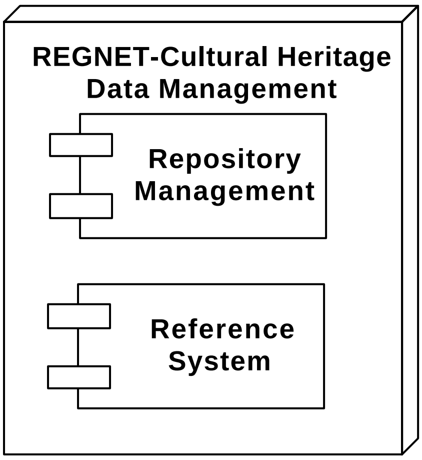
5.2.1.2 Search and Retrieval (subsystem-5)

The Search Subsystems allows the distribution of searches to different repositories and the merging of different result sets delivered by the repository subsystem. It includes a subject gateway which directs the queries in a domain or user profile specific way to the repositories. Besides a query mechanism which is well known in the library/archive/museum world (Z39.50 based), this subsystem also provides the user with the possibility to distribute queries to product catalogues related to e-Business systems (e.g. a museum store). Searches can be done on collection or item level.

5.2.1.3 e-Business (subsystem-6)

Besides the pure access to digital collections of cultural and scientific content, REGNET supports business processes based on digital surrogates. This can be a simple buying function (B2C) of digital surrogates or real objects (museum shop) or even an order to produce a personalised CD-ROM based on raw data coming and pre-selected (shopping cart) from different repositories. The second case involves Subsystem-8, which supports the generation of digital goods, and might involve several suppliers in a B2B case. All sub processes (electronic payment, copy right management, data entry, etc.) are part of this subsystem.

5.2.2 REGNET-Cultural Heritage Data Management:



The REGNET-Cultural Heritage Data Management subsystem facilitates the management of data related to scientific and Cultural Heritage. The connected repositories contain electronic documents as well as surrogates (e.g. Images) of 'real objects'. This node type consists of two components:

- Repository Management (subsystem-1);
- Reference System (subsystem-2).

5.2.2.1 Repository Management (subsystem-1)

Manages repositories containing digital surrogates of 'primary' (real world) objects. A repository may be accompanied by a data base containing meta data within this subsystem which can be accessed by standard protocols (HTTP, Z3950). Using conversion facilities data from legacy systems or not compliant with the REGNET meta data framework can be imported into the REGNET environment.

5.2.2.2 Reference System (subsystem-2)

This subsystem contains meta data related to the repositories included in subsystem-1 or subsystem-7. It allows distributed searches over those repositories. The metadata data base is populated by uploads of subsystem-1 or subsystem-7, or meta data generation done within metadata subsystem-4 (data entry, harvesting).

5.2.3 REGNET e-Business Data Management

The REGNET e-Business Data Management system facilitates the management of data related to products and services. This node is connected to the procurement and delivery of goods and services provided by cultural organisations and consists of two components:

- Product Catalogue Management (subsystem-7)
- Procurement and Delivery (subsystem-9)

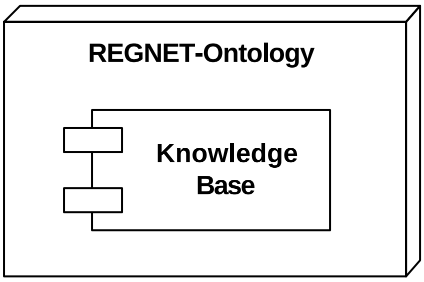
5.2.3.1 Product Catalogue Management (subsystem-7)

This subsystem includes meta data describing products (real or digital) offered by content providers up to services offered by service providers (e.g. consultancy, digitising projects, etc). REGNET allows also search and retrieval of distributed product (and service) catalogues (as included in the ebXML specifications). Doing this the user will be able to compare products which supports his/her buying decision.

5.2.3.2 Procurement and Delivery (subsystem-9)

This subsystem provides access to products and services and transforms requests into real orders or logistic processes. Contractual matters and partnerships among the different stakeholders in the REGNET System (Content Provider, Service Centres, Added Value Generator, Dealer, etc) are dealt with by this subsystem.

5.2.4 REGNET-Ontology System



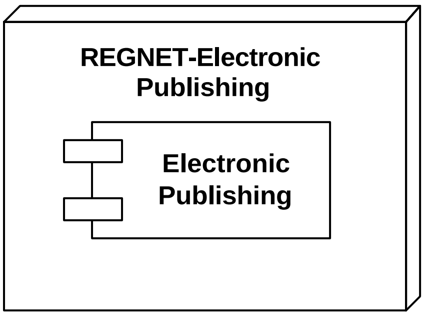
The REGNET-Ontology System is a core element of the REGNET system and guarantees unification with respect to terminologies, metadata, business rules, etc. There is one component hosted by this node:

- Knowledge Base Access (subsystem-3).

5.2.4.1 Knowledge Base Access (subsystem-3)

Subsystem-3 includes data about repositories, document types, domains, user profiles, product catalogues, terminologies, external systems, etc. It can be considered as a layer between the 'user access points' (Subsystem-4, -5, -6) and the different repository (content) related subsystems (Subsystem-1, -2, -7). In addition it will act as a common ‘knowledge base’ for all subsystems by storing and distributing information and data between the subsystems and even between multiple REGNET Systems. It includes different administration tools for managing authority files, thesauri, meta data schemas, document type definitions, etc). This subsystem might be connected to external registries.

5.2.5 REGNET-Electronic Publisher



The REGNET-Electronic Publisher system provides the production of digital products like CD-ROMS, WEB-sites, etc. This node consists of one component:

- Electronic Publishing (subsystem-8)

5.2.5.1 Electronic Publishing (subsystem-8)

Raw data located in digital repositories combined with commercial available data are the basis for personalised electronic publishing. The structuring and navigation of content is supported by the use of a theme based approach which is part of a comprehensive user support according to the LATCH (Location, Alphabet, Time, Category, Hierarchy) structuring model. The products generated are bound to a workflow which specifies the production process which is customisable to user needs. Underlying knowledge and methodologies are accessible via this subsystem allowing the reuse of existing expertise in the handling of this subsystem by other users. Besides the production of CD-ROMs (e.g. using predefined story boards and workflow) the creation of virtual galleries or exhibitions or even WEB-sites are supported. Products or new workflow specifications generated, can be used as new input to the existing range of products and services. he use of XSL and XML-DTDs in defining products will be investigated. By the use of storyboards different visualisation and content structuring techniques supporting the LATCH model will be experienced.

5.2.6 REGNET Connector

The REGNET connector provides the infrastructure that allows the communication between the participants of the network. Communication takes place between CIO and CSC, and between different CIOs. Due to that, this building block is related with all collaborative activities (e.g. distributed search, exchange of material, etc.).

This system is the entry point for B2B collaboration in the way that it is in charge of the marshalling and not-marshalling of data and messages according to the ebXML standard.

5.3 Requirements

Software architectures are derived from functional and technical requirements. In this part a synthesis view of these two kinds of requirements is given which are forming the basis of the architecture described in the following chapters.

Functional requirements express the functionality needed for the REGNET system whereas technical requirements define a set of non functional requirements that the REGNET system must address.

We propose to classify requirements according to the following priorities:

- **1: high priority**. This requirement must be taken into account in the first version of the REGNET system. REGNET V1.0 is the system we will deploy at the end of the first year.
- **2: middle priority**. This requirement will be taken into account in the second version of the system. This version will be elaborate during the second year of the REGNET project.
- **3: low priority**. This requirement should be taken into account in future versions of the REGNET system.

In the following tables we organise requirement in named groups. For each requirement of a group one can find:

- A references to the requirements according to following rules (ex: FR-PT-01-01):
  - FR stand for Functional Requirement; TR stand for Technical Requirement.
  - Second part (PT in the previous example) refer to subsystem:
    - PT: Portal
    - DG: Data Generation
    - SR: Search and Retrieval
    - EB: e-Business
    - RM: Repository Management
    - RS: Reference System
    - PC: Product Catalogue
    - PI: Procurement and Delivery
    - KB: Knowledge Base Access
    - EP: Electronic Publishing
  - Third and fourth part are respectively group and requirement number.
- A short description of the requirement.
- Priority is given according to previous rule.
- An optional comment.
- Use case column refer to the reference of the Use Case provided below and describing the requirement.
- Cross reference column gives reference to Content Provider Use Cases expressed in deliverable D1.

5.3.1 Functional requirements

Functional requirements are given by the following tables corresponding to systems and subsystems as described before.

5.3.1.1 Portal

This part describes specific functionalities provides by the REGNET Portal.

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 1: Access					
FR-PT-01-	Provides an access point to	1	For the user	UC1.0.1	N/A

01	Knowledge Base Access subsystem		and terminal profiling		
FR-PT-01-02	Provides an access point to Search & Retrieval subsystem	1		UC1.0.2	N/A
FR-PT-01-03	Provides an access point to Data generation subsystem	1		UC1.0.3	N/A
FR-PT-01-04	Provides an access point to E-business subsystem	1		UC1.0.4	N/A
Group 2: Display – UC1.0.d					
FR-PT-02-01	Displays results from Search & Retrieval subsystem	1		UC1.0.5	BUCs 24, 25.1, 25.2, 25.3, 31
FR-PT-02-02	Displays E-business related activities	1		UC1.0.6	BUCs 31, 32, 33, 38, 39, 40, 43
FR-PT-02-03	Displays Knowledge Base Access information	1	For user and terminal profiling	UC1.0.7	BUCs 2, 23.2, 31.1
FR-PT-02-04	Displays data from Repository Management subsystem	1	It is used when a user requests a real object	UC1.0.8	BUCs 25, 31.3
Group 3: Session – UC1.0.s					
FR-PT-03-01	Keeps track of user session	1		UC1.0.9	BUCs 1 and 9
Group 4: Editing – UC1.0.e					
FR-PT-04-01	Provides edit functionality for Knowledge Base Access subsystem	1	For user and terminal profiling	UC1.0.10	Several Data Generation BUCs
FR-PT-04-02	Provides edit functionality for Search & Retrieval subsystem	1		UC1.0.11	BUCs 23, 26, 27
FR-PT-04-03	Provides edit functionality for Data generation subsystem	1		UC1.0.12	Several Data Generation BUCs
FR-PT-04-04	Provides edit functionality for e-Business subsystem	1		UC1.0.13	BUCs 30, 32.1
Group 5: Navigation – UC1.0.n					
FR-PT-05-01	Provides navigation through Knowledge Base, Search & Retrieval, Data Generation, e-Business and Repository Management subsystems views	1		UC1.0.14	N/A

Table 15: Functional Requirements - Portal

5.3.1.1.1 Data Generation

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 01 : General					
FR-DG-01-01	Digitise Objects	1	Provision of methodology (How-to-Manuals online)	UC 1.1.1	BUC 10
FR-DG-01-02	Generate new data (object information/products etc.)	1		UC 1.1.2	BUC 11.1 & BUC 30.1 (shop items)
FR-DG-01-03	Import/export existing data from local system.	1		UC 1.1.3	BUC 11.2
FR-DG-01-04	Upload digitised objects	2	Features to support the process: automatic upload of scanned images (directly after scanning)	UC 1.1.4	BUC 11.3
FR-DG-01-05	Enter background material	3		UC 1.1.5	BUC 11.4
FR-DG-01-06	Enter exhibition/information	2		UC 1.1.6	BUC 11.5
FR-DG-01-07	Update & delete information related to object	1		UC 1.1.7	BUC 12.1 & BUC 30.1 (shop items)
FR-DG-01-08	Update & delete information related to product	1		UC 1.1.8	
FR-DG-01-09	Update & delete multiple issues at once	1		UC 1.1.9	BUC 12.2
FR-DG-01-10	Update & delete other information objects	1		UC 1.1.10	BUC 12.3
FR-DG-01-11	Update & delete digitised objects	1		UC 1.1.11	BUC 12.4
FR-DG-01-12	Create own storyboards, themes, fragments	2		UC 1.1.12	BUC 13.1
FR-DG-01-13	Update & delete own storyboards, themes, fragments	2		UC 1.1.13	BUC 13.2
FR-DG-01-14	Editorial functions	2		UC 1.1.14	BUC 13.3
FR-DG-01-15	Co-operation and suggestion	3		UC 1.1.15	BUC 13.5
FR-DG-01-16	Related different object types to each other	2		UC 1.1.16	BUC 14
FR-DG-01-17	Integrate items in REGNET system	3		UC 1.1.17	BUC 15
FR-DG-01-18	Highlight new entry	2		UC 1.1.18	BUC 16

FR-DG-01-19	Select highlights from collection	1	Professional users (according to their rights	UC 1.1.19	BUC 17
FR-DG-01-20	Use the data for the usage outside the system	2	Professional users (according to their rights)	UC 1.1.20	BUC 18
FR-DG-01-21	Print Out	2		UC 1.1.21	BUC 19
FR-DG-01-22	Provide information about collection & institution	1		UC 1.1.22	BUC 21 the whole use case is specified by sub use cases BUC 21.1 - 21.4
FR-DG-01-23	Integrate existing web site	3		UC 1.1.23	BUC 21.1

Table 16: Functional Requirements – Data Generation

5.3.1.1.2 *Search & Retrieval*

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 1: Search & Retrieval – General					
FR-SR-01-01	provide access to distributed repositories.	1		UC 1.2.1	N/A
FR-SR-01-02	provide access to Cultural Heritage data.	1		UC 1.2.2	N/A
FR-SR-01-03	provide access to e-Business data.	1		UC 1.2.3	N/A
FR-SR-01-04	provide access to other REGNETREGNET nodes.	2		UC 1.2.4	N/A
FR-SR-01-05	provide access to external services	2		UC 1.2.5	N/A
FR-SR-01-06	get statistic about search & retrieval system usage	2	statistical data available for the client depends on users access rights	UC 1.2.14	BUC 7
FR-SR-01-07	context sensitive help pages must be available for all search pages and result set pages	1		(Portal) User Interface	BUC 28
Group 2: Search & Retrieval – Query					
FR-SR-02-01	different levels of search complexities must be available.	1	At least simple and advanced search	User Interface, UC 1.2.1	BUC 23.1

FR-SR-02-02	the user must have support through a thesaurus system.	1	this facility will be used to support the real search	UC 1.2.6	BUC 23.2
FR-SR-02-03	scan / browse collections.	2	this facility will be used to support the real search	UC 1.2.7	BUC 23.2
FR-SR-02-04	two levels of content abstraction should be available: (collection level and item level)	1		User Interface, UC 1.2.1	BUC 23.1
FR-SR-02-05	support for 'normal' users for their searches (simple search)	1	definition of themes, provide grouping	User Interface; UC 1.2.1, UC 1.2.6 – UC 1.2.8	BUC 23.2
FR-SR-02-06	field searches must be possible (advanced search)	1		User Interface, UC 1.2.1	BUC 23.1
FR-SR-02-07	all kind of popular search options must be maintained	1	Boolean operators, truncation, ...	User Interface, UC 1.2.1	BUC 23.1
FR-SR-02-08	choose result set display options	1	number of results on one page, sorting options, presentation form	User Interface, UC 1.2.1	BUC 24.3
FR-SR-02-09	sorting search results	1		UC 1.2.1 – UC 1.2.6	BUC 24.2
Group 3: Search & Retrieval – Result					
FR-SR-03-01	present the results due to user selection (includes browsing the result)	1	Different levels of presentation: short, complete, native, user	User Interface, UC 1.2.1 – UC 1.2.6	BUC 24.1 BUC 24.3 BUC 25.1 BUC 25.2 BUC 25.3 BUC 25.4
FR-SR-03-02	integration of search results into new search requests	1		User Interface, UC 1.2.1	BUC 24.4 BUC 27
FR-SR-03-03	saving search results (result sets)	1		UC 1.2.11	BUC 25.6
FR-SR-03-04	saving search queries	1		UC 1.2.10	BUC 26.1
FR-SR-03-05	show status about current search	2		UC 1.2.13	BUC 7
FR-SR-03-06	show status about current system load	3		UC 1.2.13	BUC 7

FR-SR-03-07	search for location of the real object	2	where is the real object located	UC 1.2.1	BUC 23
FR-SR-03-08	provide browsing facilities	1		UC 1.2.7	BUC 24.1
FR-SR-03-09	advertisements shall be carried along with content	2		UC 1.2.1,UC 1.2.3	BUC 25.1
FR-SR-03-10	refine previous queries	2		UC 1.2.12	BUC 26.2 BUC 26.4

Table 17: Functional Requirements – Search & Retrieval

5.3.1.1.3 *e-Business*

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 01: General					
FR-EB-01-01	Reengineer existing processes	1	For each content provider	UC 1.3.2.3 UC 1.3.2.4 UC 1.3.2.5	BUC 44
FR-EB-01-02	Marketing	1	Internet as the new channel for marketing according to the needs of the products	UC 1.3.2.1 UC 1.3.2.2	BUC 44 – BUC 45
FR-EB-01-03	Sales and Order Management	1	Collaboration with the rest of the subsystems	UC 1.3.2.9 UC 1.3.2.10	BUC 34 – BUC 35 – BUC 37 – BUC 38 – BUC 40
FR-EB-01-04	Shopping Support for Customer	1	Collaboration with the rest of the subsystems	UC 1.3.1.1 - UC 1.3.1.39	BUC 30 to BUC 39, BUC 48
FR-EB-01-05	Assist the Procurement	2		UC 1.3.2.6 UC 1.3.2.7 UC 1.3.2.8	BUC 37- BUC 38

Table 18: Functional Requirements – e-Business

5.3.1.2 Cultural Heritage Data Management

5.3.1.2.1 *Repository management*

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 01: General					

FR-RM-01-01	Provide information	1	present the data	UC 2.1.1	N/A
FR-RM-01-02	Upload / delete data	1	Data management	UC 2.1.2	
FR-RM-01-03	Enter data	1	Data acquisition	UC 2.1.3	

Table 19: Functional Requirements – Repository management

5.3.1.2.2 Reference System

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 01 : Reference System - Meta data management					
FR-RS-01-01	create new repository	1	for Data Generation subsystem and external components	UC 2.2.1	BUC 21
FR-RS-01-02	delete repository	1	for Data Generation subsystem and external components	UC 2.2.2	N/A
FR-RS-01-03	insert data into repository	1	for Data Generation subsystem and external components	UC 2.2.3	BUC 11
FR-RS-01-04	update data in repository	1	for Data Generation subsystem and external components	UC 2.2.4	BUC 12
FR-RS-01-05	delete data in repository	1	for Data Generation subsystem and external components	UC 2.2.5	BUC 12
FR-RS-01-06	ensure data integrity	1	check with Ontology System	UC 2.2.7	N/A
FR-RS-01-07	provide (interface for) external data entry components	1	important for 1^{st} phase to get test data into the system	UC 2.2.1 – UC 2.2.5	N/A
Group 02: Reference System – Search					
FR-RS-02-01	search	1	partly XER/Z39.50 target functionality	UC 2.2.8	BUC 23

FR-RS-02-02	transform queries into proper format	2	if other query format is supported	UC 2.2.9	N/A
FR-RS-02-03	search in (local) Cultural Heritage meta data repositories	1	provide API's to connected databases	UC 2.2.10	BUC 23
FR-RS-02-04	manage search session	1	session (result set) management from Z39.50	UC 2.2.11	BUC 23
FR-RS-02-05	create ID	1		U1C 2.2.12	N/A
FR-RS-02-06	find ID	1		UC 2.2.13	N/A
FR-RS-02-07	delete ID	1		UC 2.2.14	N/A
FR-RS-02-08	process results	1	transform results according to query specifications	UC 2.2.15	BUC 23

Table 20: Functional Requirements – Reference System

5.3.1.3 e-Business Data Management

5.3.1.3.1 *Product catalogue management*

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 01: General					
FR-PC-01-01	Continuous update of the stored information	1		UC 3.1.7	N/A
FR-PC-01-02	Provide reports according to the requests	1			N/A
FR-PC-01-03	Provide access to the distributed catalogues	1	The key feature of the PCM	UC 3.1.1, UC 3.1.3	N/A
FR-PC-01-04	Refinement of the query according to desired format	1	Internal translation of the query	UC 3.1.2	N/A
FR-PC-01-05	Search the distributed catalogues	1			N/A
FR-PC-01-06	Return the results of the query to specific sub-node in the desired format	1	Results refinement in the most appropriate format	UC 3.1.3	N/A
FR-PC-01-07	Update the catalogues of the suppliers	1	Catalogue Management by the suppliers	UC 3.1.7	N/A

FR-PC-01-08	Insert new data in the distributed catalogues	1	Catalogue Management by the suppliers	UC 3.1.12	N/A
FR-PC-01-09	Verify the update of the catalogues to the responsible node	1	Catalogue Management by the suppliers	UC 3.1.8	N/A

Table 21: Functional Requirements – Product Catalogue Management

5.3.1.3.2 Procurement and delivery

This part describes specific functionalities provide by the procurement and delivery component.

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 01 : Procurement, MarketPlace					
FR-PI-01-01	Presentation of MarketPlace			UC3.2.1	BUC 43 – (and sub use cases)
FR-PI-01-02	Supplier registration			UC3.2.2a, b	BUC 1.2
FR-PI-01-03	MarketPlace Catalogue			UC3.2.3	N/A
FR-PI-01-04	Management Supplier showcase			UC3.2.4a, b,c	N/A
FR-PI-01-05	MarketPlace newsletter			UC3.2.5	N/A
FR-PI-01-06	Consultation of the MarketPlace products/services			UC3.2.6a, b,c	Information and Service use cases
FR-PI-01-07	Order			UC3.2.7a, b,c	N/A
FR-PI-01-08	Historical account			UC3.2.8	N/A
FR-PI-01-09	Feedback			UC3.2.9	BUC5
FR-PI-01-10	Statistic			UC3.2.10	BUC22
Group 02 : Delivery Invocation					
FR-PI-02-01	Registration of REGNET Structure worker			UC3.2.11	BUC1.2
FR-PI-02-02	Transport Company rates			UC3.2.12	N/A
FR-PI-02-03	Transport Company payment			UC3.2.13	N/A
FR-PI-02-04	Preparation of the order's parcel			UC3.2.14	N/A
FR-PI-02-05	Sending of parcels			UC3.2.15	N/A
FR-PI-02-06	Tracking of Order			UC3.2.16	BUC38.1
FR-PI-02-07	Customer's question			UC3.2.17	BUC38
FR-PI-02-08	Registration of returned parcel			UC3.2.18	N/A

Table 22: Procurement and delivery

5.3.1.4 Knowledge base access

This part describes specific functionalities provides by the REGNET Portal.

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 01: Ontology System: Knowledge Base Data Management					
FR-KB-01-01	Check data integrity	1	Use of proper methods (e.g. DTD or XML-Schemas	UC 4.1	Part of data generation, esp. theme-based approach in BUC 13ff. + Special collection management
FR-KB-01-02	Update Knowledge base	1	Data about doc types, domains etc	UC 4.1.1	
FR-KB-01-03	Insert data in Knowledge base	1	Data about doc types, domains, etc	UC 4.1.2	
FR-KB-01-04	Delete data in Knowledge base	1	Data about doc types, domains, etc	UC 4.1.3	
FR-KB-01-05	Check XML schemas	1	For verification	UC 4.2	
Group 02: Ontology System: Search Knowledge Base					
FR-KB-02-01	Search Knowledge base	1	Use of style sheets (XSL)	UC 4.3	Part of BUC 23
FR-KB-02-02	Export results	1	Transformed to proper format according to query (e.g. dbXML)	UC 4.4	N/A
FR-KB-02-03	Translate query	2	For later versions and various formats that are going to be supported	UC 4.5	N/A
FR-KB-02-04	Transform data to ontology data	1	Encapsulate knowledge. Use of style sheets	UC 4.6	N/A
FR-KB-02-05	Find stylesheet	1	For the transformation of data	UC 4.6.1	N/A
FR-KB-02-06	Find thematic collection	1	TopicMaps	UC 4.6.2	Part of BUC 23.2
FR-KB-02-07	Manage Search Session in knowledge base	1	Define a result set	UC 4.7	BUC 4
FR-KB-02-08	Create Session ID	1	Create result set in knowledge base	UC 4.7.1	BUC 4
FR-KB-02-09	Find Session ID	1	Find result set in knowledge base	UC 4.7.2	BUC 4
FR-KB-02-10	Delete Session ID	1	Delete result set	UC 4.7.3	BUC 4

			in knowledge base		
FR-KB-02-07	Manage User/Terminal Profiles	1	Handle the profiles	UC 4.8	BUC 4
FR-KB-02-08	Create User/Terminal Profile ID	1	Create new profile	UC 4.8.1	BUC 4
FR-KB-02-09	Find User/Terminal Profile ID	1	Find an existing profile	UC 4.8.2	BUC 4
FR-KB-02-10	Delete User/Terminal Profile ID	1	Delete an existing profile (e.g. logout)	UC 4.8.3	BUC 4

Table 23: Knowledge base access

5.3.1.5 Electronic publishing

This part describes specific functionalities provided by the Electronic Publishing.

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 01: General					
FR-EP-01-01	The selection (Metadata + artefacts) should be editable. (insert, delete records; select attributes)	1		UC 5.5, UC 5.5.2 UC 5.5.1	BUC24
FR-EP-01-02	The creation of printer-friendly version should be provided	1		UC 5.8 UC 5.7.1	BUC19, BUC20
FR-EP-01-03	A storyboard should be provided	1		UC 5.6 UC 5. 12 UC 5.10.1	N/A
FR-EP-01-04	The User should be able to extend a ResultSet with knowledge from the Ontology (User Profile, personal preferences)	1		UC 5.4.1	BUC27, BUC29
FR-EP-01-05	A reusable product that can be sold in shops should be creatable	2		UC 5.9 UC 5.7.1 UC 5.7.2 UC 5.7.3	BUC41
FR-EP-01-06	The theme approach should be supported			UC 5.10 UC 5.10.1	N/A
FR-EP-01-07	Create an exhibition			UC 5.11	BUC41
FR-EP-01-08	Create a virtual gallery			UC 5.12	BUC41
FR-EP-01-09	Create a catalogue			UC 5.13	N/A

Table 24: Functional Requirements - Electronic Publishing

5.3.1.6 REGNET Connector

This part describes specific functionalities provided by the REGNET Connector.

Reference	Requirement	Priority	Comment	Use Case	Cross reference
Group 01: General					
FR-RC-01-01	Provide access to external REGNET sites	1	Compliant with ebXML	UC6.1	N/A
FR-RC-01-02	Provide access from external REGNET sites	1	Compliant with ebXML	UC6.2	N/A

Table 25: Functional Requirements – REGNET Connector

5.3.2 Technical requirements

Technical requirements are given by the following table :

Reference	Requirement	Priority	Comment
Group 1: General technical requirement			
TR-01-01	Multi-tiers architecture	1	Multi parts of the system could be separated.
TR-01-02	Exception handling mechanism	1	
TR-01-03	Distributed architecture	1	
TR-01-04	Portability	1	
Group 2: client			
TR-02-01	Support HTTP-based browsers connected via a wired access	1	
TR-02-02	Support WAP-based browsers connected via a wireless access	1	
TR-02-03	Support HTTP-based browsers connected via a wireless access	2	JAVA-enabled handsets, PDAs, etc.
TR-02-04	Support Voice browsing	3	Wired and Wireless phones
TR-02-05	Automatic deployment	1	For heavy clients.
TR-02-06	GSM support	1	For wireless clients
TR-02-07	GPRS support	2	For wireless clients
TR-02-08	UMTS support	3	For wireless clients
Group 3: ergonomics			
TR-03-01	Common graphic presentation rules	1	Presentation from software modules must be conform to a set of graphic rules defined for the REGNET system.
TR-03-02	Software internationalisation	1	This requirement must be taken into account in the architecture definition. The English version will be deploy for the first version.
TR-03-03	Dynamic language switching	2	

TR-03-04	Help online	2	First version: basic guideline, FAQ
Group 4: data management			
TR-04-01	Back-up strategies	2	
Group 5: performances			
TR-05-01	Response time: < 5s for usual local functionalities	1	Functionalities which address other REGNET sites can be longer.
TR-05-02	Number of authorised user: Unlimited	1	
TR-05-03	Number of simultaneous user: 100	1	
TR-05-04	Scalability	2	
Group 6: security			
TR-06-01	Log of system usage	2	Can also be used for CRM strategies
TR-06-02	Login password mechanism	1	
TR-06-03	Single logon mechanism	1	Imply security context propagation between nodes.
TR-06-04	Group of users	1	
TR-06-05	Access segregation according to user / group.	2	
TR-06-06	Secure architecture	2	Possibility for the REGNET system to be deploy in a DMZ[2]. Firewall specificities must be address.
TR-06-07	On line payment	1	
Group 7: system management			
TR-07-01	Emergency system	2	Implemented by the exception handling mechanism.
TR-07-02	Statistic analysis	3	
TR-07-03	Redundant architecture	3	
TR-07-04	Availability	7/24	Accept interruption for maintenance intervention. 95% availability
Group 8: software integration			
TR-08-01	Connection with Collection Management System from OpenHeritage	1	
TR-08-02	Search engine from AIT	1	JAVA / RMI API
TR-08-03	Standard REGNET gateway for legacy systems	3	
Group 9: hardware			
TR-09-01	Client must be supported by any hardware	1	GUI based on Internet de-facto standards compliant browser

[2] Demilitarized Zone

TR-09-02	Multiple processor system	3	
Group 10: others			
TR-10-01	Use of free / open-source software's as far as possible	1	
TR-10-02	Training courses for system administrator and officials in charge	2	
TR-10-03	Help desk	3	Call centre in a regional centre
TR-10-04	Technical support: to be specified		Installation, Bug
TR-10-05	User manual	1	
TR-10-06	Reference manual	1	
TR-10-07	Installation manual	1	

Table 26: Technical Requirements

5.3.2.1 Internationalisation and localisation

The ability of a system/program to adapt to the specific cultural characteristics, such as language and presentation formats is called “localisation”. Generally, this applies for graphical user interfaces, error messages and cultural dependant data such as date, time, measuring units, collation, etc. Doing this in the same code base becomes very complex because you have to deal with different character sets, different architectures and different markets. Moreover, adding a new language for instance needs recompiling of the code and making changes to items scattered around in the code increases the risk for bugs. In order to avoid all these constraints and problems, the specific “local” elements have to be isolated from the common code of the system/program and put in a separate area. This is called “internationalisation”.

Besides the common cultural characteristics, mentioned in the previous point and which can be categorised as typical for operating systems and the more generic application programs, REGNET contains a supplementary number of strong “regional” aspects that go a step further. The first being the availability, at the content generation side, of a framework to introduce a great number of multilingual textual descriptions of Cultural Heritage themes and objects, the second concerning the feature to adapt or vary the presented content at a certain moment to specific localisation parameters. The former is based on the same approach as the common cultural characteristics but this time on a much larger scale. The latter can be best explained by an example. A person from a certain region, browsing through a Cultural Heritage description will receive by priority, if appropriate and available, a supporting image of an object related to his own region. A person from another region will receive, in the same context, another image that best relates to his region. A third supplementary internationalisation effort lies in the introduction of multilingual references/thesauri as far as possible and available.

The broad target markets of REGNET makes of the internationalisation and localisation topic a very important objective. Therefore an as open and flexible possible solution has to be chosen to comply with the internationalisation and localisation needs. Intelligent context handling within all constituting REGNET-modules will cope with all these internationalisation issues.

5.3.2.2 Integration

Need for integration is crucial when deciding to build a system which address the implementation of a collaborative environment as described before. REGNET as a network, has to integrate heterogeneous pieces of software:

- At the site level, a REGNET site consists of nodes (Ontology System, Cultural Heritage Data Management, e-Business Data Management, etc.). Nodes can be distributed on multiple computer and based on different technologies. This requirement correspond to **A2A** (Application to Application) integration.

- At the REGNET level, the REGNET network consists of an Internet linked community of sites. Each sites can query others in order to get information or services about information managed by others site. This requirement correspond to **B2B** (Business to Business) integration.

B2B integration is mainly provided by the ebXML model described before and implemented by the REGNET connector.

For A2A integration one must takes into account following aspects:

- REGNET provide a set of functionalities implemented by nodes. The availability of a functionality at the site level depend on the managed institutions and electronic availability of collections. A new functionality must be easily added as far as it becomes usable for the site.
- Different technologies must be integrated in the REGNET system. These technologies come from previous project (e.g. Search engine from COVAX) or from OpenHeritage according to the clustering organisation.
- Moreover we have to address the fact that partner provides different technical skill. Two groups can be identified: one which is related to PHP, the other to JAVA platform. Despite the fact that each technology is more or less adapted to a context, we have to define an architecture which provides easy and smooth integration of technologies.

5.4 Use cases

This part contains Actors and Uses Cases for the REGNET system described according to the UML.

The Unified Modeling Language™ (UML) is a language for specifying, visualising, constructing, and documenting the software artefacts, as well as for business modelling and other non-software systems. The UML represents a collection of best practice engineering approaches that have proven to be successful in the modelling of large and complex systems.

The UML was developed by the Object Management Group. It is the successor of the modelling languages found in the methods of Booch, Jacobson (OOSE) and others. Many companies are incorporating the UML as a standard into their development processes and products, which cover disciplines such as business modelling, requirements management, analysis & design, programming and testing.

5.4.1 Definitions

The following definitions specify the UML concepts and tools which are used in this document. They are taken from the UML 1.3 specifications of the OMG. The complete specification is available on the Rational site (http://www.rational.com/uml/resources/documentation/index.jsp).

ACTOR:

An *actor* defines a coherent set of roles that users of an entity can play when interacting with the entity. An actor may be considered to play a separate role with regard to each use case with which it communicates.

In the metamodel Actor is a subclass of Classifier. An Actor has a Name and may communicate with a set of Use Cases, and, at realisation level, with Classifiers taking part in the realisation of these Use Cases. An Actor may also have a set of Interfaces, each describing how other elements may communicate with the Actor.

An Actor may have generalisation relationships to other Actors. This means that the child Actor will be able to play the same roles as the parent Actor, i.e. communicate with the same set of Use Cases, as the parent Actor.

USE CASE:

The use case construct is used to define the behaviour of a system or other semantic entity without revealing the entity's internal structure. Each use case specifies a sequence of actions, including variants, that the entity can perform, interacting with actors of the entity.

In the metamodel Use Case is a subclass of Classifier, specifying the sequences of actions performed by an instance of the Use Case. The actions include changes of the state and communications with the environment of the Use Case. The sequences can be described using many different techniques, like Operation and Methods, Activity Graphs, and State Machines.

There may be Associations between Use Cases and the Actors of the Use Cases. Such an Association states that an instance of the Use Case and a user playing one of the roles of the Actor communicate. Use Cases may be related to other Use Cases by Extend, Include, and Generalisation relationships. An Include relationship means that a Use Case includes the behaviour described in another Use Case, while an Extend relationship implies that a Use Case may extend the behaviour described in another Use Case, ruled by a condition. Generalisation between Use Cases means that the child is a more specific form of the parent. The child inherits all Features and Associations of the parent, and may add new Features and Associations.

<u>CLASS DIAGRAM:</u>

A class diagram is a collection of (static) declarative model elements, such as classes, interfaces, and their relationships, connected as a graph to each other and to their contents. Class diagrams may be organised into packages either with their underlying models or as separate packages that build upon the underlying model packages.

<u>ACTIVITY DIAGRAM:</u>

An activity diagram is a special case of a state diagram in which all (or at least most) of the states are action or sub-activity states and in which all (or at least most) of the transitions are triggered by completion of the actions or sub-activities in the source states. The entire activity diagram is attached (through the model) to a class, such as a use case, or to a package, or to the implementation of an operation. The purpose of this diagram is to focus on flows driven by internal processing (as opposed to external events). Use activity diagrams in situations where all or most of the events represent the completion of internally-generated actions (that is, procedural flow of control). Use ordinary state diagrams in situations where asynchronous events occur.

<u>COMPONENT DIAGRAM:</u>

A component diagram is a graph of components connected by dependency relationships. Components may also be connected to components by physical containment representing composition relationships.

A diagram containing component types and node types may be used to show static dependencies, such as compiler dependencies between programs, which are shown as dashed arrows (dependencies) from a client component to a supplier component that it depends on in some way. The kinds of dependencies are implementation-specific and may be shown as stereotypes of the dependencies.

As a classifier, a component may have operations and may realise interfaces. The diagram may show these interfaces and calling dependencies among components, using dashed arrows from components to interfaces on other components.

This part detail use cases deduce from functional user requirements.

Use cases describe the sequence of events of an actor using a system to complete a process. They:

- Represent business or domain processes.
- Are a narrative description of a business process.
- Are expressed in structured prose.
- Are not directly related to object technology.

An actor in a Use Case is an external agent that uses or interacts with the system. There is one *initiator* actor and possibly several *participating* actors. Actors may be human or not human (e.g. external software systems).

In the current state of the work global Use Cases are identified. These will be elaborated in WP2, as a starting point for system design.

In the following paragraphs Use Cases are grouped according to the defined subsystems. For each subsystem, one or many Use Case diagrams are given accompanied by a table which gives a short description of each Use Case.

5.4.2 Actors

The REGNET actors (the system users) are shown in the figure below. These are the general system actors which may be adopted to each subsystems specialities. Specialised Use Case actors are detailed in the according subsystem section.

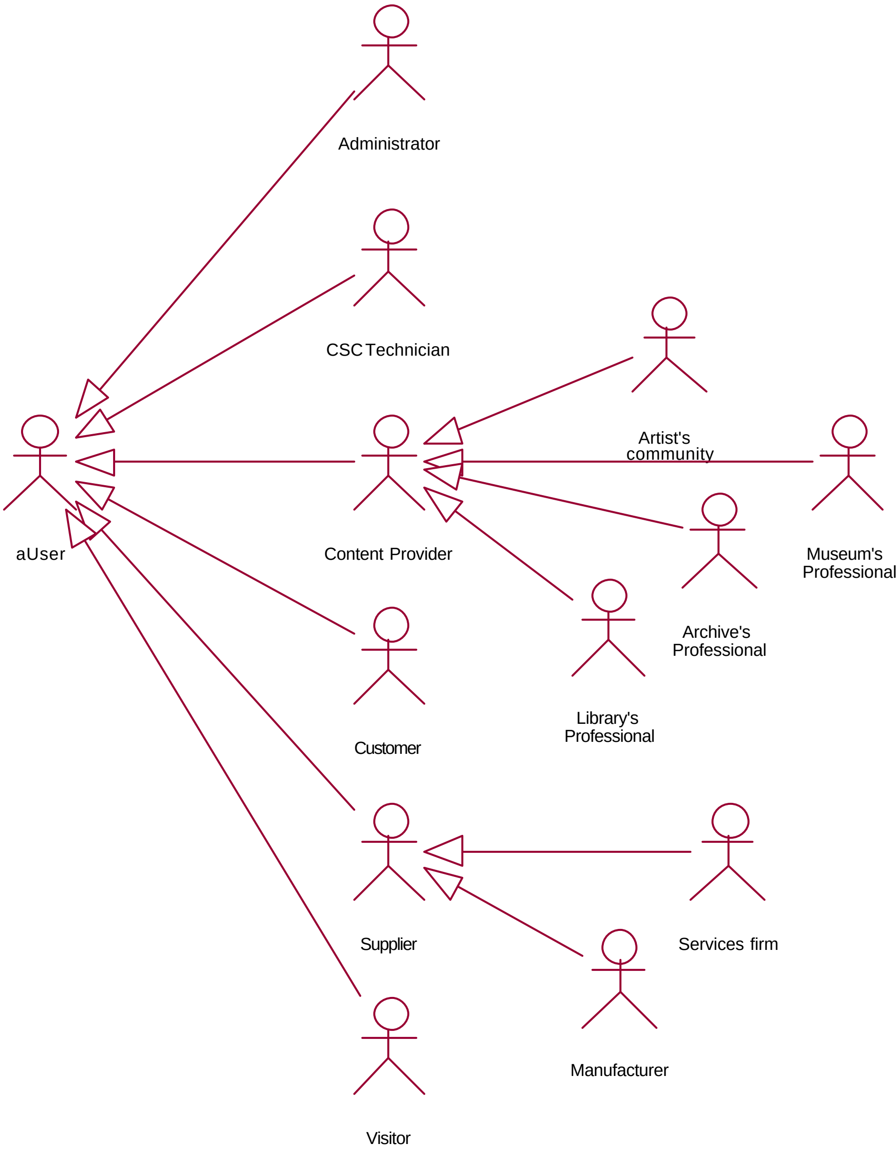


Figure 17: Actors

A REGNET's user can be:

Actors	Role
Administrator	The REGNET Administrator manages all the REGNET users identified for the system: partners, supplier, consumers. (ONTOLOGY).
CSC Technician	The CSC provides support activities to subscribe CIO
Content Provider	**The content provider or CIO places its cultural resources at REGNET user disposal.**
Customer	The customer orders REGNET products or services, and pays for them. Examples : a client of one of the REGNET e-shops. a researcher who orders special enquiries, or pays to have access and usage rights for REGNET cultural data a collector who needs expert advice.
Supplier	A supplier offers products or services to the REGNET consortium. Examples : a manufacturer who provides goods for the e-shops. a publishing firm a transport company
Visitor	A user who visits the REGNET's site. a anonymous user of the REGNET System.

Table 27: REGNET Users

5.4.3 Portal

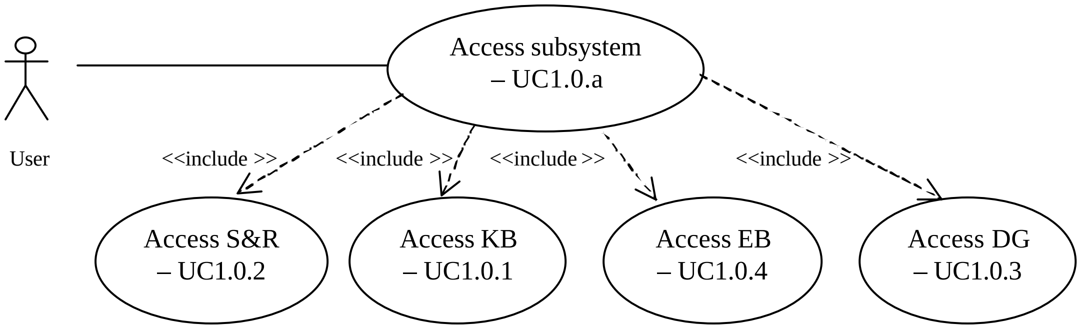


Figure 18: Use Case diagram UC1.0.a

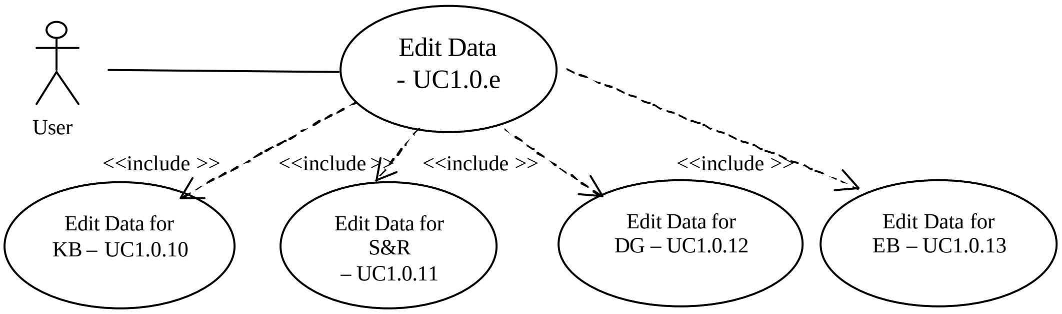

User
Edit Data - UC1.0.e
<<include >>
<<include >>
<<include >>
<<include >>
Edit Data for KB – UC1.0.10
Edit Data for S&R – UC1.0.11
Edit Data for DG – UC1.0.12
Edit Data for EB – UC1.0.13

Figure 19: Use Case diagram UC1.0.e

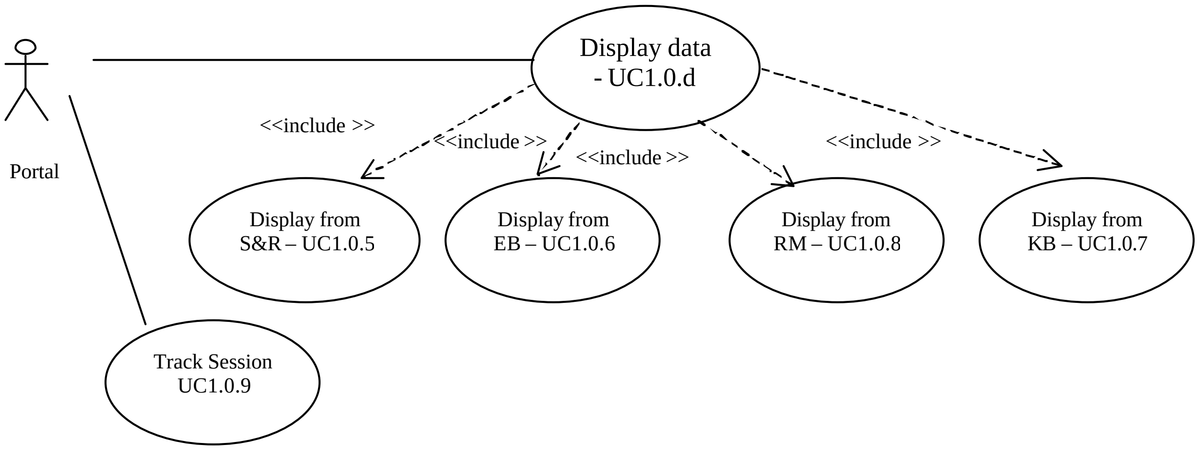

Portal
Display data - UC1.0.d
<<include >>
<<include >>
<<include >>
<<include >>
Display from S&R – UC1.0.5
Display from EB – UC1.0.6
Display from RM – UC1.0.8
Display from KB – UC1.0.7
Track Session UC1.0.9

Figure 20: Use Case diagram UC1.0.d

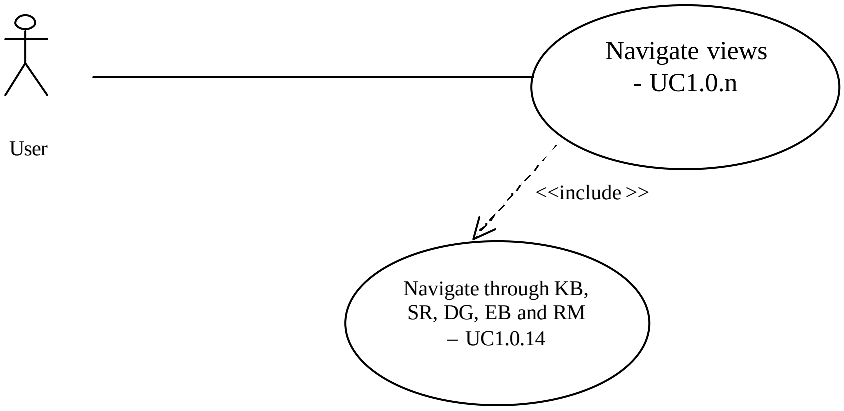


Figure 21: Use Case diagram UC1.0.n

UC Identifier	Use Case	Description
UC1.0.a	Access subsystems	Access: This main group includes several UCs that describe the starting points to access Portal's subsystems
UC1.0.1	Access KB	Represents the main access to the Ontology
UC1.0.2	Access S&R	Represents the main access to the Search & Retrieval subsystem where users can accomplish their searches

UC1.0.3	Access DG	Represents the main access to the Data Generation subsystem, here users can insert new data for REGNET
UC1.0.4	Access EB	Represents the main access to the e-Business subsystem, from here users can buy REGNET artefacts
UC1.0.d	Display data	Display: This main group includes several UCs that describe the functionality of displaying screens needed to accomplish REGNET main functionalities
UC1.0.5	Display from S&R	Represents the display scenario when users are looking for searching results
UC1.0.6	Display from EB	Represents the display scenario during e-Business activities
UC1.0.7	Display from KB	Represents the display scenario during Profiling activities
UC1.0.8	Display from RM	Represents the display scenario during Repository Management activities
UC1.0.s		Session: This main group takes care of session tracking
UC1.0.9	Track session	Track session during user's navigation
UC1.0.e	Edit data	Edit: This main group includes several UCs that describe inputting scenarios
UC1.0.10	Edit data for KB	Represents the inputting scenario to update profiles
UC1.0.11	Edit data for S&R	Represents the inputting scenario to perform a search
UC1.0.12	Edit data for DG	Represents the inputting scenario during the update of a REGNET data
UC1.0.13	Edit data for EB	Represents the inputting scenario during e Business activities
UC1.0.n	Navigate views	Navigation: This main group takes care of navigation issues
UC1.0.14	Navigate through KB, S&R, DG, EB	Represents the scenario of navigation through REGNET Portal

Table 28: Use Case Description - Portal

5.4.3.1 Data Generation

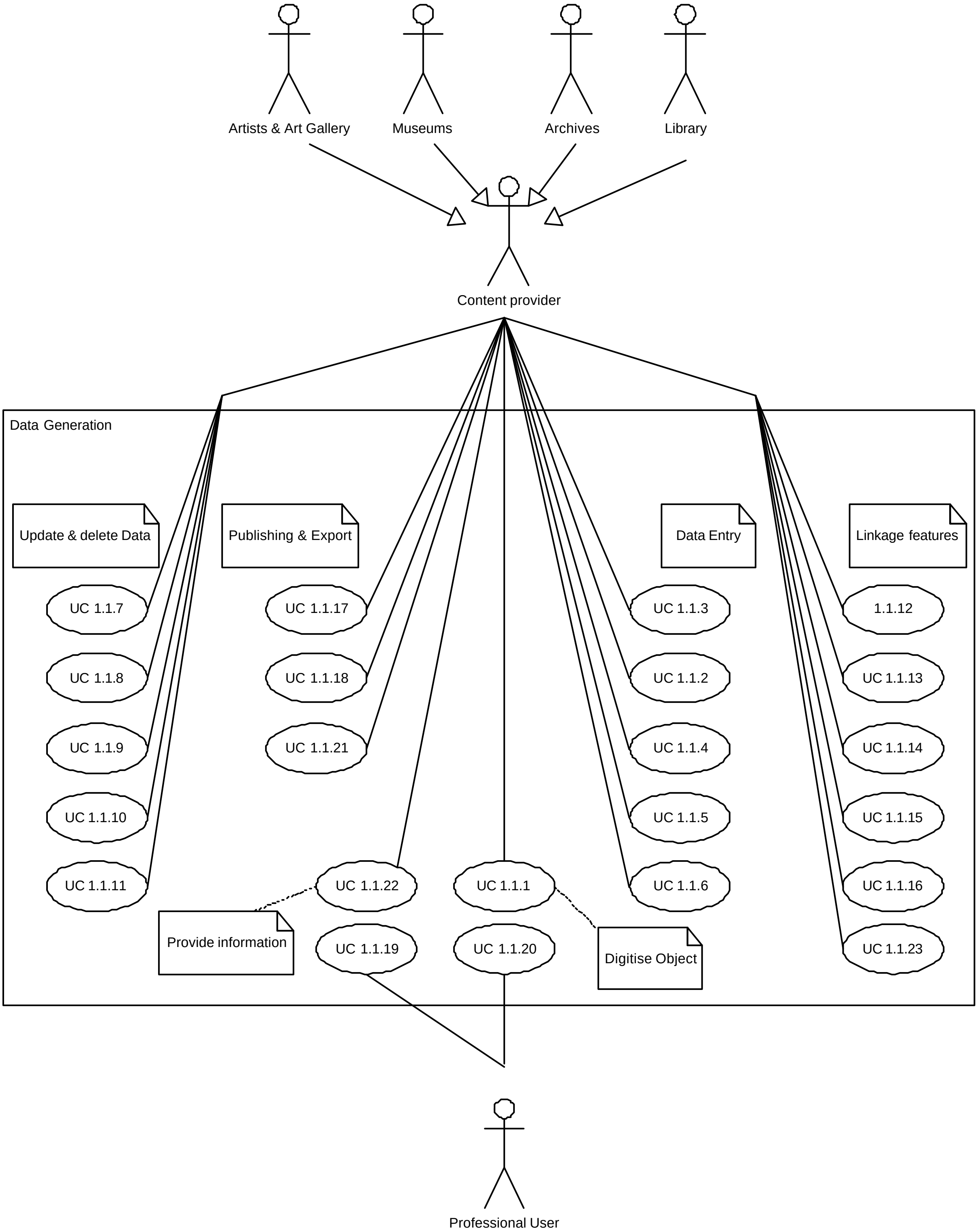


Figure 22: Use Case Diagram – Data Generation

UC Identifier	Use Case	Description
UC 1.1.1	Digitise Objects	Features to support the process of automatic upload of scanned images

UC 1.1.2	Generate new data	Data input (object information or products ...) coming from the Portal Data Entry component: general data input mask and features, adaptable by content providers.
UC 1.1.3	Import/export existing data from local system	Upload and conversion for different data management systems.
UC 1.1.4	Upload digitised objects	For uploading of image files.
UC 1.1.5	Enter background material	Create and upload new or existing texts files (e.g. press releases, sample or multimedia files). Or relate background material to existing objects.
UC 1.1.6	Enter exhibition/infor mation	Create & upload events (Calendar)
UC 1.1.7	Update & delete information related to object	Invoke editing component.
UC 1.1.8	Update & delete information related to product	Invoke editing component.
UC 1.1.9	Update & delete multiple issues at once	Invoke editing component.
UC 1.1.10	Update & delete other information objects	Invoke editing component.
UC 1.1.11	Update & delete digitised objects	Invoke editing component.
UC 1.1.12	Create own storyboards, themes, fragments	Enhanced function that permit to a Content provider to create storyboard, themes and fragments
UC 1.1.13	Update & delete own storyboards, themes, fragments	Enhanced function that permit to a Content provider to change storyboard, themes and fragments previously created.
UC 1.1.14	Editorial functions	Allows to a content provider to review or versioning articles

UC 1.1.15	Co-operation and suggestion	Collect suggestion of new themes, subscription of theme-based products
UC 1.1.16	Related different object types to each other	Possibilities to establish links between the different "areas" and different object types of the system (e.g. shop items and real objects, works and related products like books etc.).
UC 1.1.17	Integrate items in REGNET system	Selection and publishing/registration tools for the REGNET portal (all objects, items etc.) Selection of "areas" where the information should be published (e.g. registration for shop, object database, auctions, other categories)
UC 1.1.18	Highlight new entry	Selection of new items on different presentation levels (objects, artists, partners, products etc.). Goal: Object presentation in special areas of the portal and/or selection for newsletter and other services (automatically)
UC 1.1.19	Select highlights from collection	Selection of items fitting to specific criteria (which could be defined by professionals for its collection).
UC 1.1.20	Use the data for the usage outside the system	Export functions, conversion tools and publishing tools. Quality assurance editorial board. Professional users (according to their rights)
UC 1.1.21	Print Out	Printer-friendly version of the system. All content providers but: with very different needs and goals.
UC 1.1.22	Provide information about collection & institution	In order to describe the institution and collection.
UC 1.1.23	Integrate existing web site	Service On demand. Web Publishing Tools (e.g. semi-automatic creation of sub-collections for publishing on own web site) or e-Business-service (e.g. value-added services for the creation of new pages)

Table 29: Use Case Description – Data Generation

The above diagrams describe in detail the use cases of the Data Generation subsystem embedded into the Portal module. The main functionalities of the Data Generation module is to manage the data entry to the REGNET System.

It is possible to group the use cases into 6 categories of features which allow the Content Providers to enter or manipulate data. We represent this categories grouping them vertically on the UML scheme:

- Objects digitalisation

  It's the first step in the data acquirement. All the objects treated into the REGNET system have to be previously digitised.

- Entry of data information

  The information are coming from the Portal data entry component. The input mask of Portal collect information about digitised object or products or could contain calendar events, etc.

- Update & delete of data information

The portal presents an interface for the modification of the information entered in the system.

- Publishing and export

This group of functions permit the creation of storyboard, themes and fragments using the objects and products present in the system.

- Linkage features

A group of functions that allow the content provider to manage data.

- Providing of information

Describe the institution and collection.

5.4.3.2 Search and Retrieval

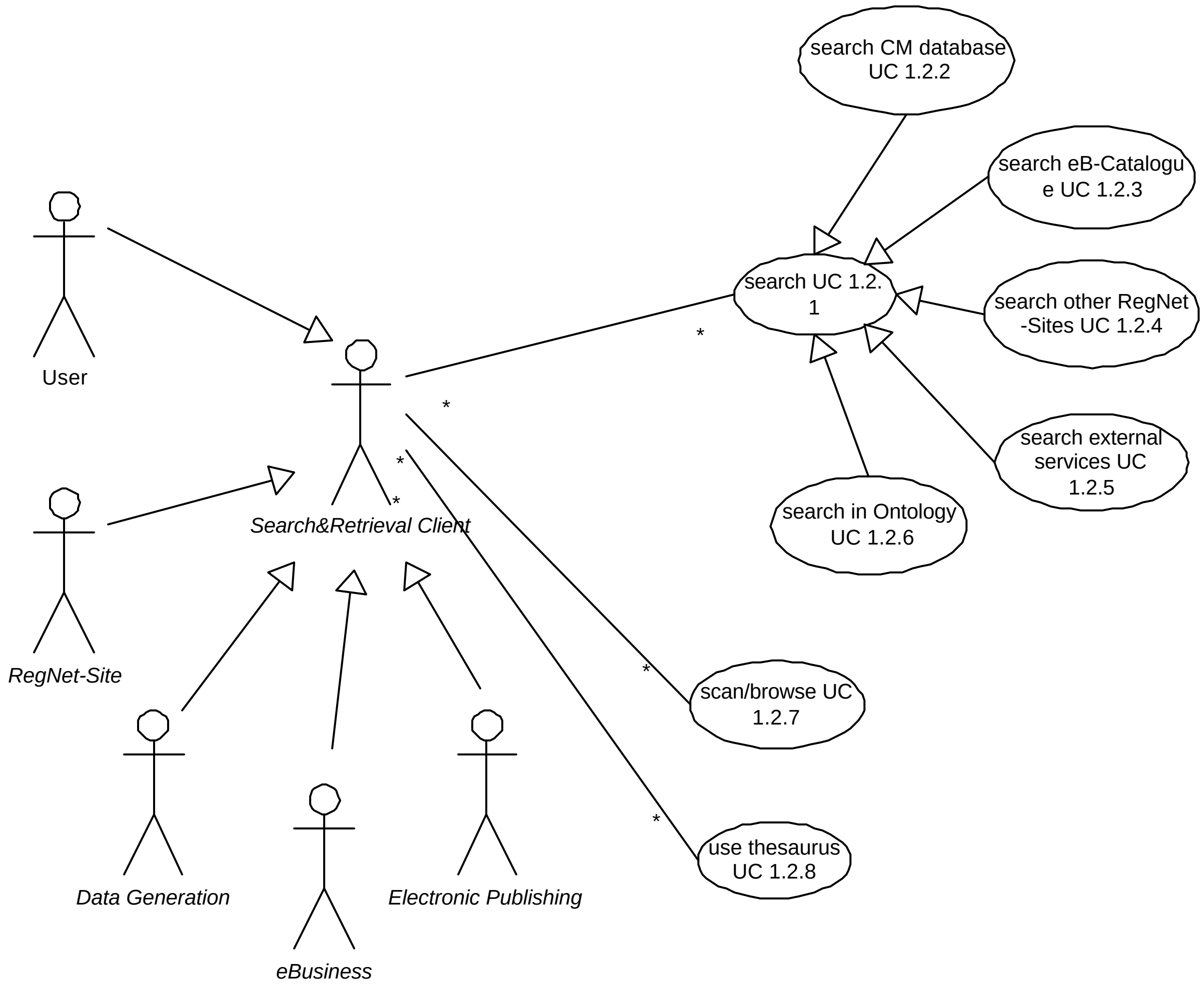

search CM database UC 1.2.2
search eB-Catalogue UC 1.2.3
search UC 1.2.1
search other RegNet -Sites UC 1.2.4
search external services UC 1.2.5
search in Ontology UC 1.2.6
scan/browse UC 1.2.7
use thesaurus UC 1.2.8
User
Search&Retrieval Client
RegNet-Site
Data Generation
eBusiness
Electronic Publishing
*
*
*
*
*
*

Figure 23: Use Case Diagram –Search & Retrieval (all Clients)

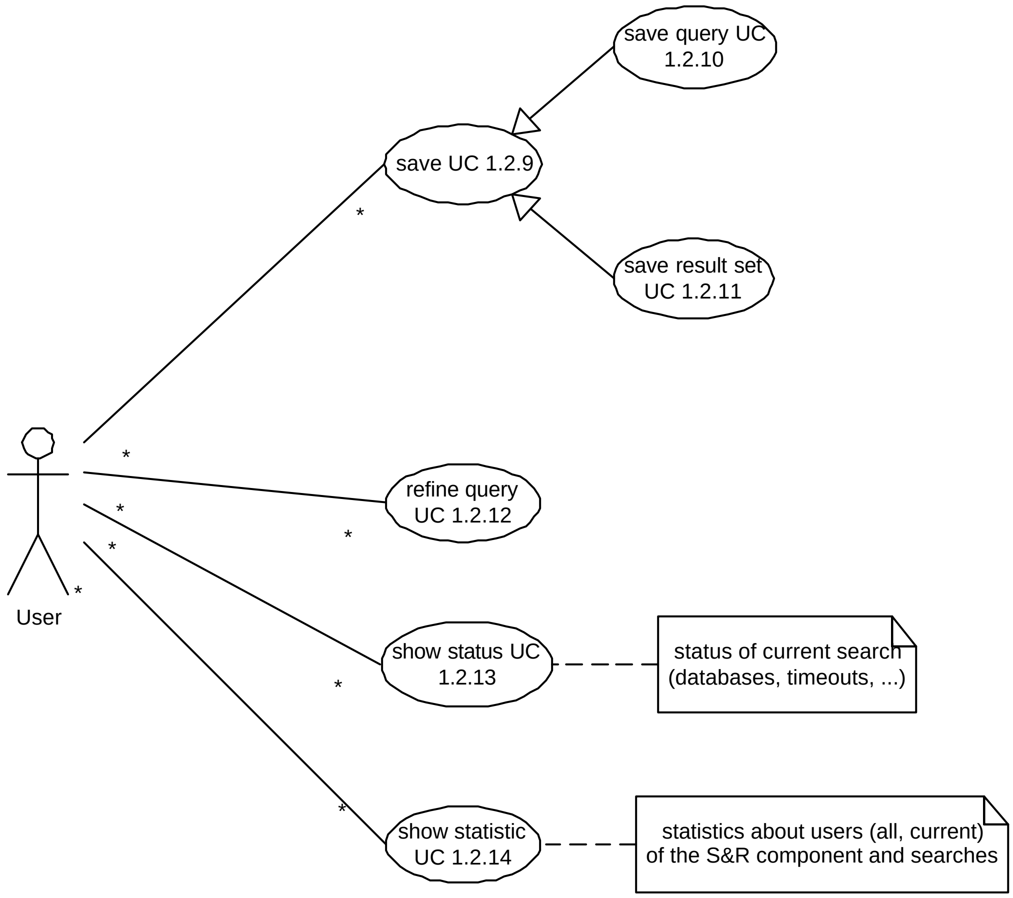

save query UC 1.2.10
save UC 1.2.9
save result set UC 1.2.11
User
refine query UC 1.2.12
show status UC 1.2.13
status of current search (databases, timeouts, ...)
show statistic UC 1.2.14
statistics about users (all, current) of the S&R component and searches
*
*
*
*
*
*
*
*
*

Figure 24: Use Case Diagram –Search & Retrieval (local Portal)

UC Identifier	Use case	Description
UC 1.2.1	Search	**Search requests must distributed to different targets and the results must be merged together. The possible targets are listed in sub-use cases UC 1.2.2 – UC 1.2.6**
UC 1.2.2	Search in Cultural Heritage data	This means to forward the search request to the Reference System.
UC 1.2.3	Search in Product Catalogue data	This means to forward the search request to the Product Catalogue Management.
UC 1.2.4	search in other REGNETREGNET-Sites	This means to forward the search request to another Search & Retrieval component in another REGNETREGNET-Site.
UC 1.2.5	Search external sources	In popular search engines, etc.
UC 1.2.6	Search in Ontology	Search within the Ontology System (Themes, Topic Maps)
UC 1.2.7	Scan / browse	provide a scan / browse functionality to support users with their searches. (related to UC 1.2.6)

UC 1.2.8	Use thesaurus	provide access to thesauri (located in the Ontology System) to support users with their searches. (related to UC 1.2.6)
UC 1.2.9	Save	this use case covers all data related to searches that can be stored with in the Search & Retrieval component. Detailed in UC 1.2.10 – UC 1.2.11
UC 1.2.10	Save query	user queries will automatically be saved within a session (to provide a history) but users can also save some of their queries to a user space
UC 1.2.11	Save result set	users can save single result items and result sets to a user space
UC 1.2.12	Refine query	previous done search request can be refined. this implies that queries must be saved (U4.1) and that a history of queries must be maintained (and accessible)
UC 1.2.13	Show status	show the status of the currently searched data sources (timeout, results found, results already returned, …)
UC 1.2.14	Show statistics	show statistics about users (all, current) and searches of the Search & Retrieval component.

Table 30: Use Case Description – Search & Retrieval

5.4.3.3 e-Business

The main features of the e-Business subsystem are the E-Shop, the Procurement, the B2B and the Delivery functionality. In this section the E-Shop and B2B use cases are described. The procurement and delivery use cases are described later.

E-SHOP:

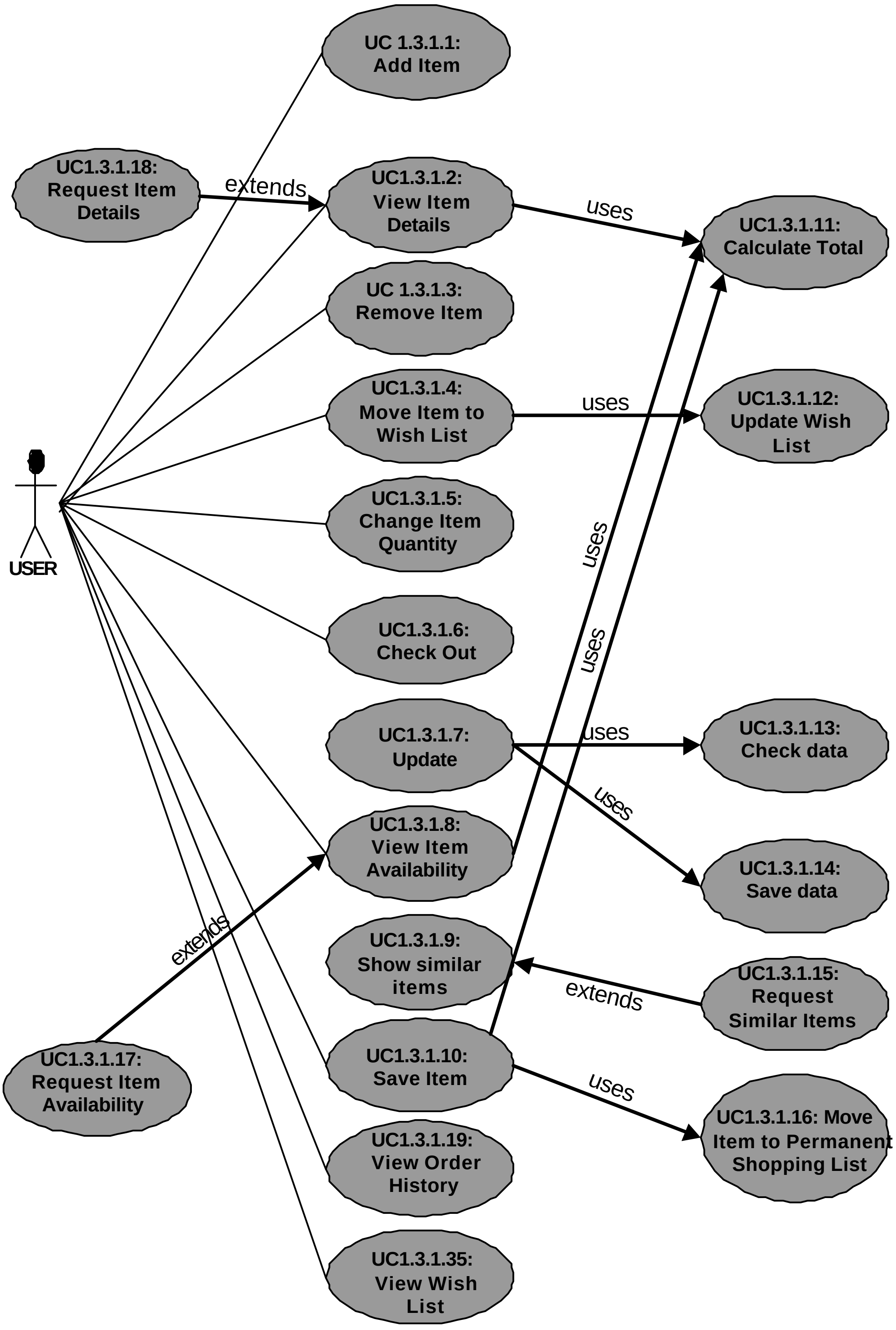

UC 1.3.1.1: Add Item
UC1.3.1.18: Request Item Details
extends
UC1.3.1.2: View Item Details
uses
UC1.3.1.11: Calculate Total
UC 1.3.1.3: Remove Item
UC1.3.1.4: Move Item to Wish List
uses
UC1.3.1.12: Update Wish List
USER
UC1.3.1.5: Change Item Quantity
uses
uses
UC1.3.1.6: Check Out
UC1.3.1.7: Update
uses
UC1.3.1.13: Check data
uses
UC1.3.1.8: View Item Availability
UC1.3.1.14: Save data
extends
UC1.3.1.9: Show similar items
extends
UC1.3.1.15: Request Similar Items
UC1.3.1.17: Request Item Availability
UC1.3.1.10: Save Item
uses
UC1.3.1.16: Move Item to Permanent Shopping List
UC1.3.1.19: View Order History
UC1.3.1.35: View Wish List

Figure 25: Use Case Diagram – e-Business (e-Shop)

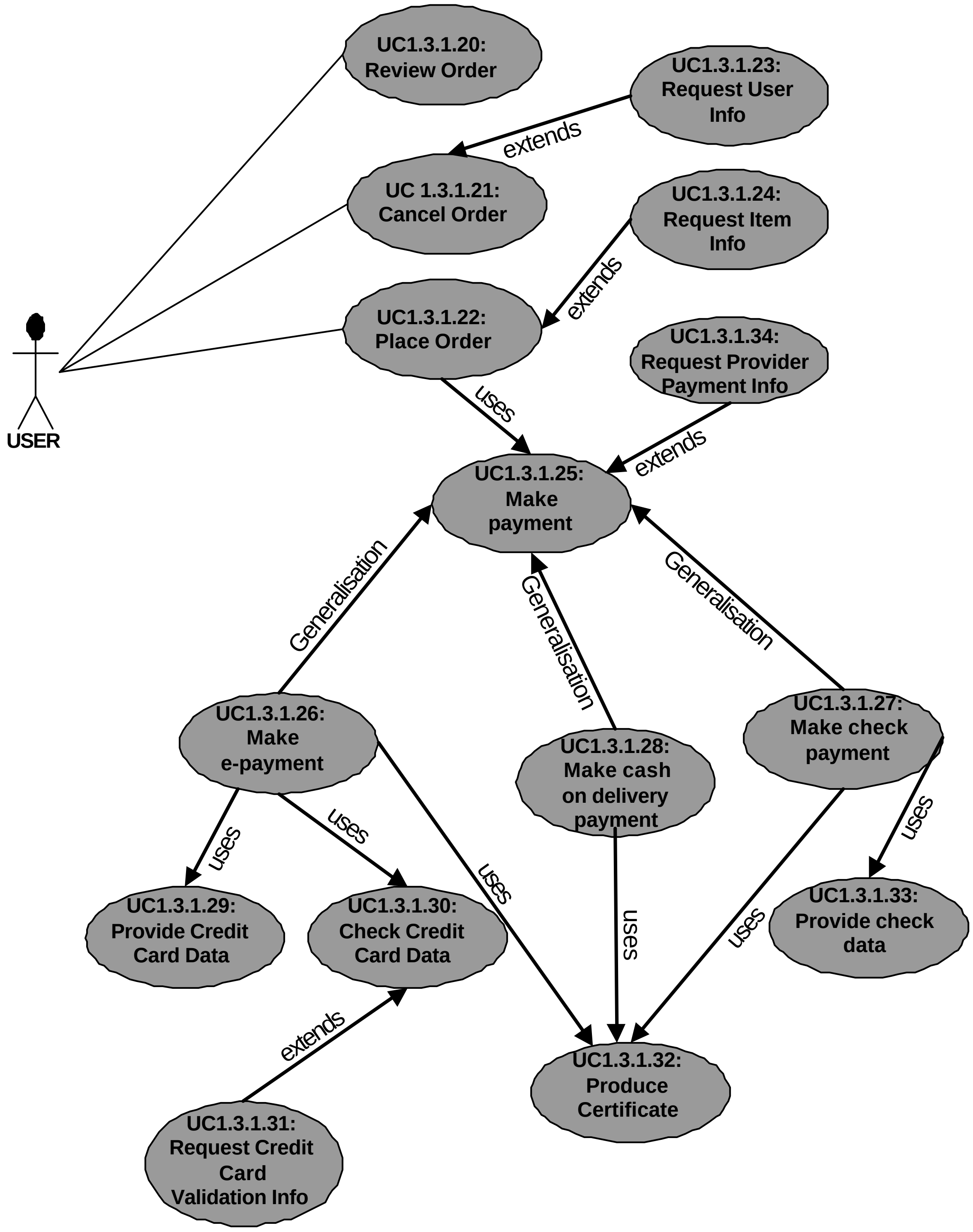

UC1.3.1.20: Review Order
UC1.3.1.23: Request User Info
extends
UC 1.3.1.21: Cancel Order
UC1.3.1.24: Request Item Info
extends
UC1.3.1.22: Place Order
UC1.3.1.34: Request Provider Payment Info
USER
uses
extends
UC1.3.1.25: Make payment
Generalisation
Generalisation
Generalisation
UC1.3.1.26: Make e-payment
UC1.3.1.27: Make check payment
UC1.3.1.28: Make cash on delivery payment
uses
uses
uses
uses
uses
uses
uses
UC1.3.1.29: Provide Credit Card Data
UC1.3.1.30: Check Credit Card Data
UC1.3.1.33: Provide check data
extends
UC1.3.1.32: Produce Certificate
UC1.3.1.31: Request Credit Card Validation Info

Figure 26: Use Case Diagram – e-Business (finalisation of order)

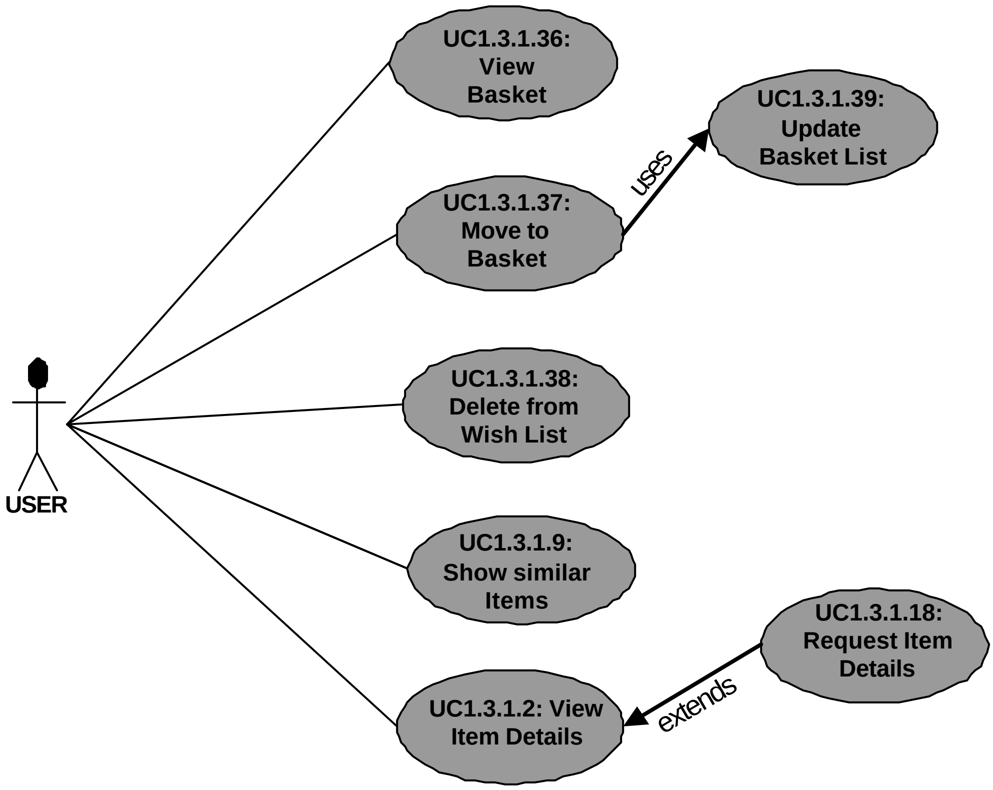

UC1.3.1.36: View Basket
UC1.3.1.39: Update Basket List
uses
UC1.3.1.37: Move to Basket
UC1.3.1.38: Delete from Wish List
USER
UC1.3.1.9: Show similar Items
UC1.3.1.18: Request Item Details
extends
UC1.3.1.2: View Item Details

Figure 27: Use Case Diagram – e-Business (wish list)

The above diagrams describe in detail the use cases of the E-Shop functionality of the REGNET e-Business System.

The diagrams shows the interactions of an end user with the shopping cart of the E-Shop system. The main aspect is the information exchange between the system and the end user. However, to complete a buying transaction functionality of other subsystems (e.g. procurement) is needed.

A typical buying transaction starts at the portal where the end user uses searches for a product of interest. If he/she selects products for buying, it will be included in the shopping cart and/or added to the wish list. These information can be administrated with the provided e-Business functionality. Besides that, the eBusiness subsystem must provide the opportunity to the end-user to select the way of payment and to accomplish the order.

At this point special care is given to the specific transactions, since that the user will provide to the system personal details that should not be published by no means. Therefore, special services will be supported by the e-Business subsystem in order to secure the relevant information.

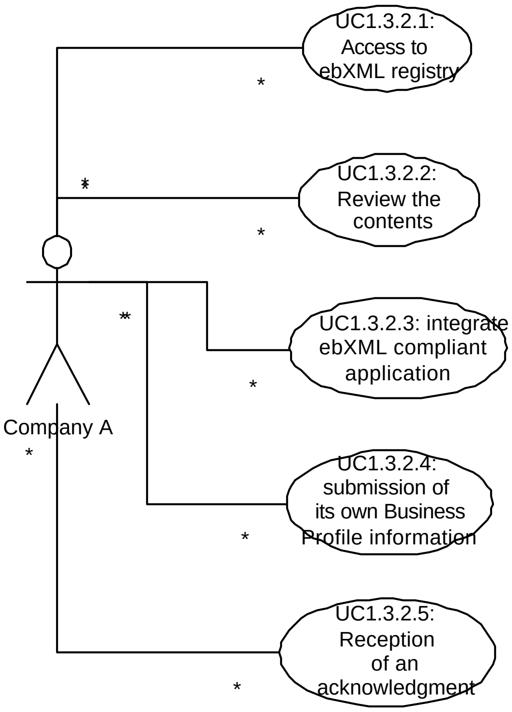

UC1.3.2.1: Access to ebXML registry
UC1.3.2.2: Review the contents
UC1.3.2.3: integrate ebXML compliant application
UC1.3.2.4: submission of its own Business Profile information
UC1.3.2.5: Reception of an acknowledgment
Company A
*

Figure 28: Use Case Diagram – e-Business (Registration to REGNET registry)

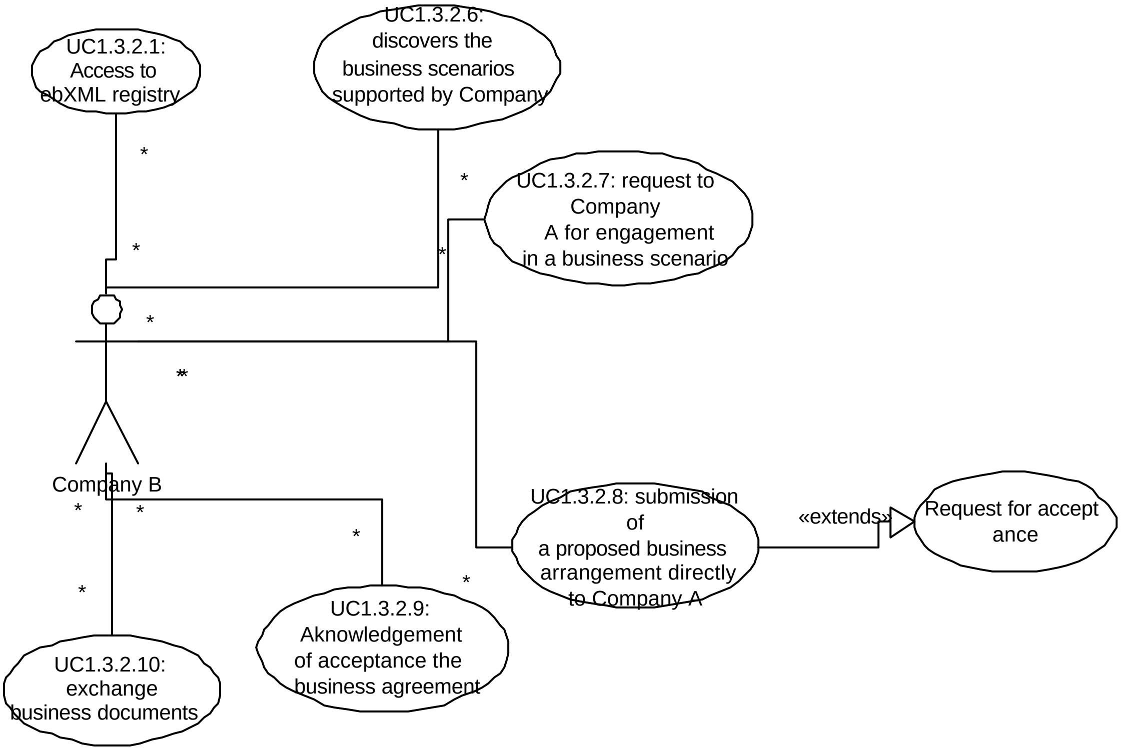

UC1.3.2.1: Access to ebXML registry
UC1.3.2.6: discovers the business scenarios supported by Company
UC1.3.2.7: request to Company A for engagement in a business scenario
Company B
UC1.3.2.8: submission of a proposed business arrangement directly to Company A
«extends»
Request for accept ance
UC1.3.2.9: Aknowledgement of acceptance the business agreement
UC1.3.2.10: exchange business documents
*

Figure 29: Use Case Diagram – e-Business (B2B)

The second part of functionality of the e-Business system deals with the Business to Business (B2B) communication between partners. The B2B functionality thereby relies on the framework defined by the ebXML specification, already well adopted and supported by the industry. A detailed description of ebXML can be found in the state-of-the-art.

UC Identifier	Use Case	Description
UC 1.3.1.1	Add Item	Add an Item (Product) in the Shopping Cart
UC 1.3.1.2	View Item Details	Present to the User all the relevant information of the product
UC 1.3.1.3	Remove Item	Remove Item from the shopping cart
UC 1.3.1.4	Move Item to Wish List	The user doesn’t want to include the product in the current order and stores the product in the wish list for a future order
UC 1.3.1.5	Change Item Quantity	Change Item Quantity in the shopping cart
UC 1.3.1.6	Check Out	Finalisation of the order by the end-user
UC 1.3.1.7	Update	Update the content of the shopping cart
UC 1.3.1.8	View Item availability	The system should check if the product is available on the store and in which quantity
UC 1.3.1.9	Show similar items	The shopping cart must inform the user about similar products with the products that he/she has stored in the shopping cart
UC 1.3.1.10	Save Item	Save an item.
UC 1.3.1.11	Calculate Total	The shopping cart calculates the total amount of money that the clients have to pay in order to fulfil the transaction.
UC 1.3.1.12	Update Wish List	When a product is stored in the wish list, the system updates the list.
UC 1.3.1.13	Check data	Check data of the shopping cart.
UC 1.3.1.14	Save data	Save Data of the shopping cart.
UC 1.3.1.15	Request Similar Items	Request Similar Items
UC 1.3.1.16	Move Item to Permanent Shopping List	Move Item to Permanent Shopping List.
UC 1.3.1.17	Request Item Availability	Request Item Availability.
UC 1.3.1.18	Request Item Details	Request Item Details.
UC 1.3.1.19	View Order History	View Order History
UC 1.3.1.20	Review Order	Before the final transaction is made, the client can make a final check.
UC 1.3.1.21	Cancel Order	Cancel Order
UC 1.3.1.22	Place Order	The client releases the order.
UC 1.3.1.23	Request user Info	Request user Info from the user’s profile
UC 1.3.1.24	Request Item Info	Request Item Info.

UC 1.3.1.25	Make payment	Make the payment.
UC 1.3.1.26	Make e-payment	If the user decides to pay with credit card using the online payment, this use case is invoked.
UC 1.3.1.27	Make check payment	Make check payment.
UC 1.3.1.28	Make cash on delivery Payment	Make cash on delivery Payment.
UC 1.3.1.29	Provide Credit Card Data	To provide the data needed for the online payment.
UC 1.3.1.30	Check Credit Card Data	Check Credit Card Data
UC 1.3.1.31	Request Credit Card Validation Info	Request Credit Card Validation Info
UC 1.3.1.32	Produce Certificate	Produce Certificate to prove the transaction and the order
UC 1.3.1.33	Provide check data	Provide check data
UC 1.3.1.34	Request Provider Payment Info	Request Provider Payment Info
UC 1.3.1.35	View Wish List	The client may access and manage his wish list.
UC 1.3.1.36	View Basket	The client may return to the basket.
UC 1.3.1.37	Move to Basket	A product may be transferred from the wish list to the basket
UC 1.3.1.38	Delete from Wish List	Delete from Wish List
UC 1.3.1.39	Update basket list	Update the basket list
UC 1.3.2.1	Access to ebXML REGNET registry	Access to ebXML REGNET registry
UC 1.3.2.2	Review the Contents	Review the Contents of the registry
UC 1.3.2.3	Integrate ebXML compliant application	Integrate ebXML compliant application
UC 1.3.2.4	Submission of Business Profile Information	Submission of Business Profile Information

UC 1.3.2.5	Reception of an acknowledgement	The company receives an acknowledgement according to its registration.
UC 1.3.2.6	Discover Business Scenarios	The business scenarios supported by a company are discovered.
UC 1.3.2.7	Request for engagement in business scenario	A company is invited to take part in a business scenario.
UC 1.3.2.8	Submission of Business arrangement	The submitter sends a proposed business arrangement to its partner company.
UC 1.3.2.9	Acknowledgement of the acceptance	The partner company sends an acknowledgement of the business agreement.
UC 1.3.2.10	Exchange Business Documents	Exchange Business Documents

Table 31: Use Case Description – e-Business

5.4.4 Cultural Heritage data management

5.4.4.1 Repository management

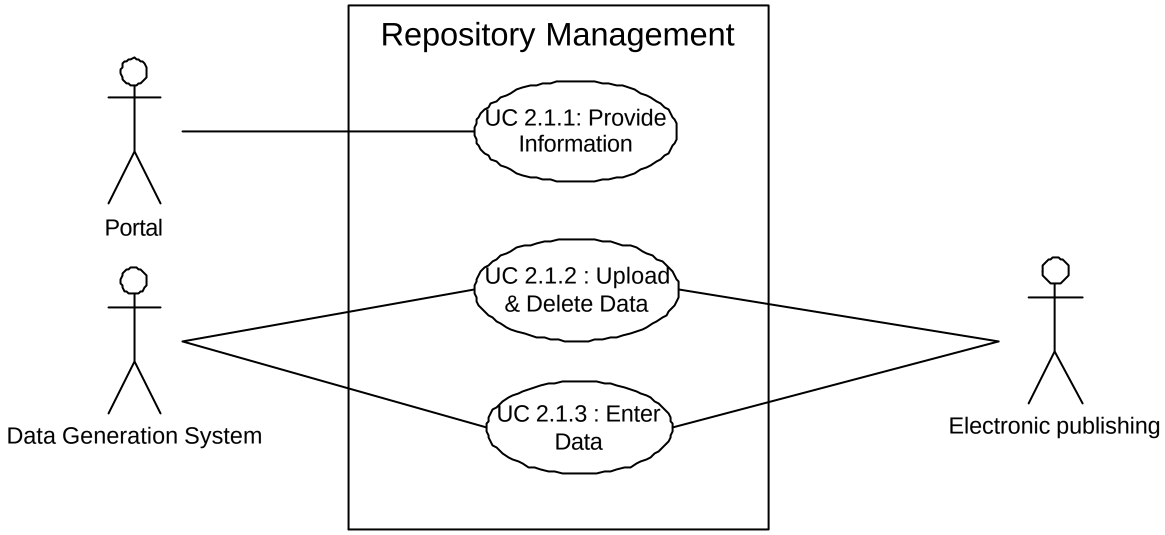


Figure 30: Use Case Diagram – Repository Management

UC Identifier	Use Case	Description
UC 2.1.1	Provide information	Present data to user interface
UC 2.1.2	Upload / delete data	Data management due to Data Generation or Electronic Publishing request
UC 2.1.3	Enter data	Data acquisition due to Data Generation or Electronic Publishing request

Table 32: Use Case Description – Repository Management

5.4.4.2 Reference System

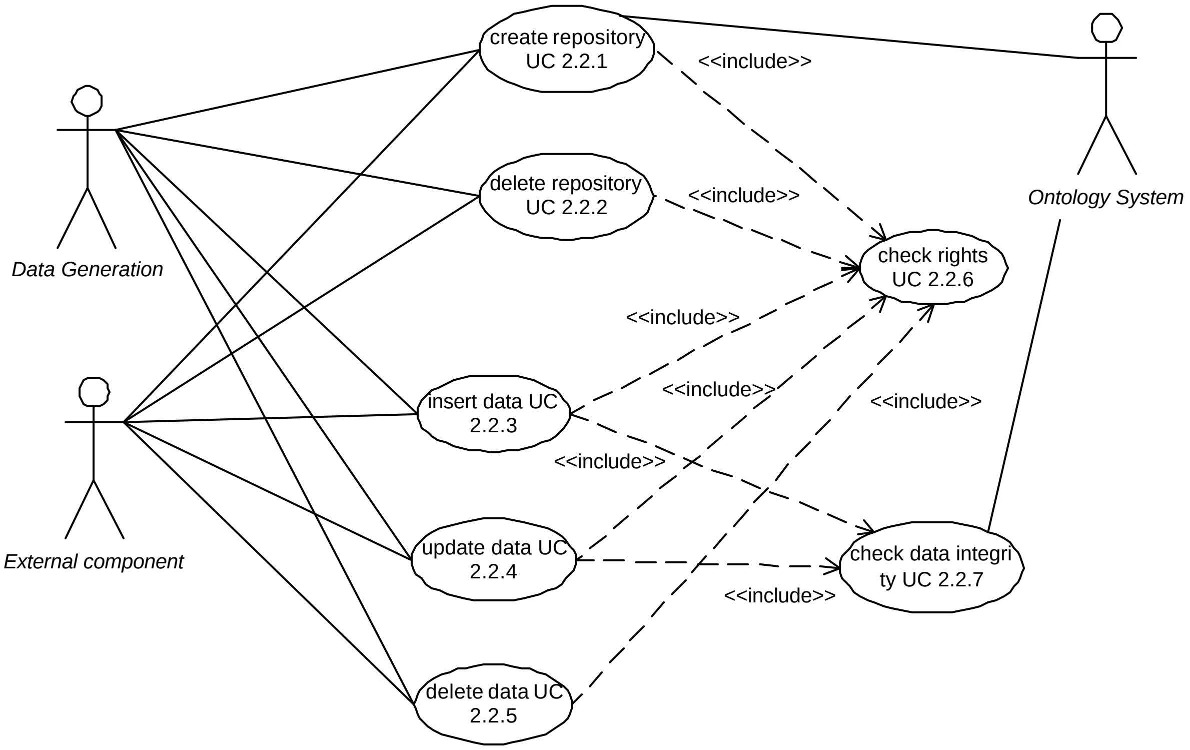


Figure 31: Use Case Diagram – Reference System (data management)

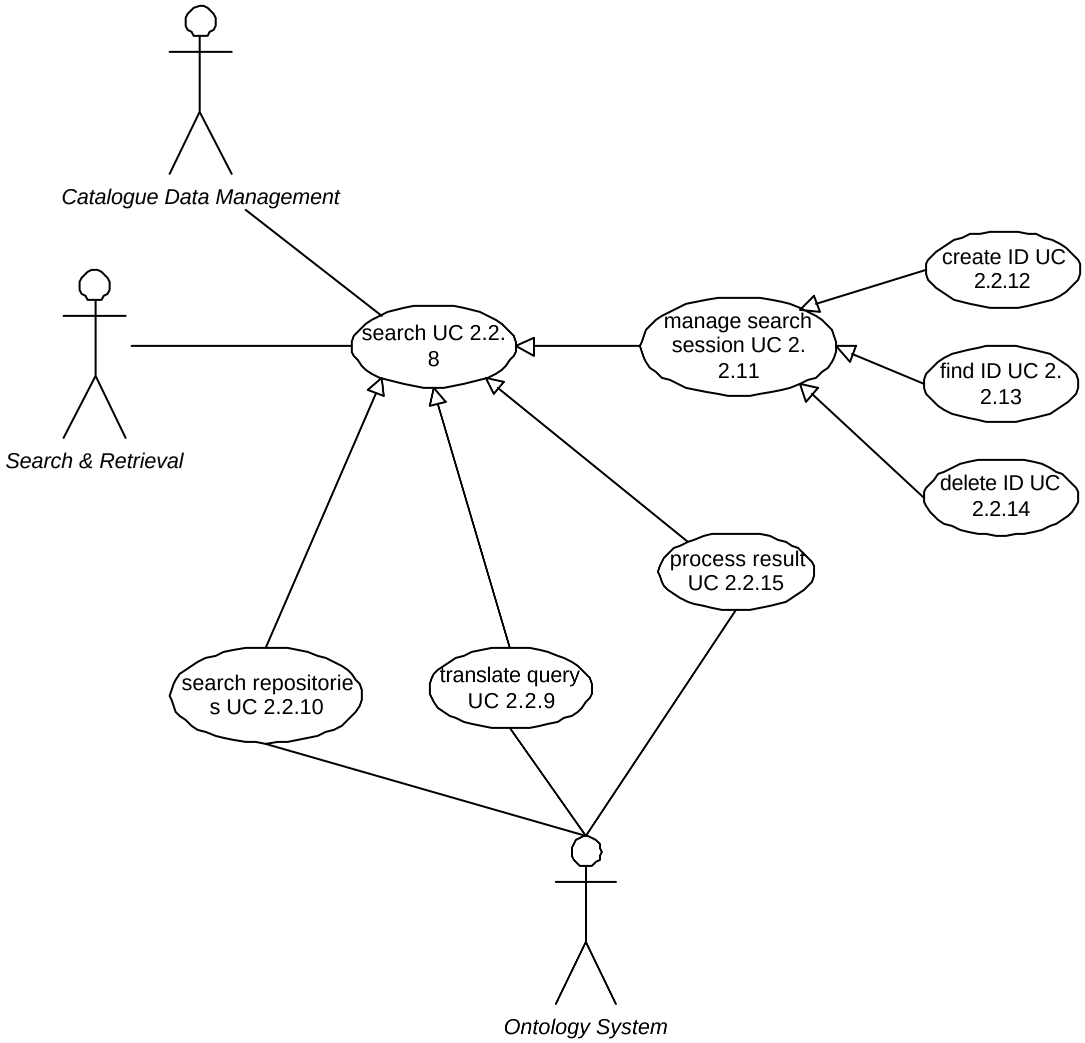


Figure 32: Use Case Diagram – Reference System (search)

UC Identifier	Use Case	Description
UC 2.2.1	Create repository	A new repository is created (all data needed to define a repository is needed)
UC 2.2.2	Delete repository	delete repository
UC 2.2.3	Insert data	insert data into repository (needs UC 2.2.7)
UC 2.2.4	Update data	update data in repository (needs UC 2.2.7)
UC 2.2.5	Delete Data	delete data in repository
UC 2.2.6	Check rights	check rights (is request source allowed to create, insert, update, delete?)
UC 2.2.7	Check data integrity	check data integrity for inserts and updates (check whether the data conforms to a DTD/XML-Schema.
UC 2.2.8	Search	search: provide search and retrieval functionality for clients (following the Z39.50 standard)

UC 2.2.9	Translate query	translate query: in further versions the Reference System might be able to handle other query formats than in Z39.50 format. Therefore a query translator will be used.
UC 2.2.10	Search repositories	query specific repository: distribute the search requests to the different repositories. For each database system included in the CH data management node a interface has to be provided
UC 2.2.11	Manage search session	Search session and result set management according to needs for the Z39.50 protocol
UC 2.2.12	Create ID	Create new ResultSetId and initialise all necessary Z39.50 states
UC 2.2.13	Find ID	Fnd ResultSetId and status on subsequent sort/present requests.
UC 2.2.14	Delete ID	Delete ResultSet due to logout, timeout
UC 2.2.15	Process result	Process and build results according to specifications in the request

Table 33: Use Case Description – Reference System

5.4.5 e-Business Data Management

Good data management is crucial for e-Business success. On-line enterprises can use data to attract customers to web sites, encourage them to buy products and services, and to build up customer loyalty. By collecting and using clickstream data, enterprises can implement web personalization. Web personalization techniques adopt the shopping and browsing experience for each user—obviously a powerful way to satisfy customers and keep them returning to a web site. E-Business data management also allows enterprises to implement customer self- service via the web, another opportunity to make the web experience more satisfying. Effective data management is also necessary for B2B ventures as they strive to collaborate with partners and suppliers.

B2B enterprises will be challenged to become "open books," freely sharing up-to-the-minute data with other organisations in their market space. This requirement for near instantaneous diffusion of information across the marketplace will favour competitors who have systems in place to both, disseminate and collect large amounts of data.

5.4.5.1 Product catalogue management

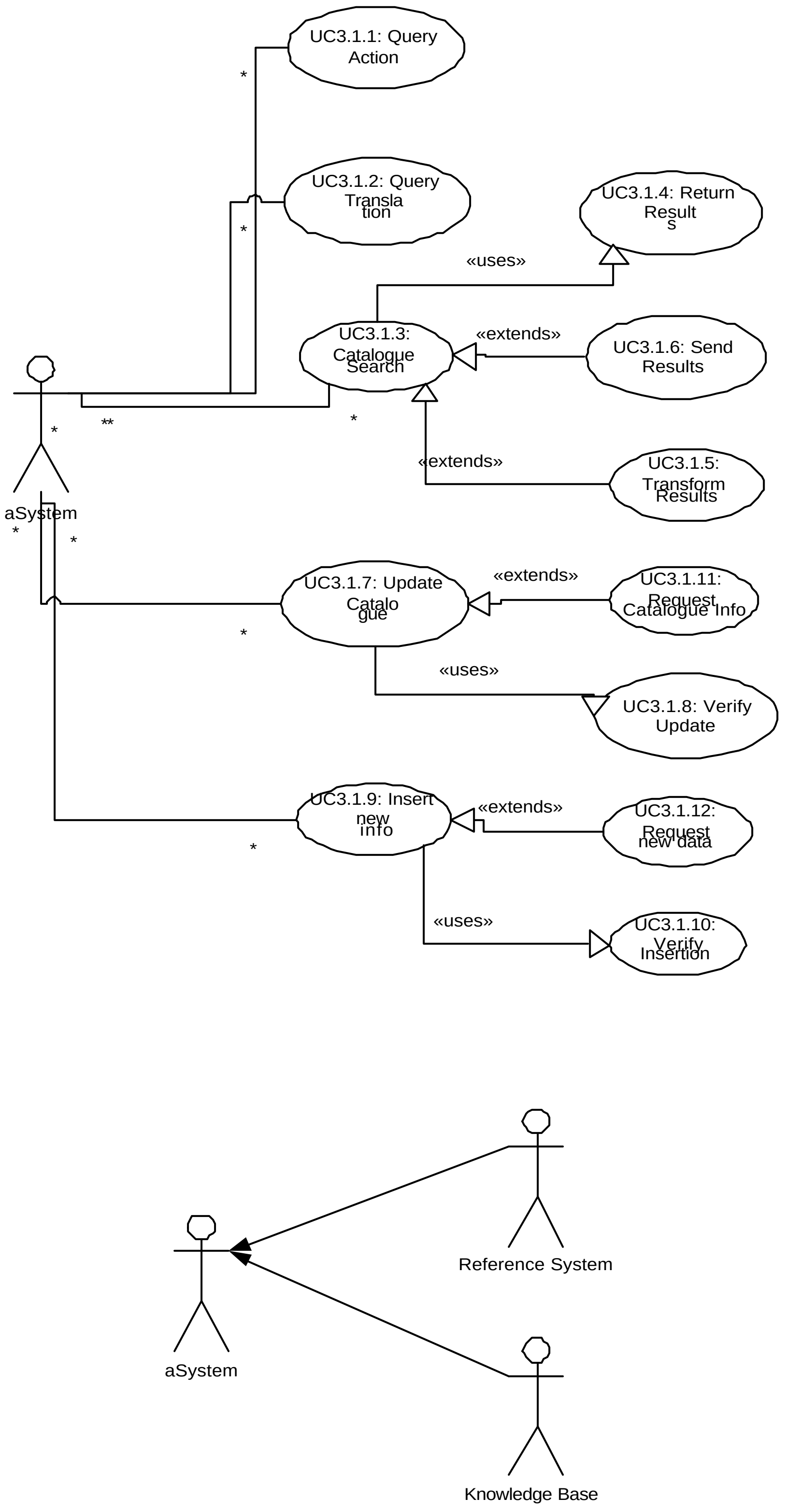


Figure 33: Use Case Diagram – Product Catalogue Management

The Product Catalogue Management is part of the Business Logic sector of the REGNET System. It is important to note that the Product Catalogue Management is responsible for the management of the catalogues of the different museums.

The interesting point is that the PCM is being triggered only by two other subsystems of the REGNET system, which are the Knowledge Base subsystem and the Reference System. The output of the results of the PCM are the Search and Retrieval, Repository Management, Reference System, Data Generation and e-Business subsystems. In all cases the PCM is the intermediate component for the final answer in all aspects that are relevant to the several digital catalogues.

In the above diagram the main functionalities of the PCM are represented and the relevant use cases.

The PCM gets input from the KB and the RS. The PCM decides about the function that is needed and perform the relevant steps of the functionality. The end of each operation provides the output of the PCM to the correspondent subsystem.

UC Identifier	Description
UC 3.1.1	Introduction of the query to the PCM and to the query translator
UC 3.1.2	Refine the query to the internal needs of the PCM
UC 3.1.3	Define the task that will be implemented (e.g. search the catalogue, update info etc.)
UC 3.1.4	Search the product catalogues according to the query
UC 3.1.5	Decide if the results satisfy the terms of the query
UC 3.1.6	Preparation of the results to the desired format
UC 3.1.7	Update the information for a product within the catalogue
UC 3.1.8	Verify the update to the rest of the subsystems
UC 3.1.9	Insert new products or new information within the products catalogue
UC 3.1.10	Verify the insertion to the rest of the subsystems

Table 34: Use Case Description – Product Catalogue Management

5.4.5.2 Procurement and Delivery

This sub-system can be split into two parts :

- A **MarketPlace** which allows the REGNET content providers trade with "dedicated" suppliers.

  It is a wide marketplace : the suppliers are SMEs*, manufacturers of museum shop products, of fine arts supplies, or they can be services providers as a transport company, a museum, ...

- An **e-Shop Logistics** which includes the parcels preparation according to the order, their delivery, and the invoices management.

5.4.5.2.1 Procurement

In order to allowing the REGNET Partners to have the independence of choice, we consider the procurement trough the **MarketPlace** system.

A MarketPlace enables multiple buyers and sellers to buy and sell to each other at a fixed/dynamic price in real time and at low transaction costs, while building long-lasting relationships.

The main functions of a marketplace are:

- **Match buyers and sellers:**
  - Disseminate product information
  - Search for buyers and sellers
  - Price discovery

* SME : Small and Medium sized Enterprise

- **Reduce purchasing costs:**
  - Reduce costs of goods via aggregation
  - Reduce transaction costs
- **Manage infrastructure:**
  - Build trust
  - Govern rules of exchange, monitoring, dispute resolution, ….
  - Collect market statistics for market participants
  - Absorb technological risks

Two solutions are possible:

1. A unique wide MarketPlace which connects all the REGNET buyers with all the suppliers.
2. Each geographical site has a specific MarketPlace with its own buyers and suppliers.

The main advantage of the first solution is to connect a larger community : more choice for the buyers, more potential sale for the suppliers.

But in the perspective that the REGNET Consortium is composed by small and medium sized Cultural Heritage Institutions which would trade with Europeans SMEs, this solution can exclude some participation: a small size supplier hasn’t always the means of trading throughout the Europe (insufficient production, logistic,...)

The buyer and the supplier will be induced to arrange deal terms (shipping, timeframe, payment terms,...), the negotiation will be easier if the traders have the same size. Moreover, even if the official language of the MarketPlace has to be English, the traders will be induced to communicate (arrange terms, complaints, ...), it is easier in your own language.

For these reasons, we considered the second solution : there is **a MarketPlace by CSC**.

A buyer could trade on only one MarketPlace, the MarketPlace he is attached to.

But, a supplier could present his catalogue on several REGNET MarketPlaces.

Actors

The actors of the marketplace are:

- **The Buyers Group:** the marketplace is reserved for the REGNET partners : museums, archives, artists community, REGNET e-shop, ...

  Each buyer as REGNET partner is attached to just only one CSC, so to just only one marketplace.

- **The Suppliers Group:** the marketplace is reserved for the registered suppliers.

  A supplier manages a products or services catalogue. He can present his catalogue in different REGNET MarketPlaces.

- **The administrator:** he manages all the REGNET identified users : partners, suppliers, consumers. In the marketplace frame, he has to pre-qualify all the suppliers to ensure quality and security.

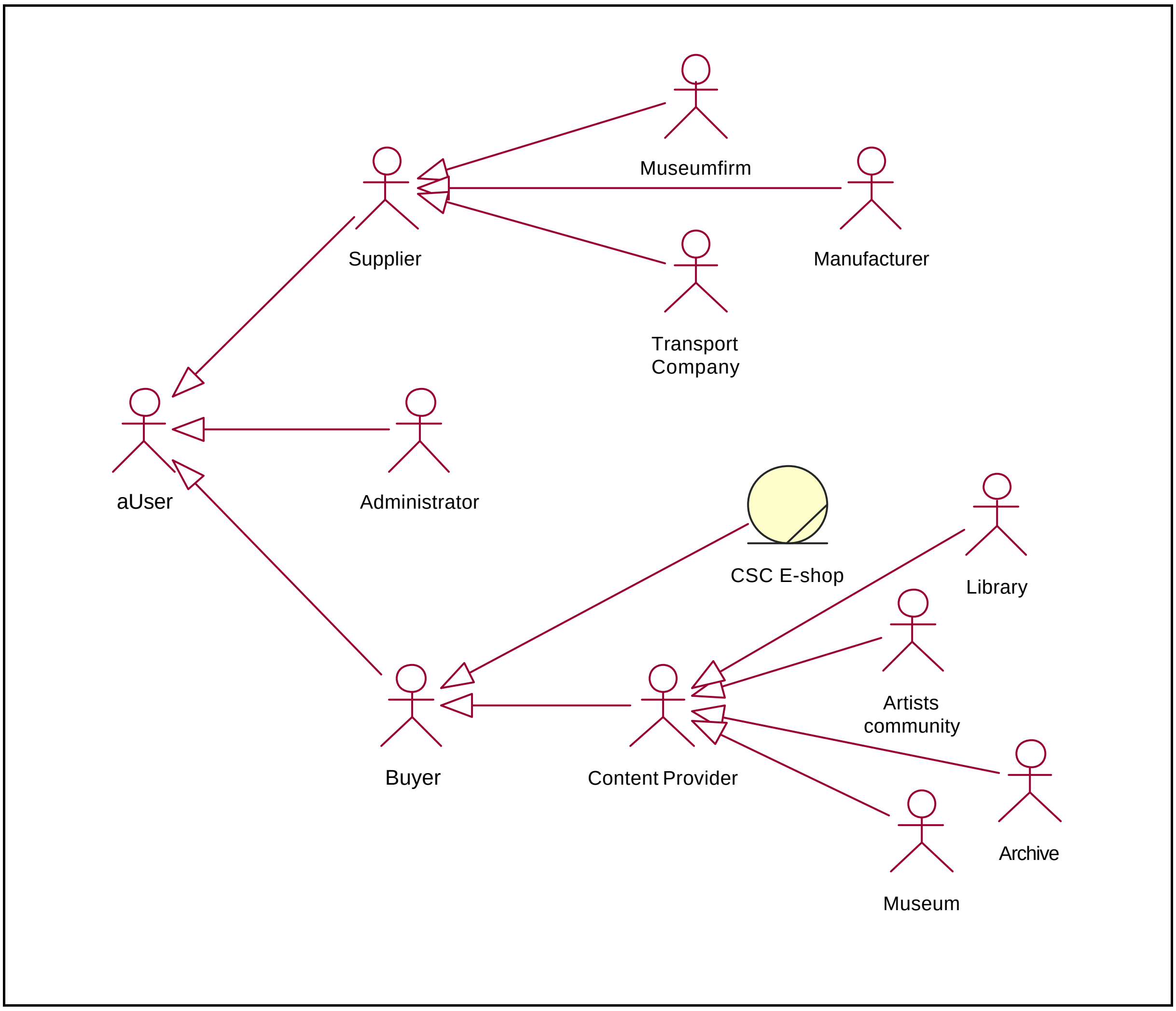


Figure 34: Use Case Diagram – Procurement (Actors hierarchy)

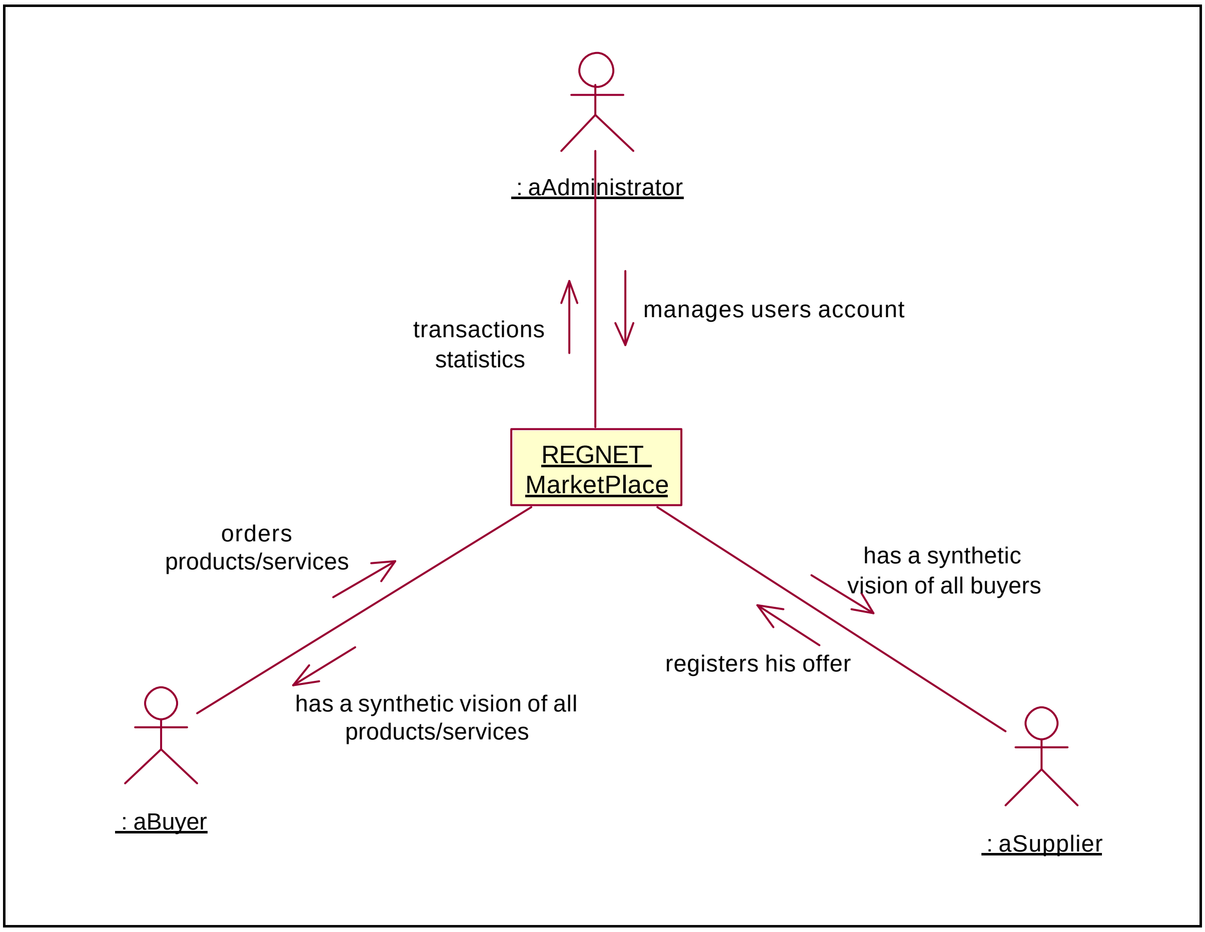


Figure 35: Use Case Diagram – Procurement (Context)

Use Cases

We defined 4 main groups of use Cases:

- **Access Group:** functionalities of presentation and registration to the MarketPlace .
- **Catalogue Group:** functionalities of the management of the MarketPlace Catalogue
- **Order Group:** functionalities and the steps of the progress cycle of an order.
- **History/Reports Group:** functionalities of measurements and improvement of the MarketPlace

Access Group:

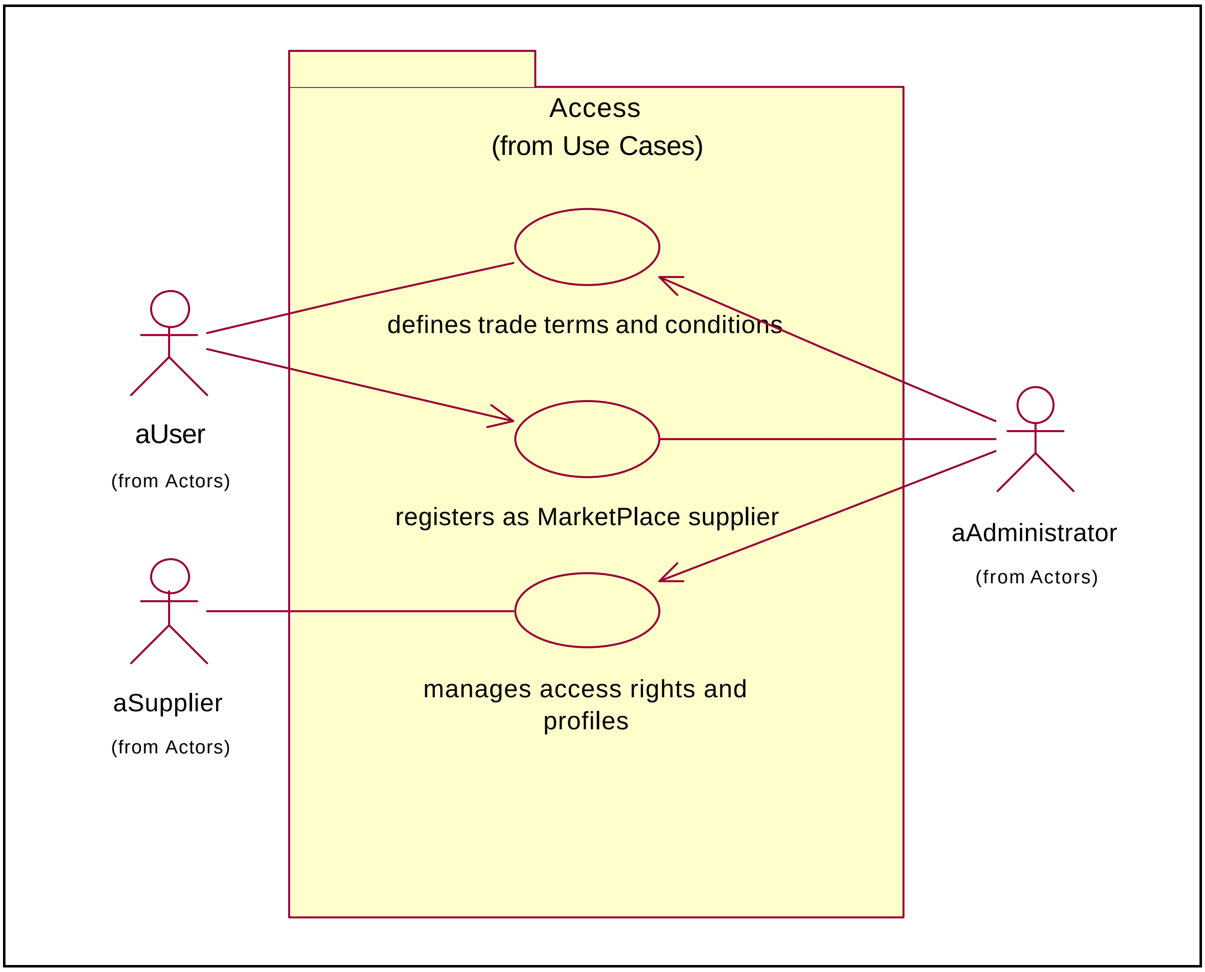

Access
(from Use Cases)
defines trade terms and conditions
registers as MarketPlace supplier
manages access rights and profiles
aUser
(from Actors)
aSupplier
(from Actors)
aAdministrator
(from Actors)

Figure 36: Use Case Diagram – Procurement (Access Group)

Ident	Use case	Descriptions
UC 3.2.1	defines trade terms and conditions	the administrator defines according with the REGNET Partner the terms and conditions of the MarketPlace. Every User can make oneself acquainted with it
UC 3.2.2a	registers as MarketPlace supplier	the potential supplier has to fill out a form with his personal information and trades references
UC 3.2.2b	manages access rights and profiles	the administrator verifies references supplier, he can accept or not the supplier's candidature.

Table 35: Use Case Description – Procurement (Access Group)

Catalogue Group:

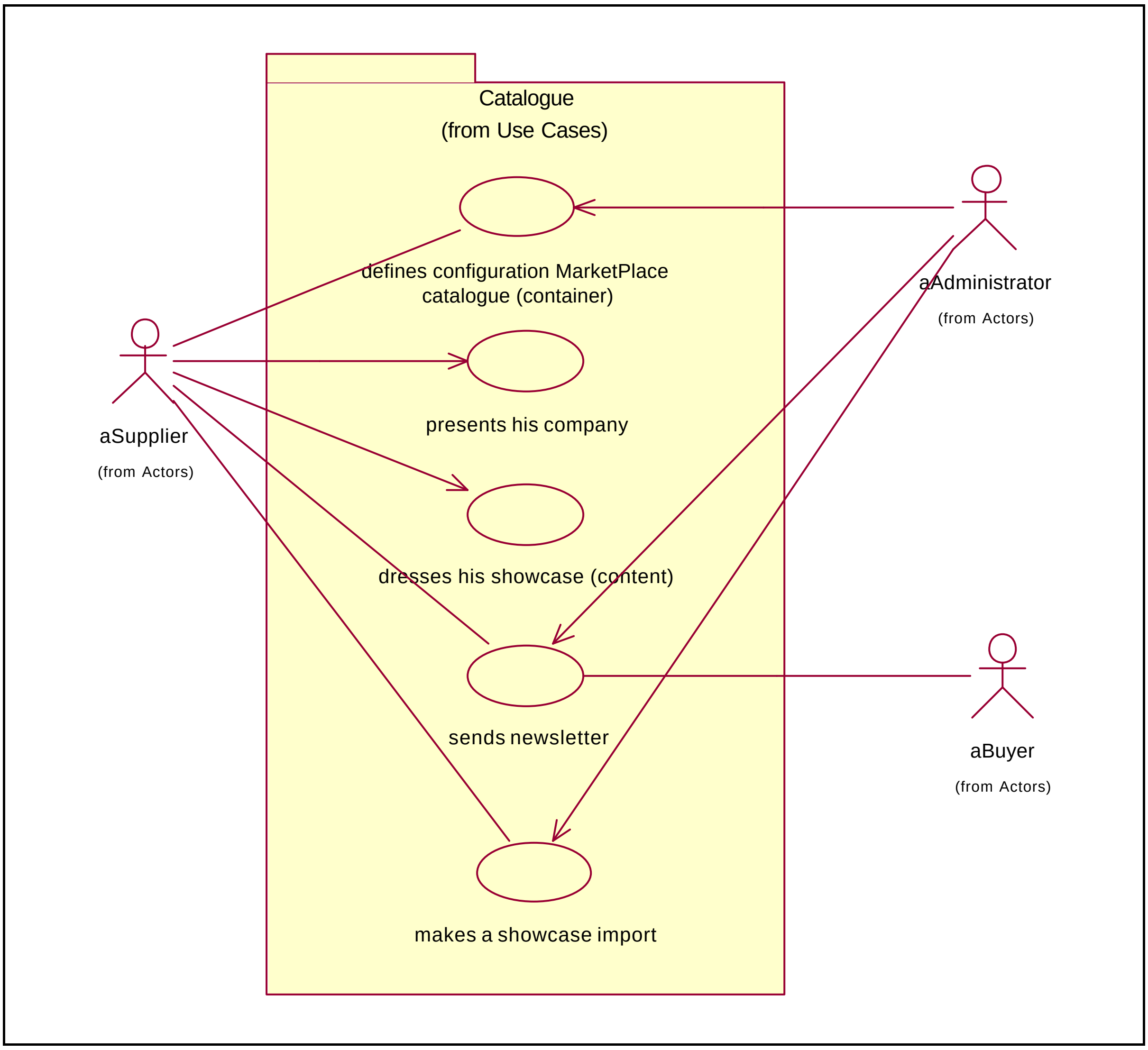


Figure 37: Use Case Diagram – Procurement (Catalogue Group)

UC Identifier	Use Case	Description
UC 3.2.3	defines configuration MarketPlace catalogue (container)	the administrator defines the catalogue information : each offered product or service has to be described by some information: product name, category (choice among a list defined by the administrator, a supplier or a buyer can propose a new category), description (with an image), size, weight, supplier name

UC 3.2.4a	presents his company	The supplier gives information about his company, these information can be modified anytime
UC 3.2.4b	dresses his showcase (content)	the Supplier has to manage his showcase : he complete catalogue content by the products and services he offers on marketplace. He can modify his showcase anytime.
UC 3.2.5	sends newsletter	The administrator informs the MarketPlace community of the coming of a new supplier or buyer, of the new offer of a product or service.
UC 3.2.4c	makes a showcase import	When a supplier wants to offer his items to a new REGNET marketplace, the administrator can import the supplier's showcase from a marketplace to an another.

Table 36: Use Case Description – Procurement (Catalogue Group)

Order Group:

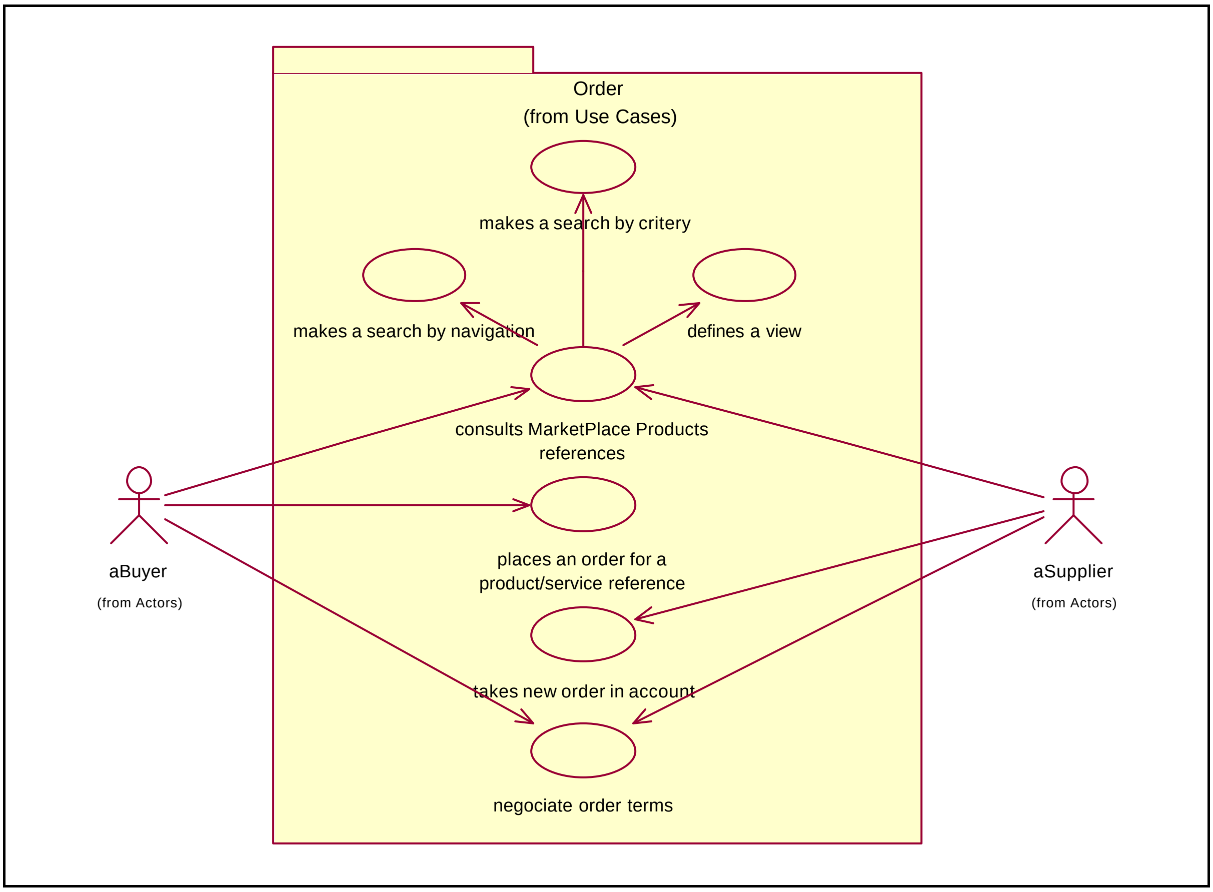

Order
(from Use Cases)
makes a search by critery
makes a search by navigation
defines a view
consults MarketPlace Products references
places an order for a product/service reference
takes new order in account
negociate order terms
aBuyer
(from Actors)
aSupplier
(from Actors)

Figure 38: Use Case Diagram – Procurement (Order Group)

UC Identifier	Use Case	Description
UC 3.2.6	consults MarketPlace Products references	a Buyer or a Supplier of the MarketPLace can consult all the products and services proposed on the MarketPlace
UC 3.2.6a	makes a search by navigation	The buyer or the supplier can navigate on the catalogue by product, category, vendor
UC 3.2.6b	makes a search by criteria	The buyer or the supplier can search a product or a service by giving a keyword.
UC 3.2.6c	defines a view	The buyer can predefine for example, the list of the suppliers he wants to consult the products, or the list of his preferred categories of products
UC 3.2.7a	places an order for a product/service reference	The buyer order a product or a service. The order is automatically sent to the product/service's supplier
UC 3.2.7b	takes new order in account	The supplier consults his news orders
UC 3.2.7c	negotiate order terms	The buyer and the supplier arrange the term of the order : shipping, timeframe, payment terms

Table 37: Use Case Description – Procurement (Order Group)

History/Reports Group:

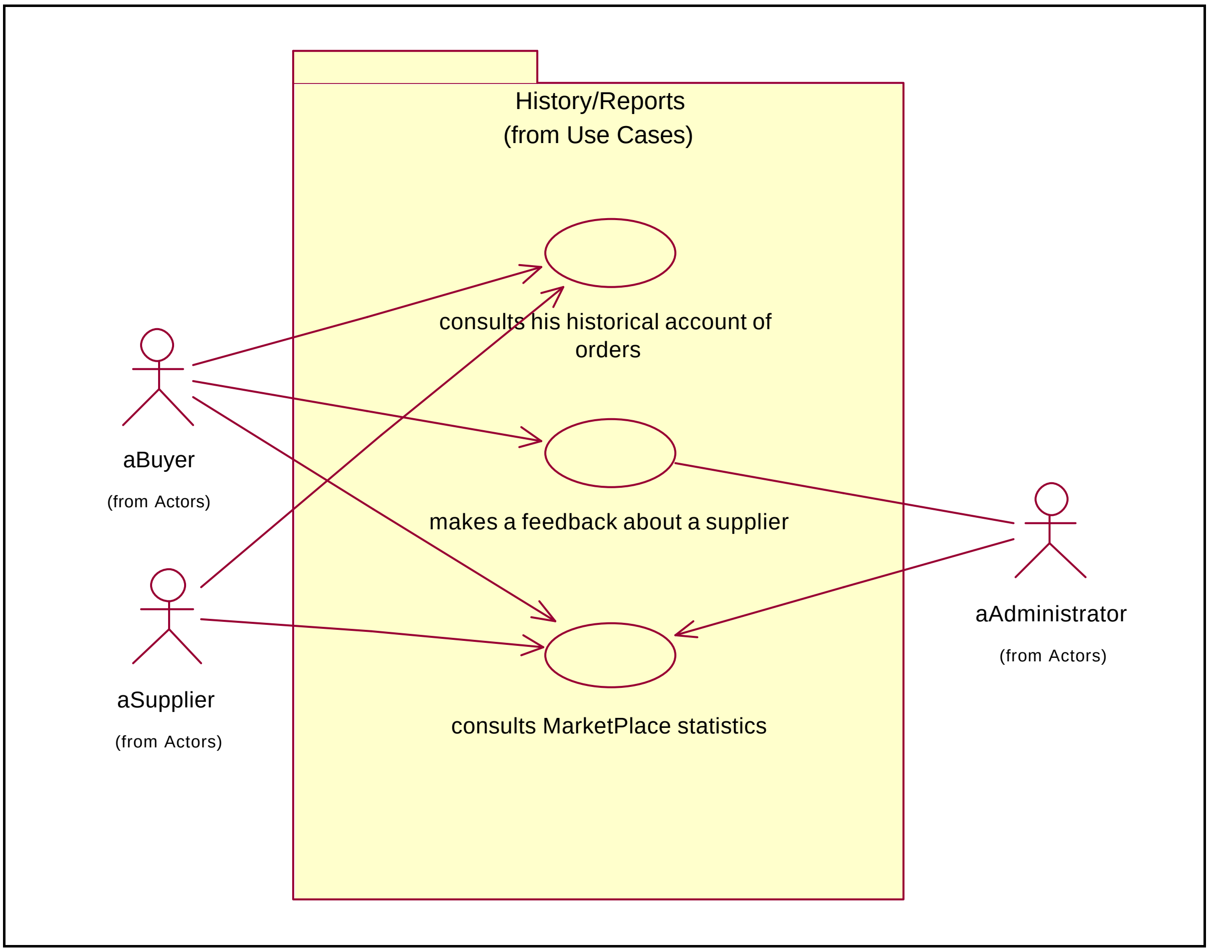

History/Reports
(from Use Cases)
consults his historical account of orders
makes a feedback about a supplier
consults MarketPlace statistics
aBuyer
(from Actors)
aSupplier
(from Actors)
aAdministrator
(from Actors)

Figure 39: Use Case Diagram – Procurement (History/Reports Group)

UC Identifier	Use Case	Description
UC 3.2.8	consults his historical account of orders	The buyer and the supplier can consults all the orders with regard to them.
UC 3.2.9	makes a feedback about a supplier	A Buyer can gives an appreciation, a note about a supplier
UC 3.2.10	consults MarketPlace statistics	The system provides some statistics : top 10 of products, the best supplier, ...

Table 38: Use Case Description – Procurement (History/Reports Group)

5.4.5.2.2 Delivery Invocation

As exposed in the paragraph about the organisation infrastructure, we have considered that each CSC manages just one e-Shop.

In this framework, the CSC has to manage the delivery and the invocation related to its e-Shop.

Actors

The actors of the delivery and invocation services are:

- **The Consumer**: He is the customer of a REGNET e-Shop: he ordered e-Shop items. He has to pay the delivered items. The customer wants to track his parcel.
- **The Delivery and Invocation Group**:
  The Shipping Clerk prepares the parcels; he collects the items related to each order, he packs them, he weighs the parcel in order to calculate the shipping rate, he prints the invoice and the customs formalities for the export and encloses with the parcel, he labels the parcel, and puts it on the collection quay
  The Logistic Responsible contacts the shipping companies and negotiates contracts (e.g. : collection conditions), he updates the shipping rates (or manually or by downloading), he manages the stocks of packaging.
  The Return Clerk registers the returns, he has to verify the return delay isn't missed, the items condition, he notes the return's reason.
  He puts back the item in stock if it isn't damaged**.**
- **The e-Shop Accountant** refunds the returned parcel.
- **The REGNET administrator** manages all the REGNET identified users : partners, workers, and consumers. (ONTOLOGY).
  In the Delivery frame, he has to define all the delivery workers : packer, logistics responsible, accountant, return service.
- **Others actors**
  The Shipping company : It is an external enterprise (a REGNET supplier) which takes charge of parcels transport and delivery (UPS, FedEx, Chronopost, ...)
  The Banking System : it is the financial institution which verifies the customer's solvency, consequently the payment is accepted or refused.

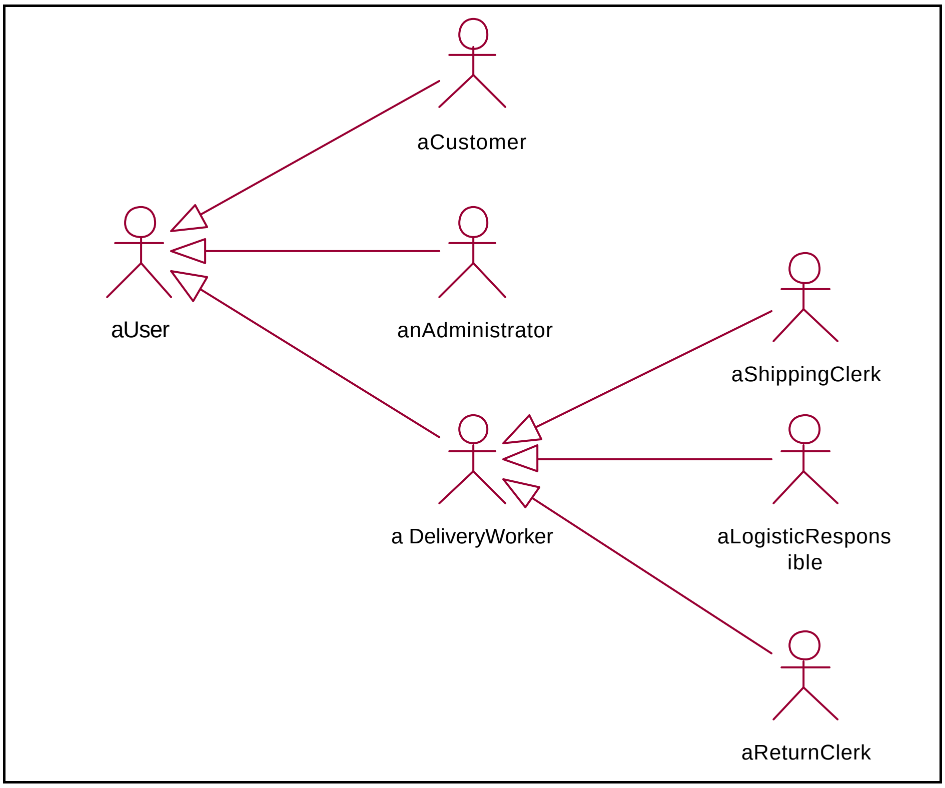


Figure 40: Use Case Diagram – Delivery (Actors hierarchy)

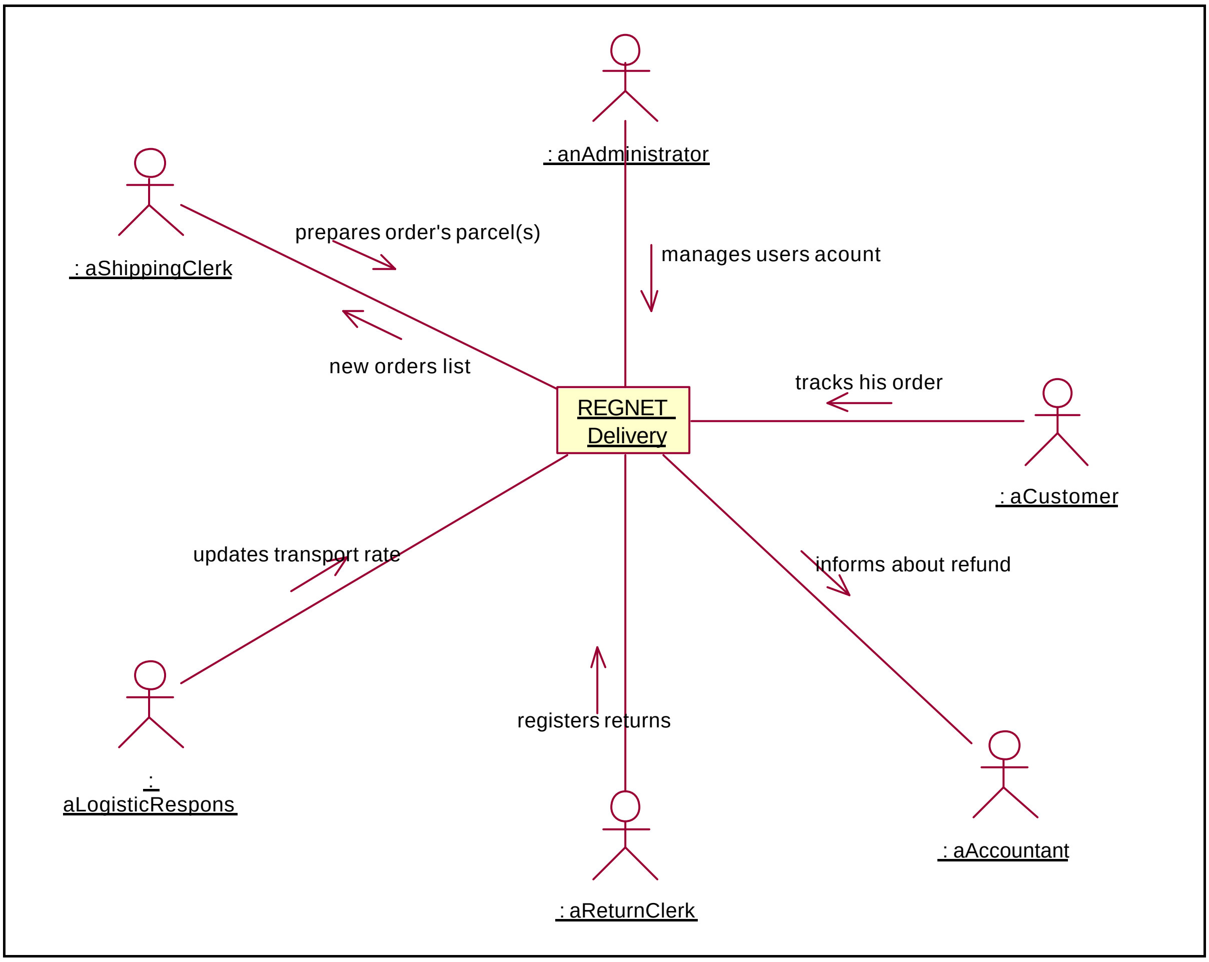


Figure 41: Use Case Diagram – Delivery (Context)

Use Cases

We defined 4 main group of use Cases:

- **Access Group:** functionality of the management of delivery and invocation department account.
- **Transport Costs Group:** functionality of the management of the shipping companies services.
- **Shipping Group:** functionality of the order’s parcel preparation, shipping and tracking.
- **Return Group:** functionality of the management of returned parcels.

Access Group:

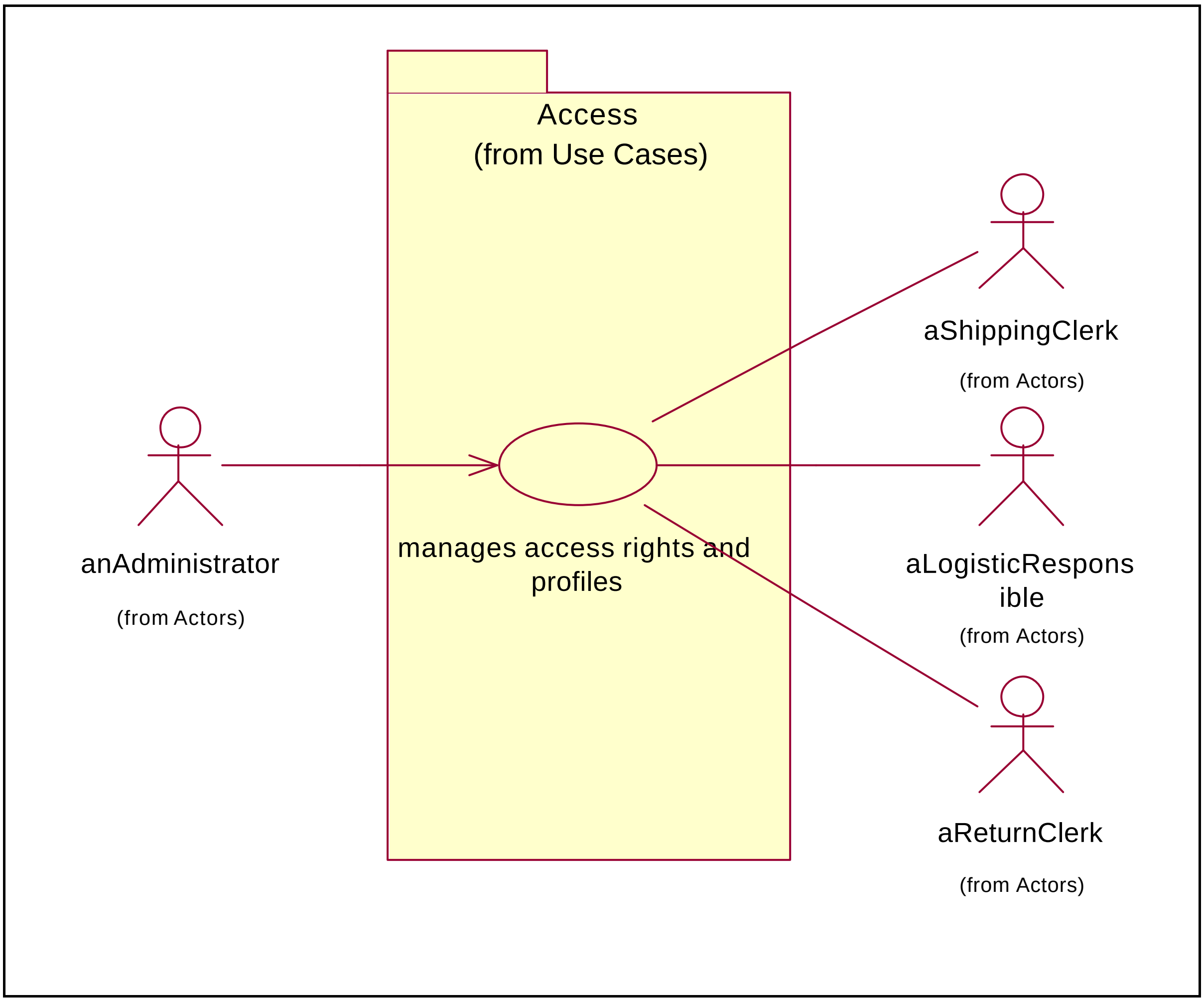


Figure 42: Use Case Diagram – Delivery (Access Group)

UC Identifier	Use Case	Description
UC 3.2.11	manages access rights and profiles	The administrator creates and deletes the account of each worker of delivery and invocation structure.

Table 39: Use Case Description – Delivery (Access Group)

Transport Costs Group:

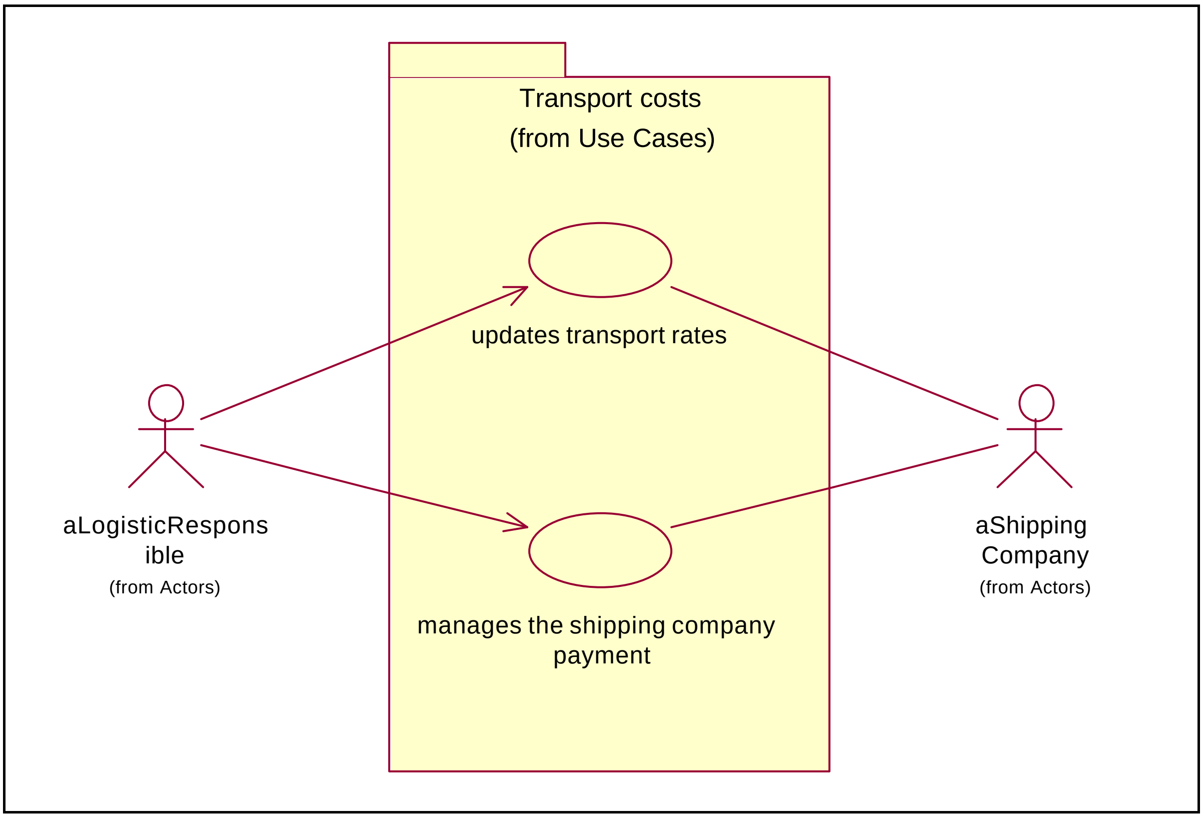


Figure 43: Use Case Diagram – Delivery (Transport Costs Group)

UC Identifier	Use Case	Description
UC 3.2.12	updates transport rates	The logistics responsible updates the transport rates of the shipping company (manually or by files downloading).
UC 3.2.13	verifies the transport's invoice	The logistic responsible verifies that the shipping company's invoice corresponds to the parcels collections and pays it.

Table 40: Use Case Description – Delivery (Transport Costs Group)

Shipping Group:

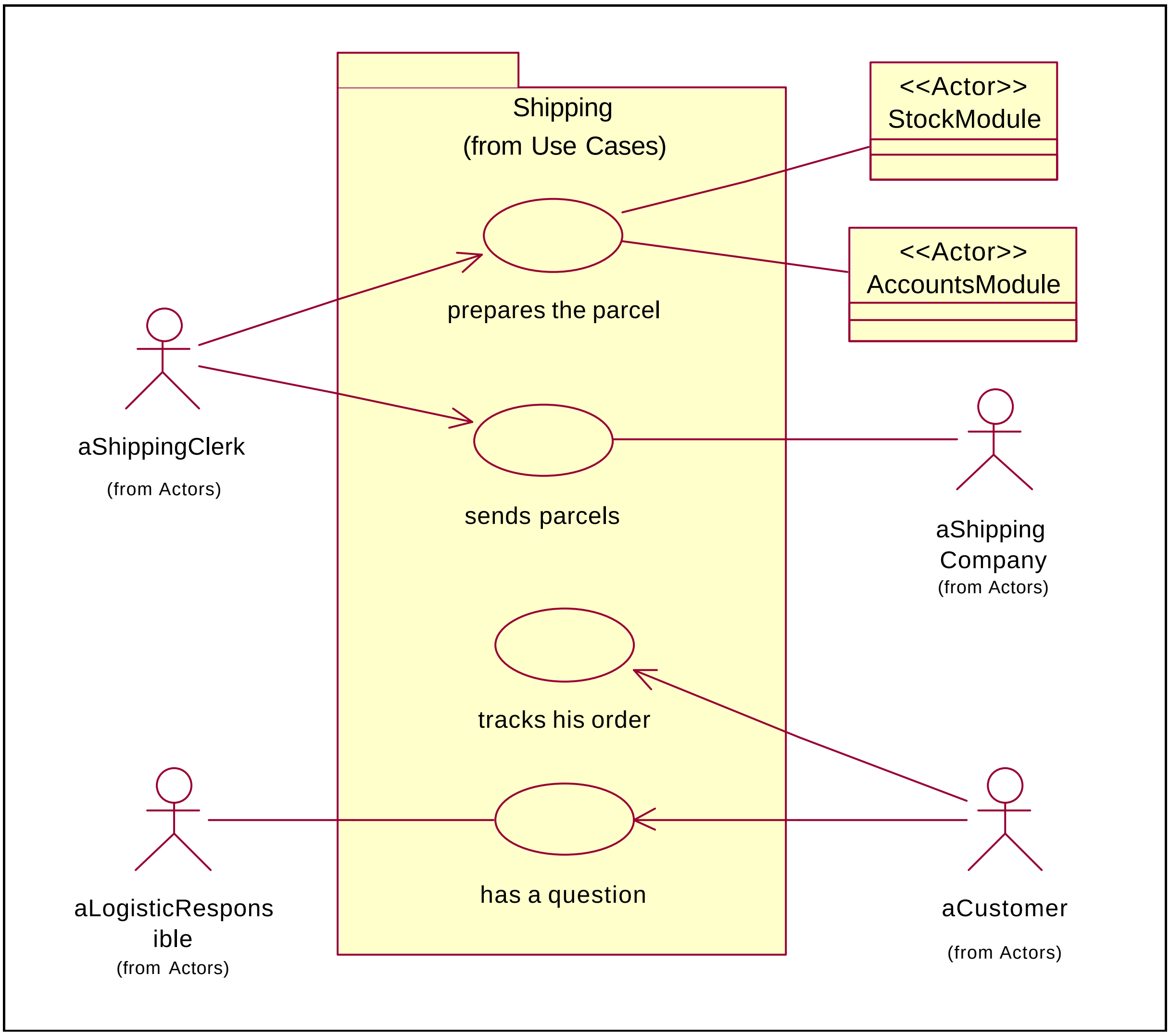


Figure 44: Use Case Diagram – Delivery (Shipping Group)

UC Identifier	Use Case	Description
UC 3.2.14	prepares the parcel	The shipping clerk selects a new order to be taken in charge. He verifies that all ordered items are available, if it isn't : the order "is waiting for item", if it is : the order is "packed", the shipping clerk prepares the parcel(s). He encloses the invoice in the parcel. He completes the waybill. The stock is updated and the invoice is sent to the Accounts module.

UC 3.2.15	sends the parcel	The shipping clerk puts the parcel on the collection quay. The shipping company takes in charge the parcel (the collection can be done at request or can be planned) The shipping company takes in charge the pick up, the transport and the delivery of the parcels. The shipping company must be able to give the proof of delivery.
UC 3.2.16	tracks his order	The customer can track his order : He wants to know if his order "is waiting for item" or "sent". Some shipping companies proposes tracking services with a tracking number.
UC 3.2.17	has a question	The customer needs a precision about his order, or has a problem as damaged parcel, not received parcel.

Table 41: Use Case Description – Delivery (Shipping Group)

Return Group:

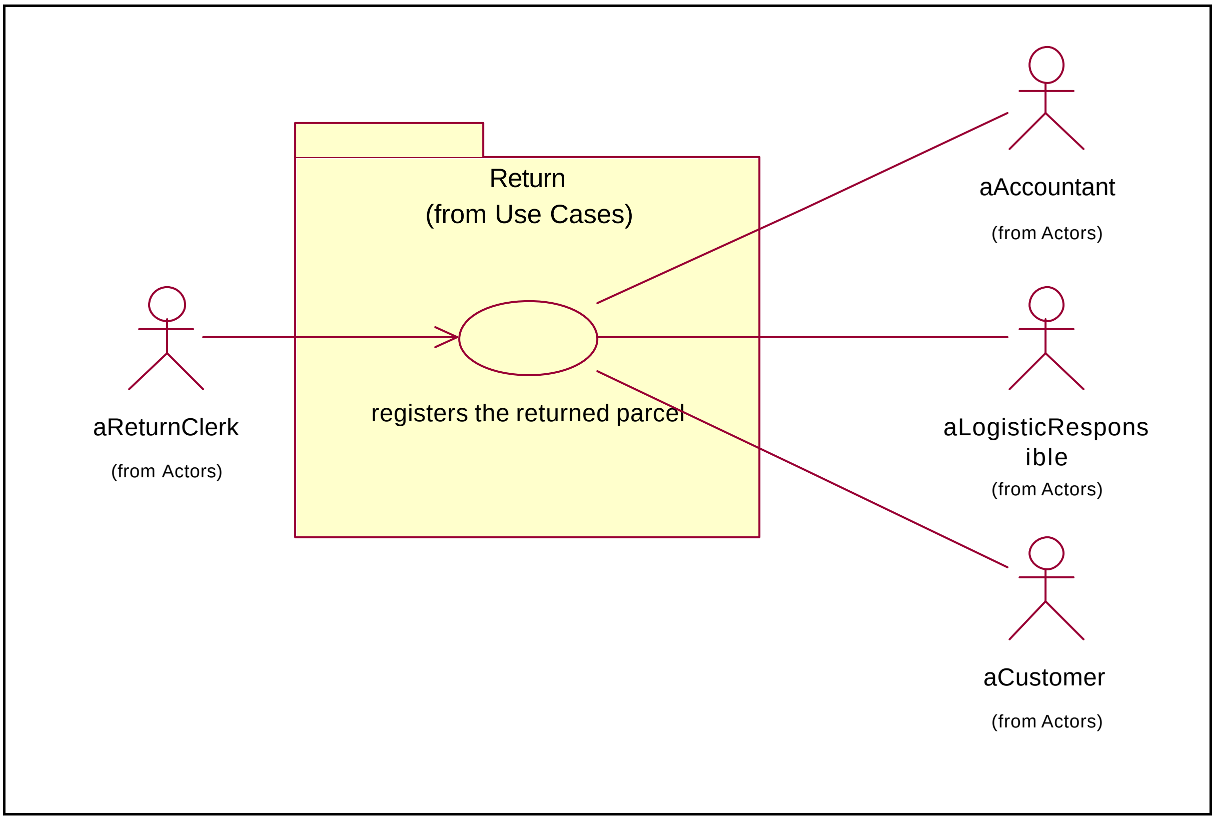


Figure 45: Use Case Diagram – Delivery (Return Group)

UC Identifier	Use Case	Description
UC 3.2.18	registers the returned parcel	For each returned parcel, the clerk registered it. He verifies that the return requirements are respected. He puts back the items in stock. He notifies the Accountable of the refund if all is OK, or he informs the customer of the problem if there is. If the return is because of the shipping company, the logistic responsible is informed about it.

Table 42: Use Case Description – Delivery (Return Group)

5.4.6 Knowledge base access

Add : Authority files, Thesaurus (DTD from Aquarelle)

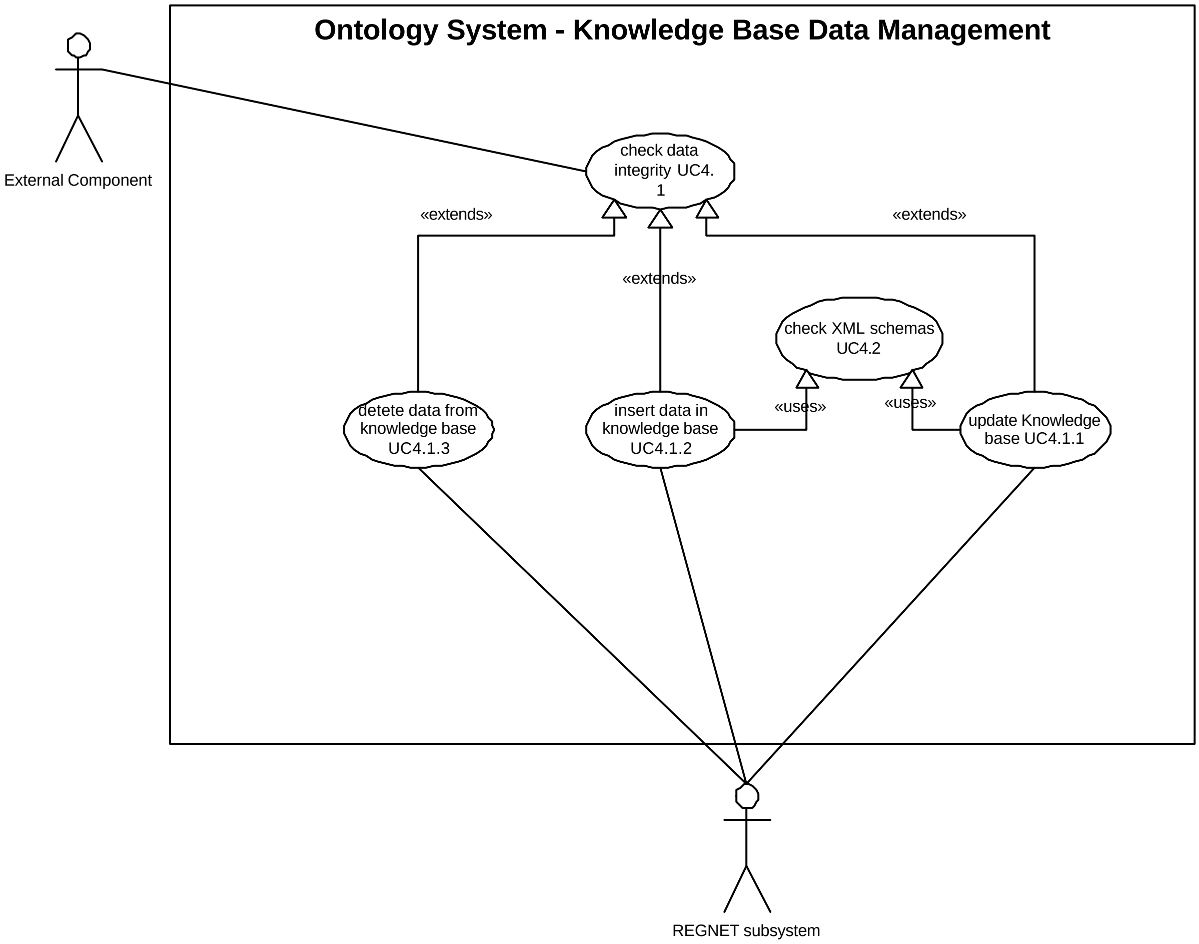

Ontology System - Knowledge Base Data Management
External Component
check data integrity UC4. 1
«extends»
«extends»
«extends»
check XML schemas UC4.2
«uses»
«uses»
detete data from knowledge base UC4.1.3
insert data in knowledge base UC4.1.2
update Knowledge base UC4.1.1
REGNET subsystem

Figure 46: Use Case Diagram – Knowledge Base Access (data management)

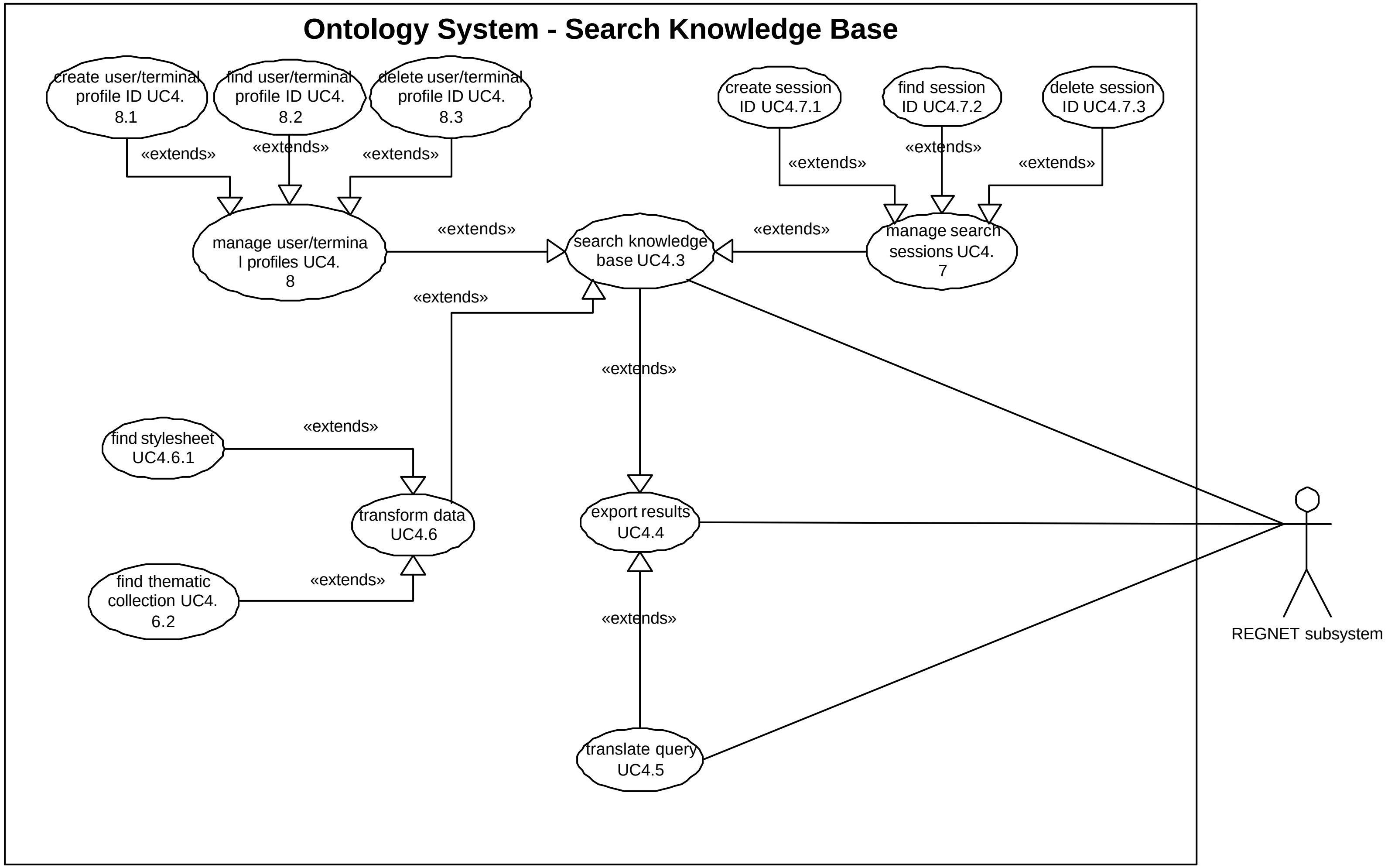

Ontology System - Search Knowledge Base
create user/terminal profile ID UC4.8.1
find user/terminal profile ID UC4.8.2
delete user/terminal profile ID UC4.8.3
«extends»
«extends»
«extends»
manage user/terminal profiles UC4.8
«extends»
search knowledge base UC4.3
«extends»
create session ID UC4.7.1
find session ID UC4.7.2
delete session ID UC4.7.3
«extends»
«extends»
«extends»
manage search sessions UC4.7
«extends»
«extends»
find stylesheet UC4.6.1
«extends»
transform data UC4.6
find thematic collection UC4.6.2
«extends»
export results UC4.4
«extends»
translate query UC4.5
REGNET subsystem

Figure 47: Use Case Diagram – Knowledge Base Access (search)

UC Identifier	Use Case	Description
UC 4.1	Check data integrity	The use of proper DTDs and XML-Schemas will be used to check data integrity
UC 4.1.1	Update Knowledge Base	Update knowledge base
UC 4.1.2	Insert data in Knowledge Base	Insert data in knowledge base
UC 4.1.3	Delete data from Knowledge Base	Delete data from knowledge base
UC 4.2	Check XML schemas	Check XML schemas
UC 4.3	Search Knowledge Base	Search knowledge base
UC 4.4	Export results	Export results in appropriate format like dbXML for example
UC 4.5	Translate query	Translate query in any format which is going to be used from a future version of the REGNET system
UC 4.6	Transform data	Transform data to appropriate format depended by the subsystem (for example XSL for EP-subsystem)
UC 4.6.1	Find stylesheet	Find style-sheet
UC 4.6.2	Find thematic collections	Find thematic collections
UC 4.7	Manage search sessions	Manage search sessions
UC 4.7.1	Create session ID	Depends on administrative level of access.
UC 4.7.2	Find session ID	Depends on administrative level of access.
UC 4.7.3	Delete session ID	Depends on administrative level of access.<
UC 4.8	Manage User/Terminal Profiles	Manage User/Terminal Profiles
UC 4.8.1	Create User/Terminal Profile ID	Depends on administrative level of access.
UC 4.8.2	Find User/Terminal Profile ID	Depends on administrative level of access.

UC 4.8.3	Delete User/Terminal Profile ID	Depends on administrative level of access.

Table 43: Use Case Description – Knowledge Base Access

Almost all priorities are set to “1” because Ontology System is the backbone of the REGNET System since every sub-system is connected with it. Priority of “Translate query” is set to “2” because it refers to future versions of REGNET and the possible future formats that are going to be used.

5.4.7 Electronic publishing

The Electronic Publishing Use Cases consist of two main Use Case diagrams. The first one dealing with more general functionality of the whole publication process (see 6.6.4) and the interfaces to other subsystems and the second one following the functional requirements from the Content Provider Group.

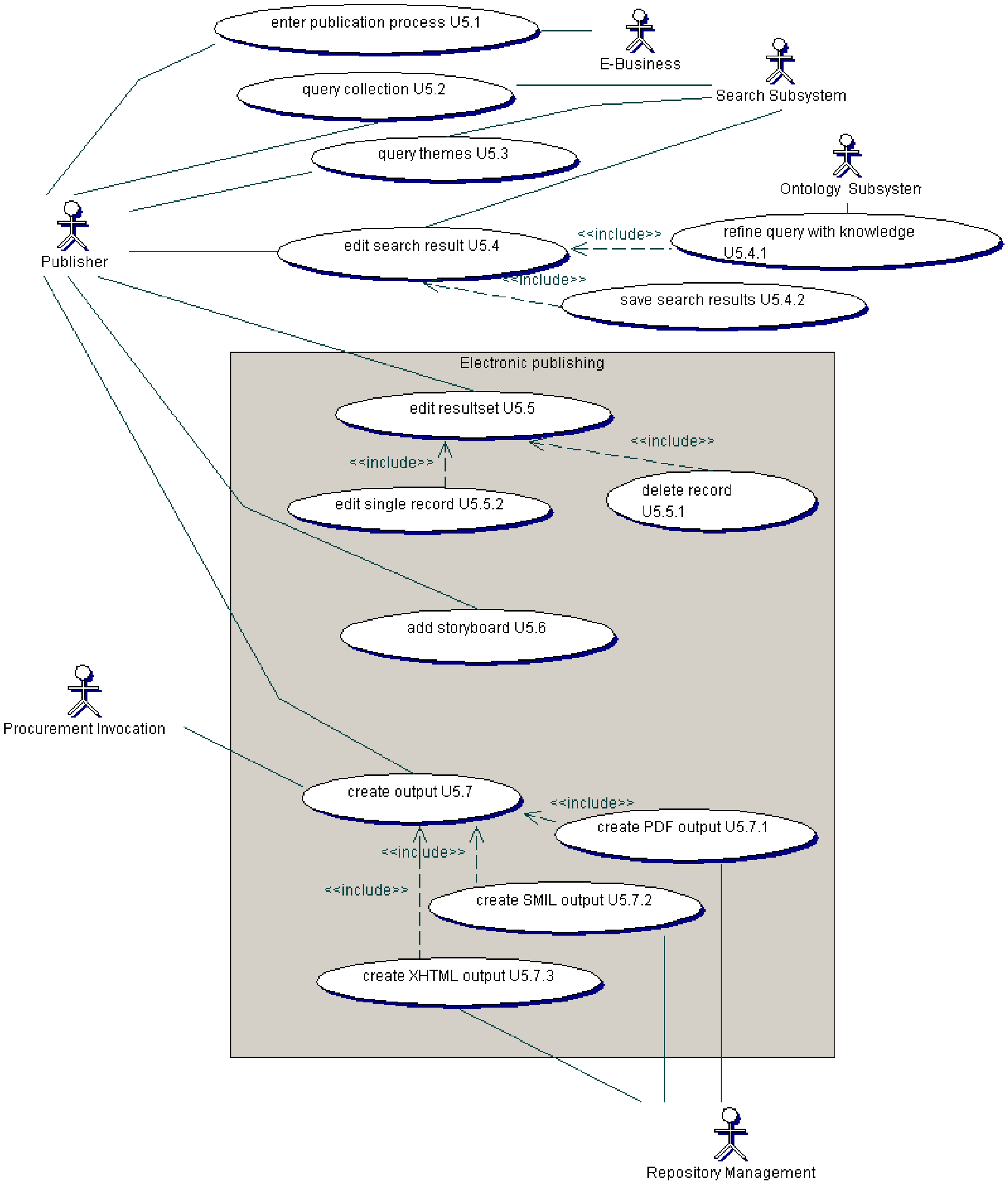


Figure 48: Use Case Diagram - Electronic Publishing

The Use Cases concerning the Electronic Publishing System are embodied in a grey box. The Use Cases U5.1 to U5.4 deal with tasks belonging to the whole publication process, but are performed in other subsystems (e-Business, Search and Retrieval, Ontology Subsystem). These subsystems are pictured as Actors in the diagram. Publisher is the only main human Actor.

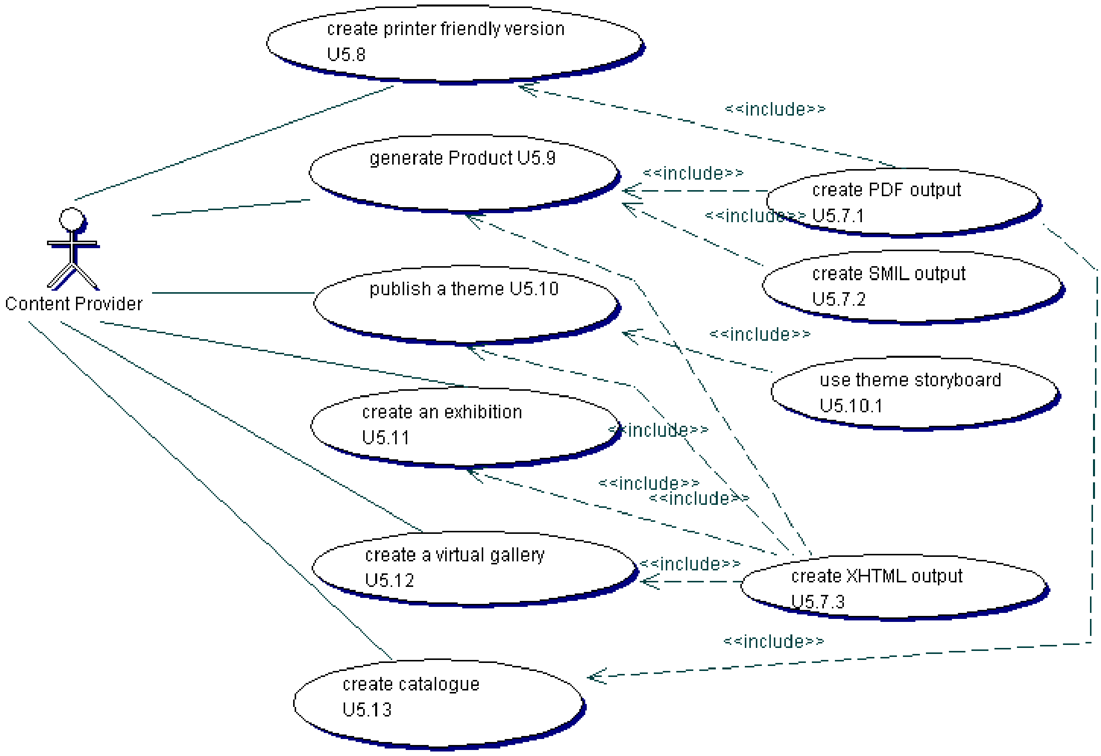


Figure 49: Use Case Diagram – Electronic Publishing (Content Group)

Table 45 shows the Use Cases extracted from the required functionalities of the Content Provider Group. All these Use Cases can be matched into the more general Use Cases and therefore all include U5.7.1,U5.7.2 or U5.7.3, which lead to a publication prototype.

UC Identifier	Use Case	Description
U5.1	Enter publication process	The Publishing Subsystem can be entered via the e-Business Subsystem
U5.2	Query collection	To get objects for publishing the publishing all the collections must be queried
U5.3	Query themes	The User can query themes or select them from a list
U5.4	Edit search result	The query result can be edited
U5.4.1	Refine query with knowledge	If results are not satisfying the result set can be extended by using the Ontology Subsystem
U5.4.2	Save search results	The ResultSet can be saved
U5.5	Edit resultset	The result set can be edited
U5.5.1	Delete record	Single records can be deleted from the result set

U5.5.2	Edit single record	Single records can be edited
U5.6	Add storyboard	If the result set is satisfying a storyboard can be added
U5.7	Create output	At the end of the publishing process an output format dependent on the publishing device can be chosen
U5.7.1	Create PDF output	A PDF output can be chosen
U5.7.2	Create SMIL output	A SMIL output can be chosen
U5.7.3	Create XHTML output	A XHTML output can be chosen
U5.8	Create printer friendly version	Create a printer friendly output (See U5.7.1)
U5.9	Generate product	Generate a publication prototype that can be used for further editing and refinement
U5.10	Publish a theme	Publish a theme: Get all the objects belonging to a theme and provide a navigational aid through a theme
U5.10.1	Use theme storyboard	Use theme storyboard
U5.11	Create an exhibition	Create a publication prototype for CD-ROM or PDF (See U5.7.1,U5.7.2,U5.7.3)
U5.12	Create a virtual gallery	Create a virtual gallery (See U5.7.1, U5.7.2,U5.7.3)
U5.13	Create catalogue	Create a catalogue (See U5.7.1)

Table 44: Use Case Description – Electronic Publishing

5.4.8 REGNET connector

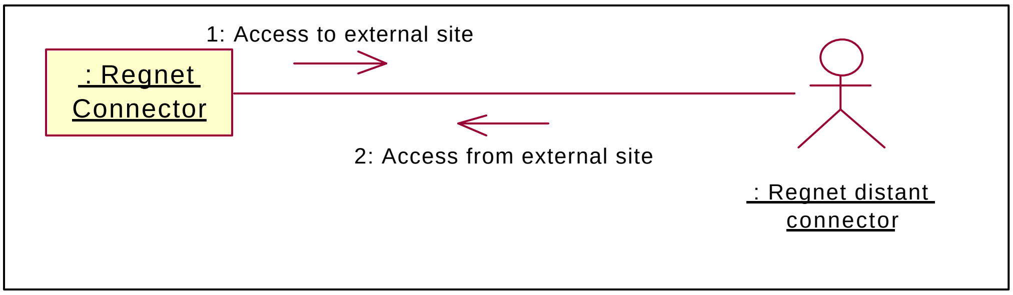

1: Access to external site
: Regnet Connector
2: Access from external site
: Regnet distant connector

Figure 50: Use Case Diagram – Connector (Context)

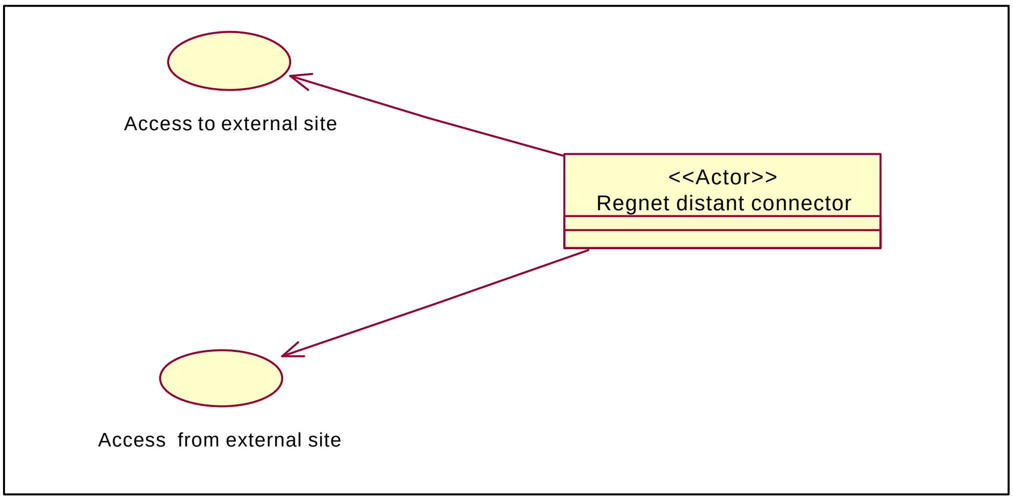


Figure 51: Use Case Diagram – Connector

Ident	Use case	Descriptions
UC6.1	Access to external site	This Use Case takes place when an external REGNET Cultural Organisation submit a query to the local system. The connector un-marshals the message and transmit it to the right local node.
UC6.2	Access from external site	This Use Case takes place when the local system want to submit a query to another organisation of the REGNET system. The connector marshal the message according to ebXML standard and send it to the distant organisation.

Table 45: Use Case Description – Connector

5.5 Functional architecture

Functional architecture present the REGNET components according to their function. These functions are given by functional requirement.

REGNET system is a market place dedicated to cultural-heritage domain. It provides sophisticated collaborative functionalities smoothly integrated thanks to an ontology system. These functions are:

- Distributed search and retrieval.
- Distant acquisition of Cultural Heritage data.
- Domain virtual directory and information.
- Small advertising (demand and sold) about CH objects.
- Virtual shop on-line in order to sell objects.
- E-procurement in order to get best prices from supplier.
- Auction system in order to sell some objects.
- Electronic publication in order to online elaborate CD-ROM.

REGNET is a network of collaborative systems. Each REGNET site (museum, library and so on) has a REGNET system which is link with others. It is on the responsibility of each site to maintain data about its object collections (this can be done locally to the institution or at the site level).

A user of a REGNET site can get information from local collections but also from collections provided by connected sites. This mechanism must be as transparent as possible for users. REGNET strength is correlated to this network mechanism.

Deployment diagram below illustrates a typical REGNET topology where the master site manage ontology data available from every regional centre. This figure gives a fractal view of REGNET network: each site seam to provide all functionalities of the system even if some of them are not locally available.

REGNET master site manages ontology data which are shared by all nodes.

REGNET regional centre manages REGNET sites: these sites may have installed functionalities (e.g. E-shop) or functionalities may be provided by the regional centre.

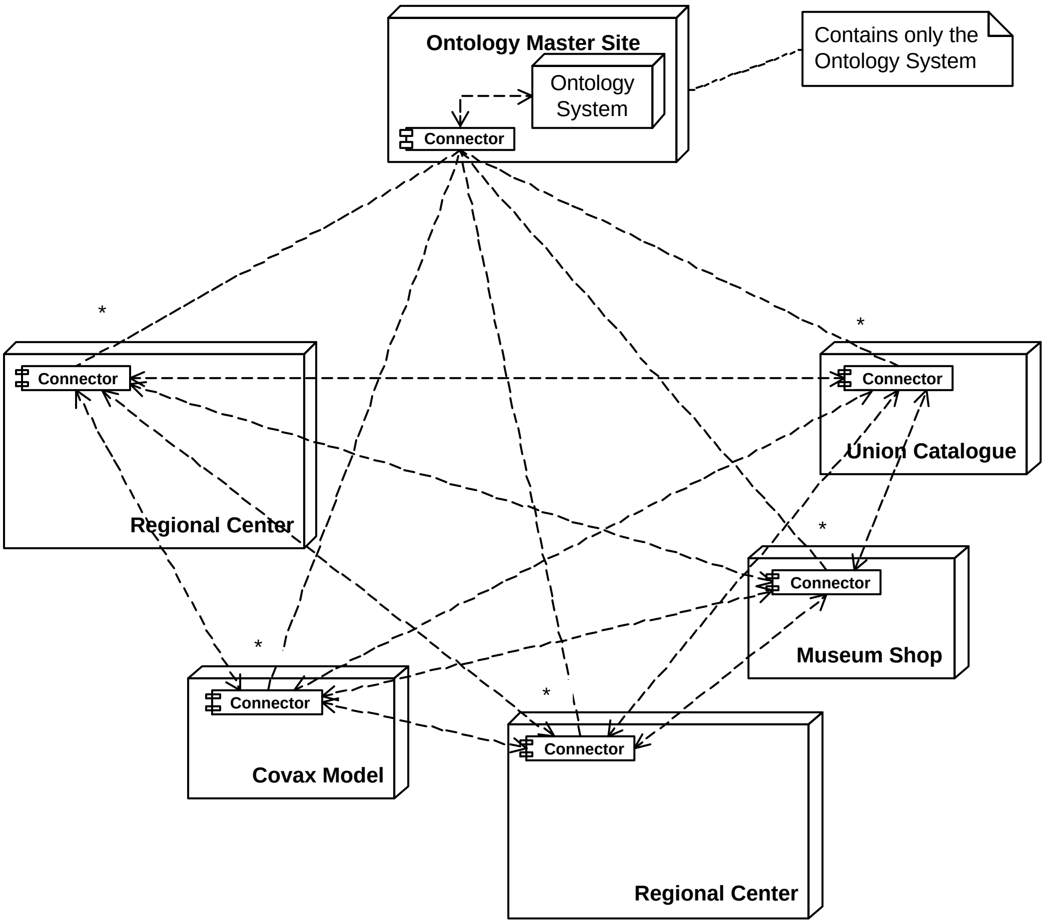


Figure 52: REGNET Network

REGNET architecture will be based on a three-tier architecture whose advantages are discussed in the state-of-the-art part of the document.

The functional architecture is represented by UML Component diagram. Links between components are detailed by tables.

5.5.1 REGNET functional architecture

According to tiers-architecture, REGNET functional architecture is given by the following figure:

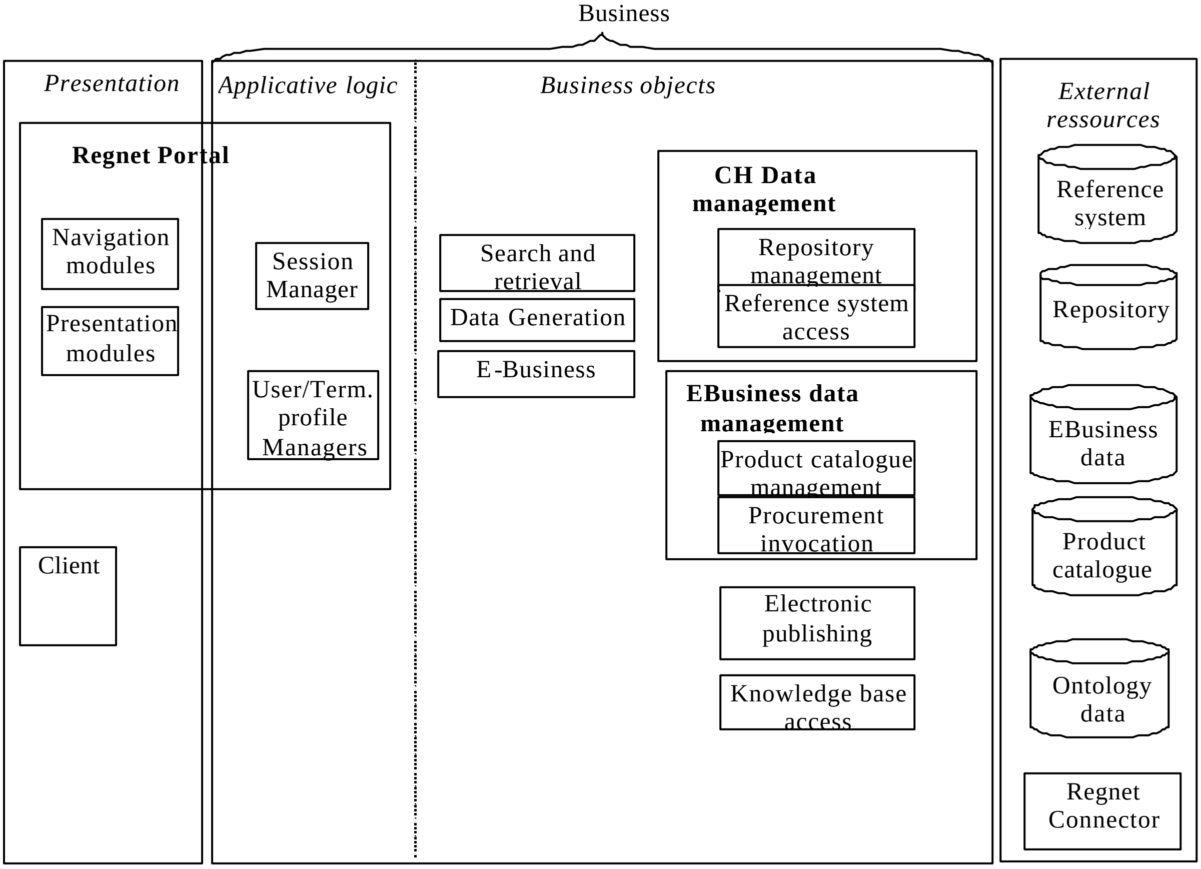


Figure 53: REGNET functional architecture

Presentation layer

This layer contain functionalities which are available from the client. It contain only user interfaces necessary for data-generation, search system and e-Business functionalities.

Different kind of client are available:

- Consultation clients: they only access search and eBusiness systems. These clients are Internet deployed and can be embedded either into standard web navigator or WAP or PDA devices.
- Acquisition clients: they can both consult and add information into the REGNET system. This information is kept by the data generation subsystem and deals with new assets for the e-Business or CH database.

Business logic layer

This layer can be logically separated into following sub-layers:

- *Applicative logic* which contains objects dedicated to applications. Examples of such objects are session management and user profiles objects.
- *Business objects* which contains "true" business objects which are dedicated to business domain, reusable and often persistent.

External resources layer

This layer contain local databases as well as access to external resources such as catalogues and data provided by other REGNET nodes.

Remote access from one CSC to other CSCs is provided by the REGNET Connector. This connector allows a site to communicate with others.

More description of connectors between CSC and CIO

5.5.2 REGNET Portal

The REGNET Portal is the entry point to access the REGNET services. It includes two layers: presentation and applicative logic.

In the presentation logic, the Portal provides presentation services and navigation services, adapted to the specific clients who request them, in support of REGNET's business functions.

- Presentation services
  - Adaptation of contents to be presented according to client capabilities
  - Graphical user interface for user agent clients
- Navigation services
  - Binding of Portal resources

In the applicative logic, the Portal hosts the Session Manager, the User Profile Manager, and the Terminal Profile Manager.

- Session manager: it tracks the user interactions with the Portal.
- User profile manager: it manages the data flow of user profiles.
- Terminal profile manager: it manages the data flow identifying the type of client terminal.

The following figure shows the functional architecture of the Portal:

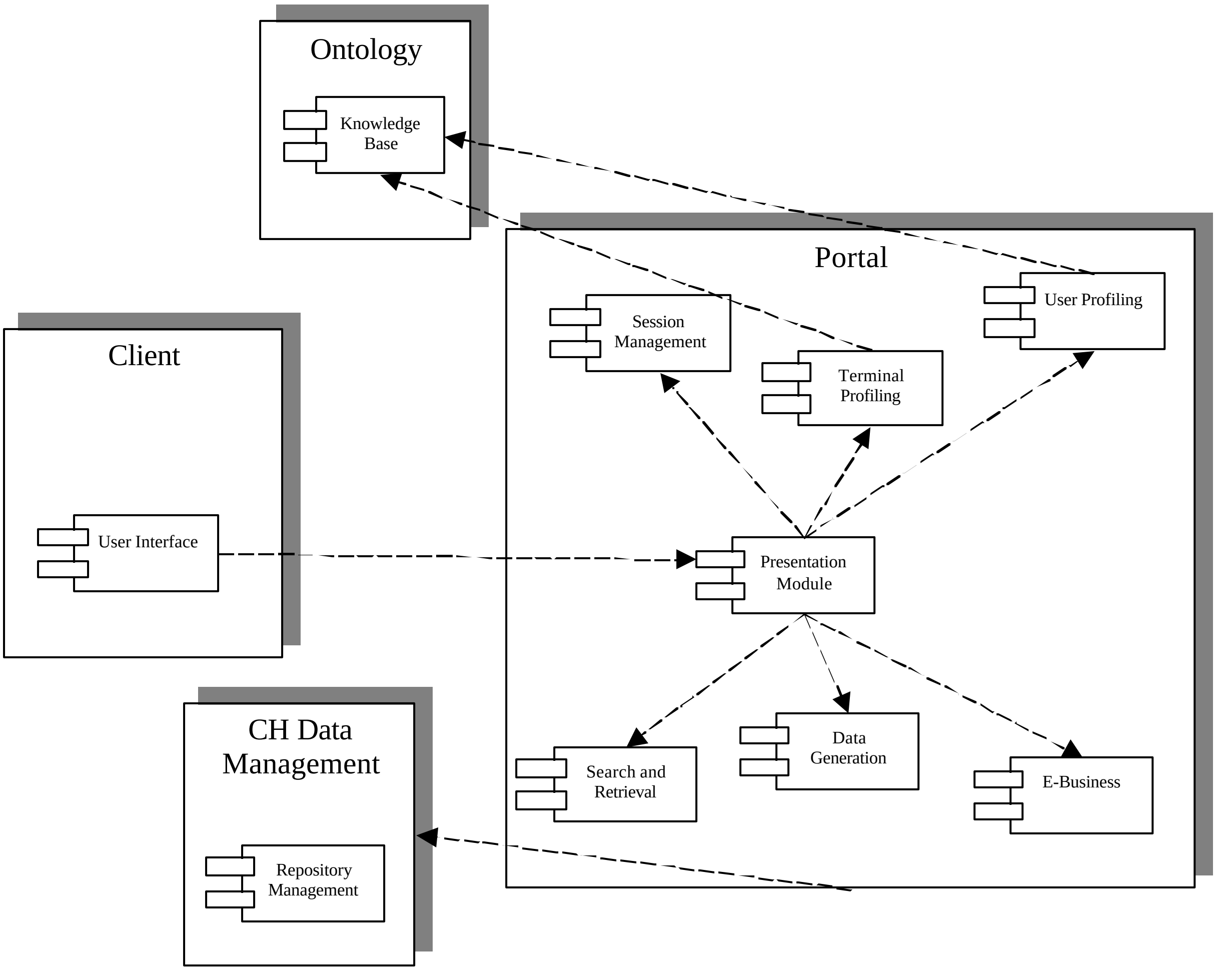


Figure 54: Deployment Diagram - Portal functional architecture

From this architecture, there are arrows that point out the functional dependencies:

- Session, Terminal profile, and User profile need data from Presentation Module, that communicates with the Client's User interface.
- Search & Retrieval, Data generation, and e-Business need data from the Presentation Module.
- Terminal and User profile need data from the Knowledge Base Access of Ontology subsystem when the user and terminal recognition is requested.
- User interface needs data from Repository management of CH Data management when after receiving the metadata as result of a searching process the user requests to download the real objects.

Contents are presented to the user in a personalised fashion to enhance his navigation experience and improve the effectiveness of the services. This requires the contents (both information and layouts) and services, where applicable, to present customisable alternatives. While the user navigate the Portal, it can interact with customisable elements, or it can supply his preferences through on-line forms, or the system stores user's actions and derives from them his preferences. This combined approach is valuable to address terminals constraints. User profiling necessitates of a mechanism to identify user identity: this is detailed in following sections.

To address the different capabilities of REGNET's target user's devices (PC, WAP handsets, Laptops, etc.) contents must be tailored depending on device constraints, and presented in the appropriate language. This task is performed in the Presentation Layer. Again, this requires the contents and services to allow filtering and tailoring.

As the front-end of the system, the Portal addresses security requirements, in terms of the following properties:

- Non-occurrence of improper alterations of information (integrity)
- Non-occurrence of unauthorised disclosure of information (confidentiality)
- Readiness for usage (availability)

These are particular relevant, due to the business-oriented nature of the system. Security issues must be taken into account at every level of the communication (from physical to application layers) and in end-to-end paradigm.

5.5.2.1 Data Generation

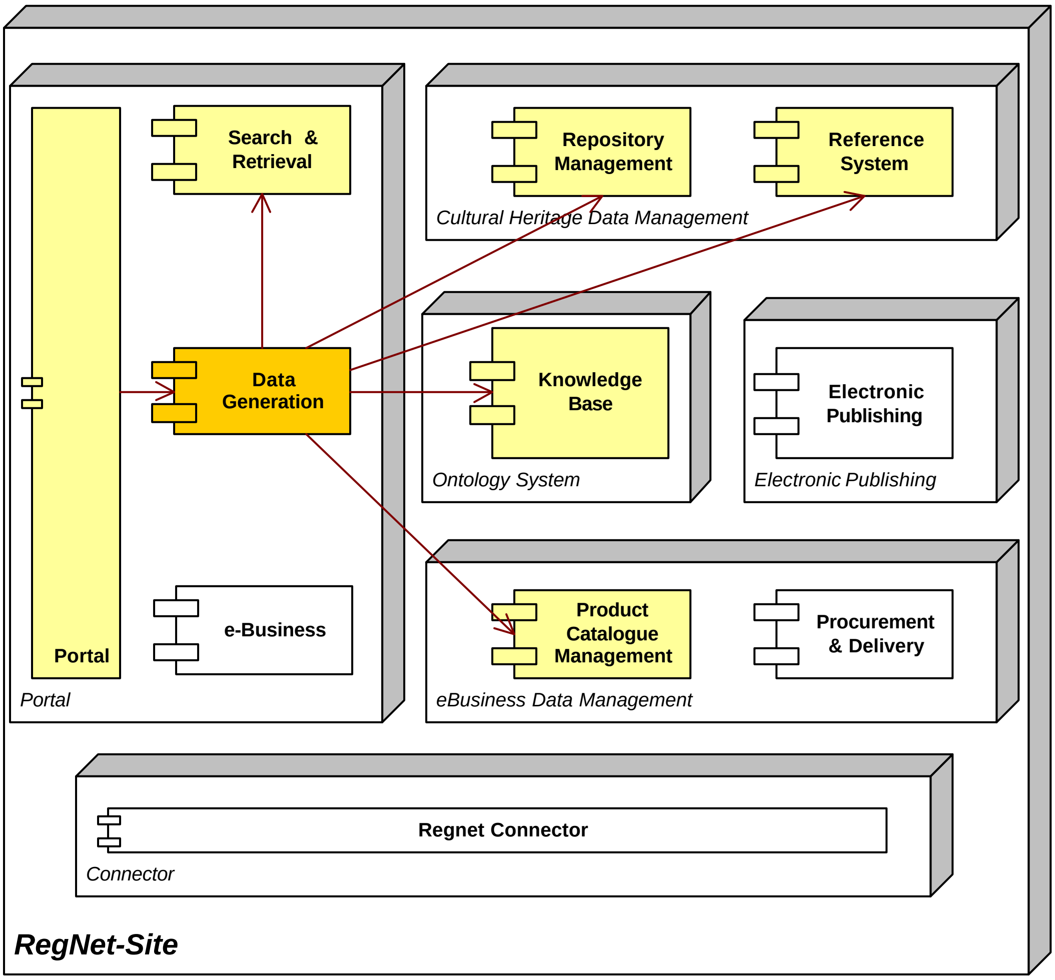


Figure 55: Deployment Diagram - Data Generation functional architecture

The functional dependencies of the Data Generation component are pointed out in the above figure: here are isolated the connections to the other node within the REGNET site.

The Data Generation component is placed inside the Portal node, which it communicates with, through the presentation sub-module of Portal.

The data inserted (through the Web Acquisition Client) of the user are parsed by the Data Generation subsystem that will be responsible for their management.

In particular the Data Generation component will communicate with:

- The Search & Retrieval component to query for object/product in the Repository.
- The CH Data Management node to which can ask the insertion, the update or the delete of objects and information related either to local or to remote repository.
- The Ontology System used as the common knowledge base of the REGNET system. This node considers of the access to thesauri, to themes, registered users, location and availability of repositories, etc.
- The Product Catalogue Management.

5.5.2.2 Search and Retrieval

The Search & Retrieval component will be located in the 'web-tier' but it has mainly business logic components as presentation and applicative logic will partly be handled by the Portal in common for the 3 subsystems in the Portal node.

The task is to provide access to search functionalities for:

- clients (= Portal) (internet browsers, handhelds, …)
- the Data Generation Subsystem (Subsystem-4)
- the e-Business Subsystem (Subsystem-6).
- the Electronic Publishing Subsystem (Subsystem-8).
- REGNET Connector (i.e. other Search & Retrieval components from other REGNET-Sites).

Therefore the Search System has to be able to retrieve data from the:

- Reference System (Subsystem-1)
- Catalogue Management (Subsystem-7)
- REGNET Connector (i.e. other Search & Retrieval components from other REGNET-Sites).

The Ontology System (Subsystem-3) is used as a common knowledge base. This node supports the Search & Retrieval component with:

- access to thesauri to support users with their searches
- access to themes)
- XML-Schemas and DTD's for metadata formats, query definitions, protocols between nodes and result format definitions.
- User data to handle queries and result sets for registered users.
- Location (and availability of repositories) in the REGNET Network.

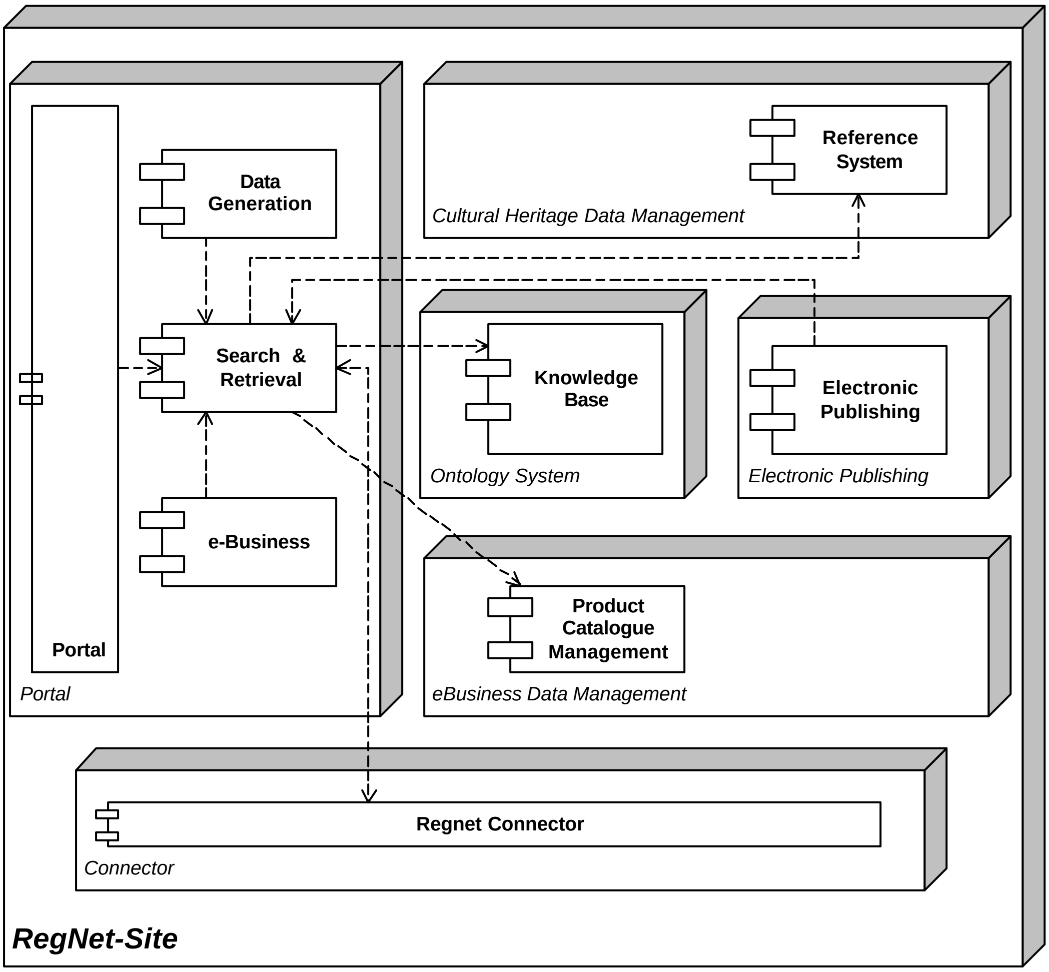


Figure 56: Deployment Diagram – Search & Retrieval

The above diagram shows only the Search & Retrieval dependencies to other nodes within one REGNET-Site.

The main task is to distribute queries to all the possible targets (Reference System, Catalogue Data Management, Ontology System and Connector (= other REGNET Sites) and to merge the results together.

Though it is a general task of the Portal, the Search & Retrieval will also have to check user actions and save queries for reuse. This is why user profiles are requested from the Ontology System.

Also logging of usage for the Search & Retrieval component will be established. This statistical data will be very useful for surveys of system usage and comparison with other systems.

5.5.2.3 e-Business

5.5.2.3.1 General framework

The basic concept of the e-Business sub-node will be based on the ebXML specifications. Therefore our approach will establish the ebXML configuration of the e-Business subsystem, which will reflect the special needs of the REGNET project.

It is very important to define in great detail the functional architecture of the specific component. Since that the e-Business subsystem is consisted by two main modules, the B2C and the B2B. It is necessary to distinguish the functionalities of the node in the above two main features.

5.5.2.3.2 e-Business (B2C)

It is very important to provide the architecture of the e-Business services of the REGNET system. The basis of the approach that it was presented earlier in the functional requirements can now be presented as procedural model.

The procedural model describes in detail the architecture and the specifications of the REGNET e-Business solution. The basis of the proposed subsystem architecture is the definition of the three tier architecture of the REGNET core system.

The requirements of the customers are establishing the basis for the design of the architecture.

e-Business means doing financial transactions on the Internet, at an online store or over a secure business-to-business extranet.

A successful e-Business architecture requires a merged approach combining expertise from both the network and application development camps to make sure that all aspects of Web-based e-Business systems are addressed properly.

The component model of the e-Business services is described in the following figure.

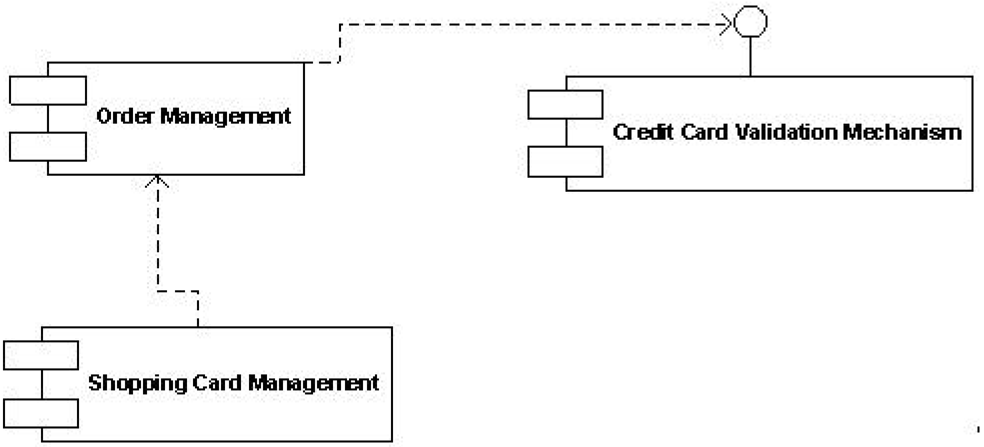


Figure 57: Component Diagram – e-Business system architecture

The above diagram describes in detail the behaviour of a customer in relation to the services that the REGNET system will provide.

To be more specific, the activity diagrams of the E-Commerce (B2C) subsystem are described below.

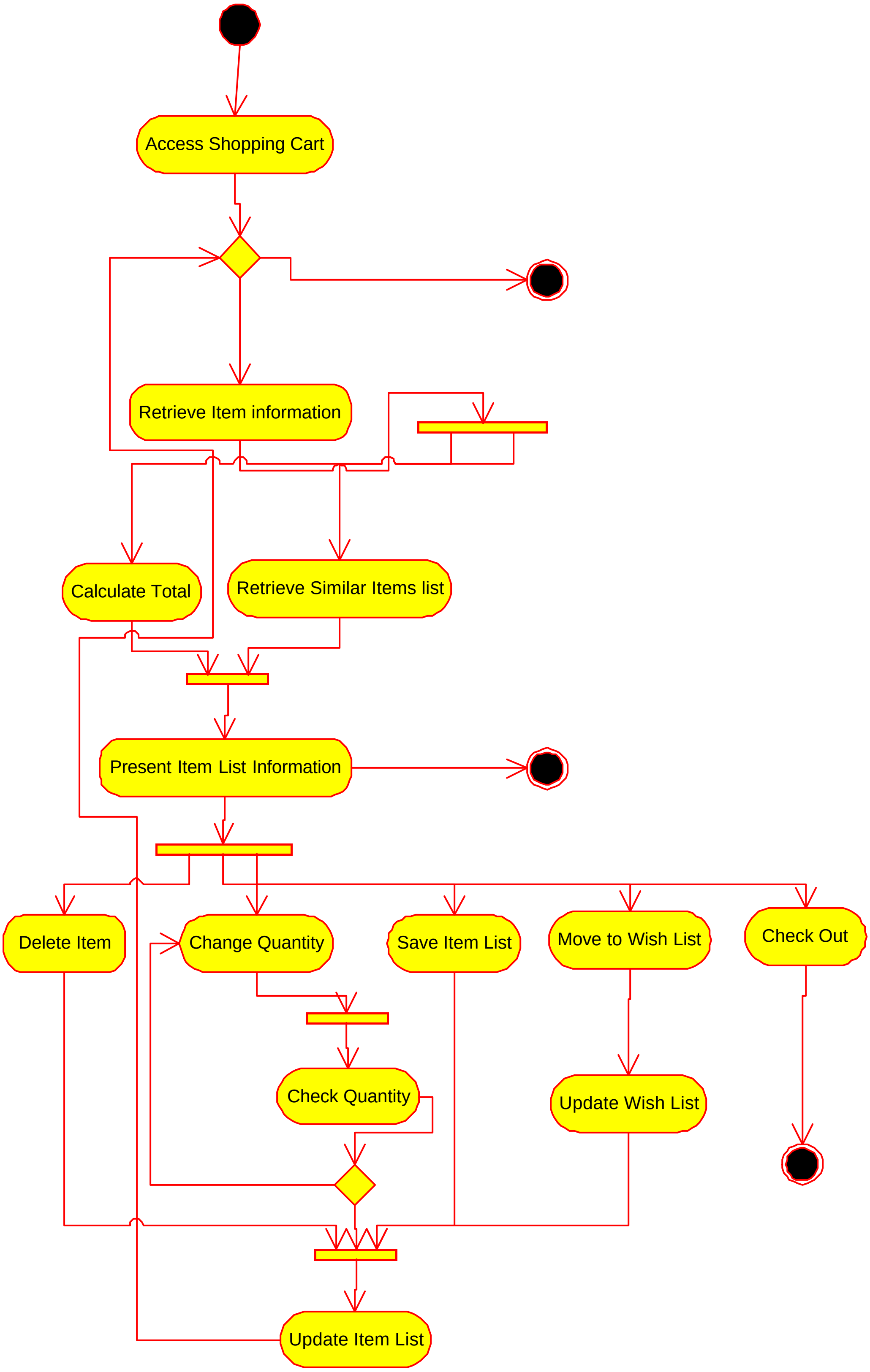


Figure 58: Activity Diagram - e-Shop

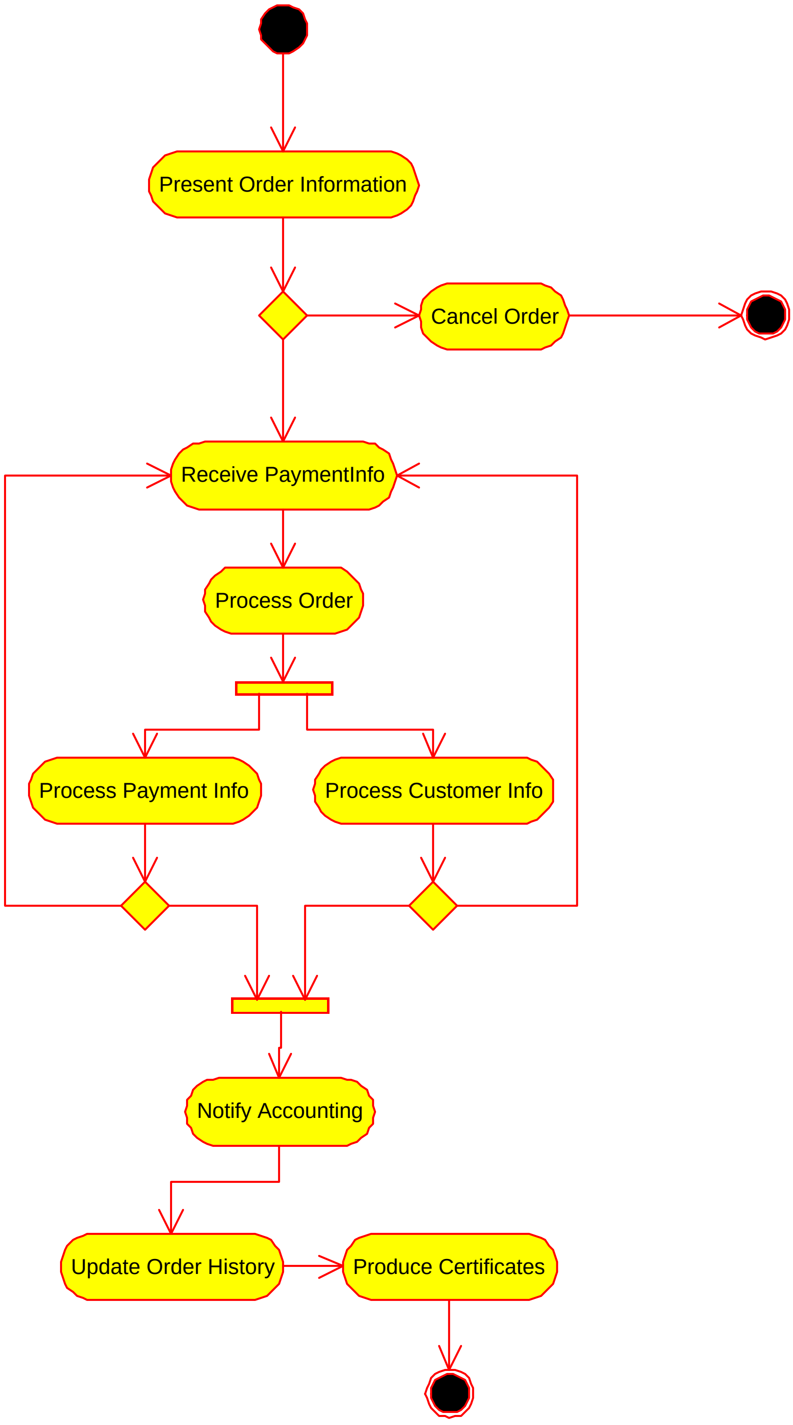


Figure 59: Activity Diagram - Order Management

5.5.2.3.3 B2B

The main aspect of the B2B feature of the REGNET system is the integration of the ebXML, which is a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet. Using ebXML, companies now have a standard method to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes.

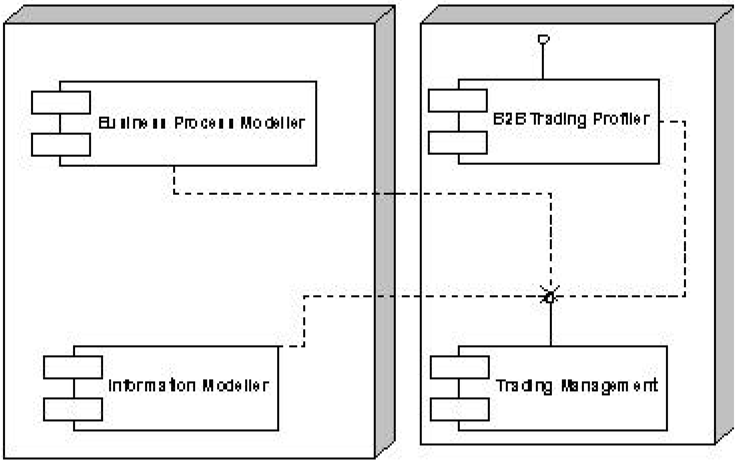


Figure 60: Deployment Diagram - B2B components

It is important to clarify the exact actions and functionalities of the E-Business subsystem, which are relevant to the special issue of the B2B implementation of the REGNET system. The main aspect is the integration of the ebXML specifications in the description of the B2B case.

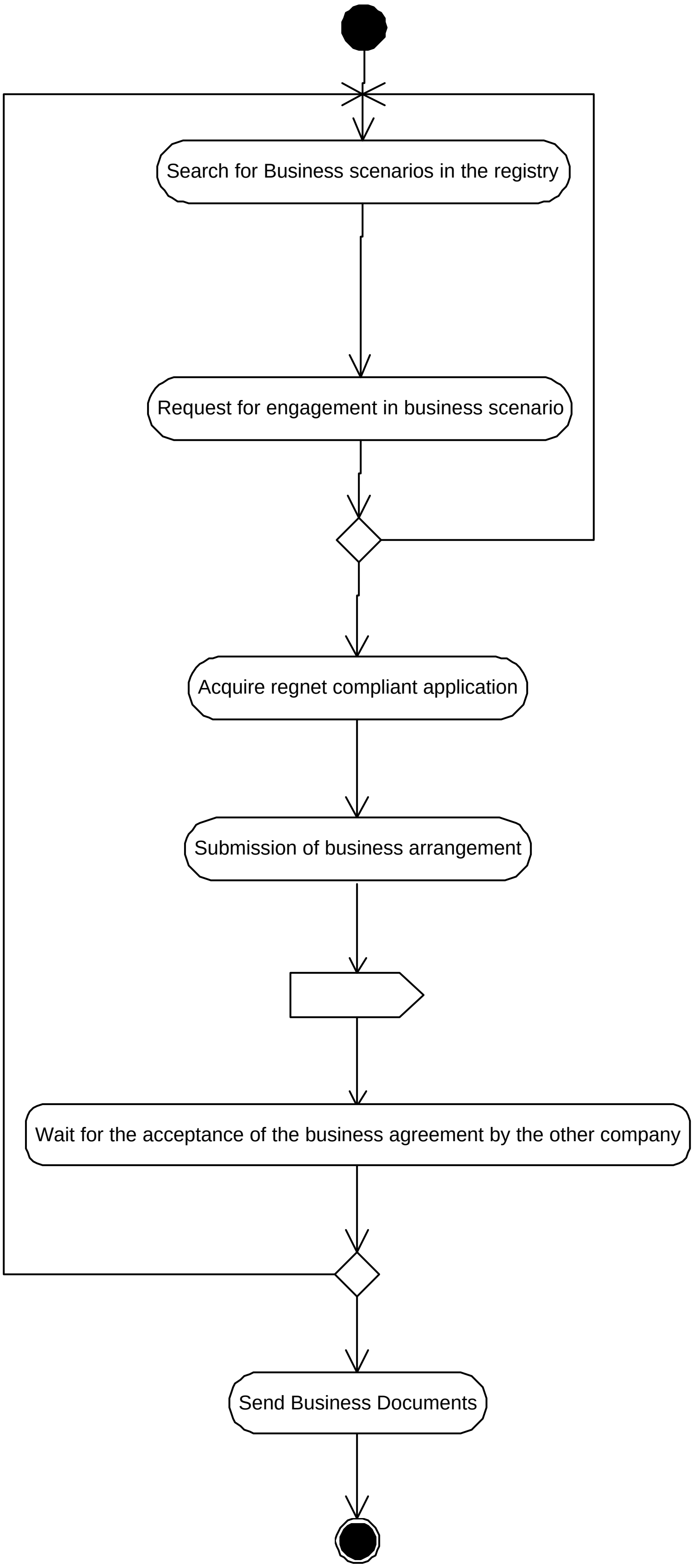

Search for Business scenarios in the registry
Request for engagement in business scenario
Acquire regnet compliant application
Submission of business arrangement
Wait for the acceptance of the business agreement by the other company
Send Business Documents

Figure 61: Activity Diagram - B2B activity according to ebXML specifications

Besides that, the E-Business subsystem will provide the ability to all users to take advantage the open feature of the ebXBML standard.

5.5.3 Cultural Heritage data management

The goal of the Cultural Heritage Data Management (REGNET Node 2) is to supply and promote an application-independent interoperability framework that can be used by a variety of partners who are engaged in publishing digital content on the Web.

REGNET is composed of an arbitrary number of repositories (data storage), containing digital contents (electronic documents, images, audio files, video streams, etc.) coming from REGNET content providers such as museum collection, photograph library catalogue and any kind of interesting archives.

The distributed repositories of REGNET are network accessible, filtered by Portal subsystem, to which user's requests can be submitted. It is expected that Cultural Heritage Data Management (Node-2) will be able to return a record about each object held in the content's provider database.

Searching various databases concurrently is a complicated matter for several reasons: their records differ, covered fields differ, the syntax in the fields differ and access protocols differ (if network accessible overall).

The Cultural Heritage Data Management aim is to choose an appropriate strategy to develop a joint database record profile, so individual fields in each original database will be mapped to this common profile.

As the next step each partner (content providers over all) will extract the relevant fields from their database and expose their data to a common REGNET front end, containing similar resource descriptions (but describing different material).

To create the initial common record profile, the REGNET partners need a known resource description method, which could serve as a "switching language" between individual database records. For this purpose we will use the Dublin Core Metadata Element Set.

As shown in the following figure the Cultural Heritage Data Management consists of 2 subsystems:

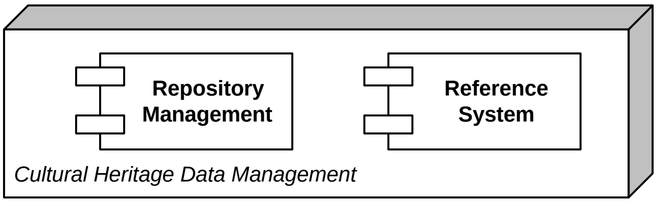


Figure 62: Deployment Diagram – Cultural Heritage Data Management

- Repository Management

This subsystem contains digital surrogates of 'primary' (real world) objects (e.g. example images, audiovisual and audio files, text files, …).

- Reference System

This subsystem contains meta data about real world objects. The meta data can be related to digital surrogates in the Repository Management system but can also be stand alone. This Meta data is searchable and provides links to the digital surrogate if existing.

5.5.3.1 Repository management

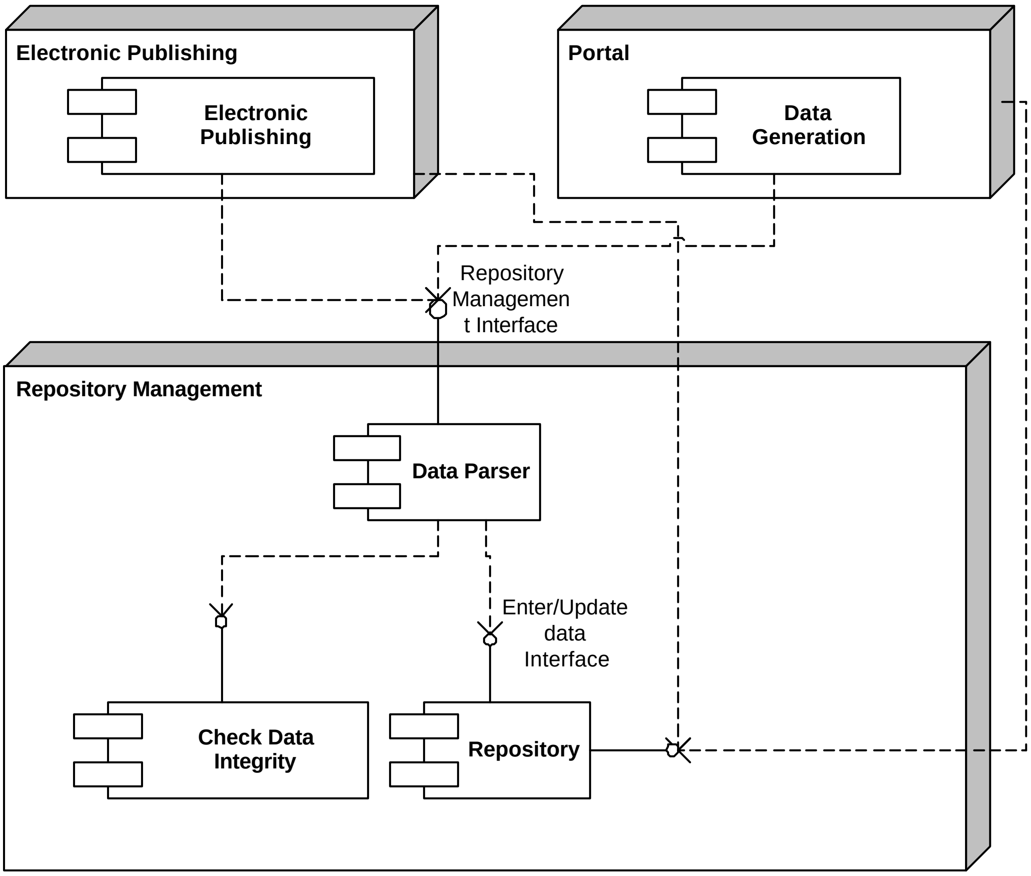


Figure 63: Deployment Diagram - Repository Management

The Repository Management aim is to hold digital surrogates of CH objects.

The Repository Management will be composed by suitable components depicted in the figure above;

essentially its composed from a component that is responsible for parsing the data in entry (e.g. distinguish the data context), a component which checks the data integrity and the repository for data storage.

The data flow entering the Repository Management is fed by the Data Generation component into the Portal node for what concern data coming directly from user (through the Web acquisition client) and the Electronic Publishing node for what the publishing regards.

The output of the Repository Management component concerns the presentation of the results due to request coming from the Portal.

5.5.3.2 Reference System

This subsystem supports data management and searches of Cultural Heritage meta data on distributed repositories in one Cultural Heritage Data Management node.

The clients for data entry are:

- the Data Generation subsystem from the Portal and
- additional external data entry facilities.

The clients for the search functionalities are:

- the Search & Retrieval subsystem from the Portal and
- the Catalogue Data Management subsystem in the e-Business Data Management node.

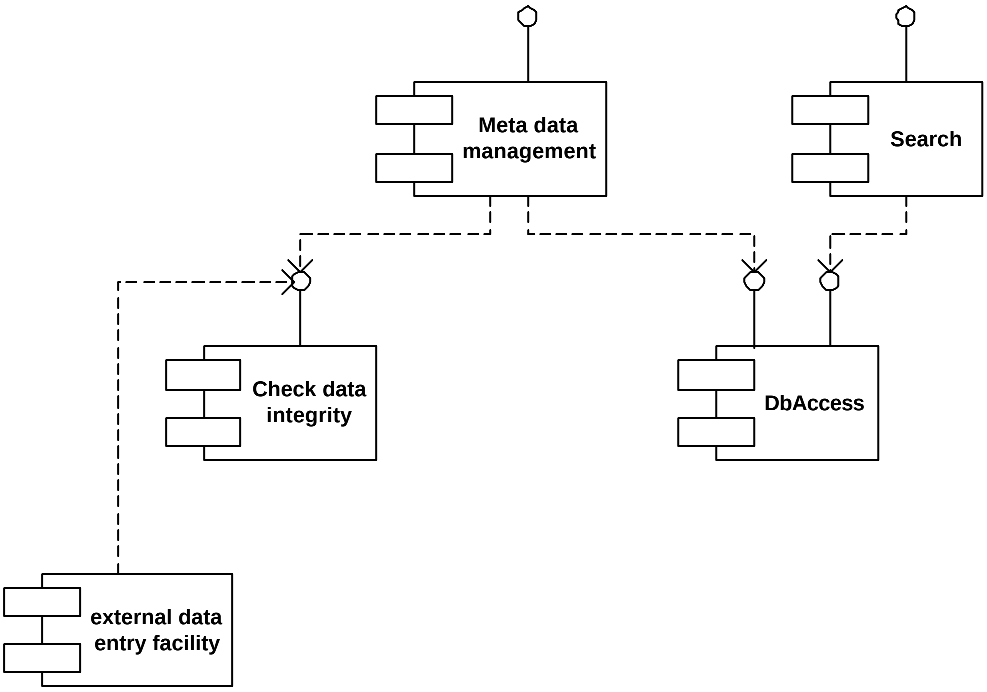


Figure 64: Component Diagram – Reference System

As stated above the Reference System has two main components: one for meta data management and one for search functionalities.

The meta data management is for creating and deleting repositories and inserting, updating and deleting in records in the repositories. The interface to this component has to be defined. One option is to use the extended services of the Z39.50 protocol, but it has to be evaluated if this is a useful solution. For each request in this component a security check has to be performed ensuring that the request origin (user) is allowed to process the requested action.

Every repository creation process needs support from the Ontology System to ensure that all necessary data is provided to create a new repository.

For insert and update requests the check data integrity component is responsible to check whether the data is in a valid data format. This assures that all repositories contain only checked and valid data.

The Search component is for searching the distributed repositories hosted in the surrounding Cultural Heritage Data Management node. The interface to the Search component is based on the XER/Z39.50 approach. All requests (search, sort, scan, present) have to follow the XER XML-Schema. The Schema can be found at http://asf.gils.net/xer/ez.xsd.

The DbAccess component is an abstraction layer for accessing the repositories. It provides an interface for data management and one for searching. For each database system used in the Cultural Heritage Data Management node an interface has to be provided. The DbAccess layer decouples the rest of the Reference System from the real database specific interface implementations. There are no restraints for databases to use as soon as an interface to the Reference System is available.

Finally components (mainly already existing tools for specific databases) will be used to create repositories from outside the REGNET System. The component must use the check data integrity component to check the data which will be inserted into the repositories.

5.5.4 e-Business data management

Data management of eBusiness related data. These data are coming from CH data node or from user through the Web acquisition client. Each piece of data can be split into two parts: meta-data and data. Meta-data can be represent with XML notation and are validate by a DTD or schema. These

DTD or schema are stored into the ontology node. Data format depend of the represented information.

5.5.4.1 Catalogue management

The Product Catalogue Management (PCM) is the basis for the description of products by the use of meta data.

The following diagram illustrates functional links between components. A migration system is used in order to transfer data from an external legacy system to the REGNET Catalogue Management system. The metadata repository is a sub-part of the Ontology system.

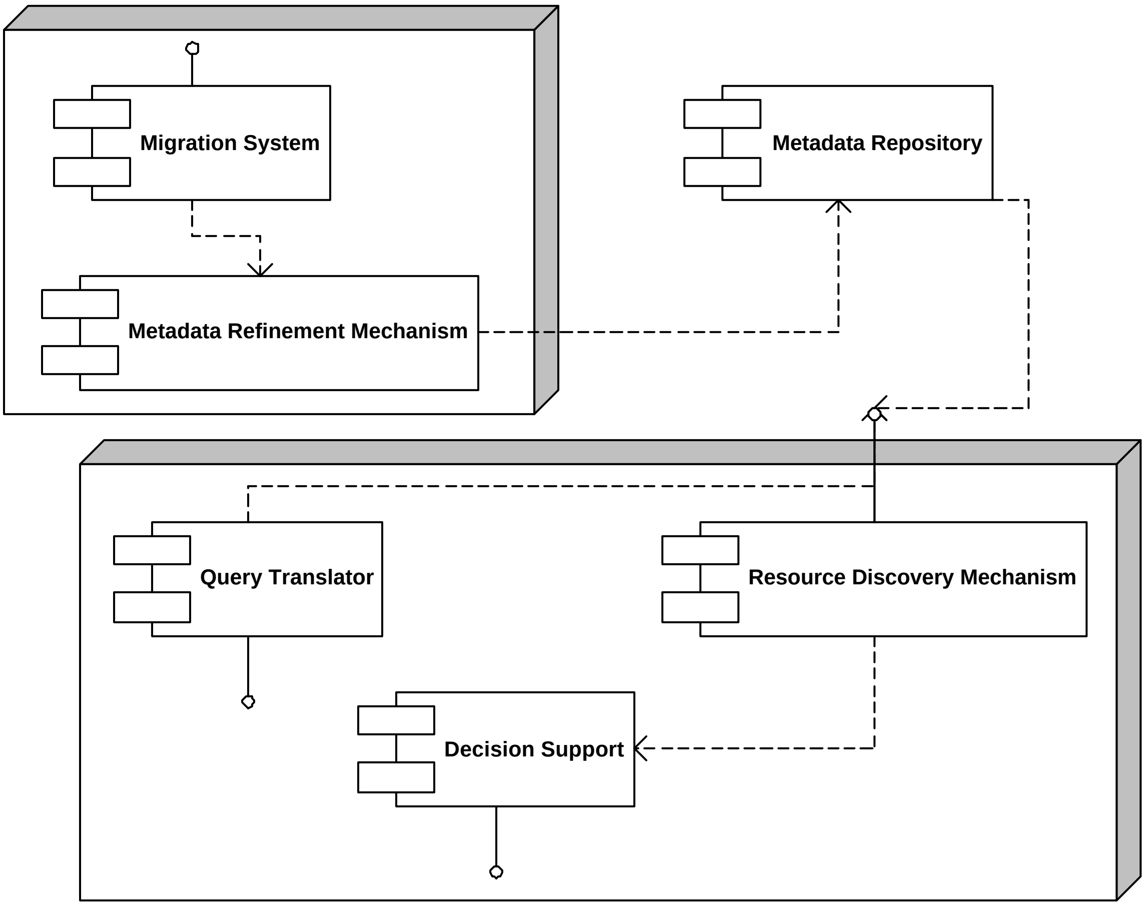


Figure 65: Deployment Diagram – Product Catalogue Management

A) Metadata Management System

Metadata Refinement Mechanism:

Since the current feature extraction methods are not perfect, the initial subject clustering of the database images clustering may be inaccurate. In addition, new images and new subject domains may be added into the database. Thus, metadata must be dynamically updated. The metadata refinement involves two aspects:

- Updating existing metadata based on database updates. (Input from **Migration System**)
- Updating existing metadata based on query feedback

The metadata refinement process is often time-consuming and should be performed off-line.

Migration System:

The Migration system will reflect all changes of a product features (i.e. price etc.) to the Product Catalogue Management. Therefore it will provide the input to the metadata refinement mechanism to update existing metadata based on database updates.

B) Decision – Query Management System

Resource Discovery Mechanism:

The goal of the RDM is to find the best collections for a query that an end-user has made to the system. This is really important and efficient for the systems performance, because the end users have a large variety of searchable collections available to them. Accessing all these collections for each query is not practical and the response time in too many collections will vary. Besides that, only few collections will contain useful items for a given query. Therefore, the demand for a tool that assists users in discovering the useful resources for their queries is great. This process will be covered by the RDM.

Query translator:

The queries that are given to the system by the user need correspond directly to requirements on our metadata architecture.

Decision Support:

The Decision Support mechanism will provide the decision for the output of the results of the queries into a necessary format in the presentation layer.

C) Metadata Repository

The model correspondences in the modelling phase separate the different kinds of meta-information. Following the same strategy these different kinds of meta-information are stored into different repositories. So, there are different sub-repositories, in order to store the information models, the meta-information models and the dependencies of the above two. The last sub-repository stores the meta-information for the selective extraction of the query results.

The above components constitute the operational and functional features of the Product Catalogue Management and form the basis of the e-Business system.

However the activity diagram of the PCM is the following:

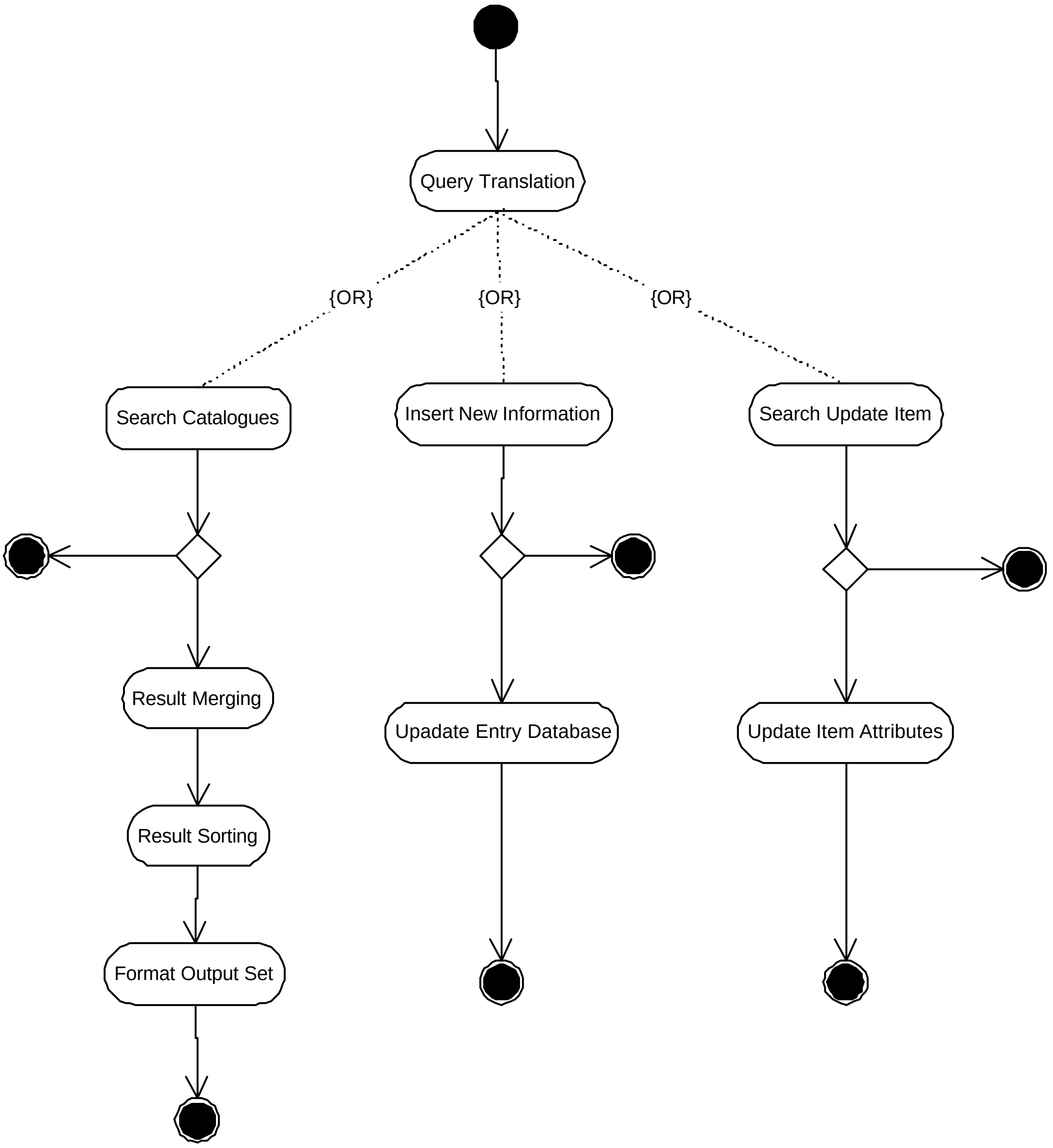


Figure 66: Product Catalogue Management

The above figure describes in great detail the three main activities of the PCM:

- Provide access to the distributed catalogues and inform the interested user about the services and the products that are offered by the providers.
- Provide the ability to each supplier to manage the content of the catalogues automatically.
  - Update the attributes of the catalogues
  - Insert new products or services to the stored catalogues.

5.5.4.2 Procurement and Delivery

5.5.4.2.1 Procurement

In our study, the procurement unit is functionally simple and independent : it is a market place where suppliers can offers products/services and buyers can contact them if they are interested in buying.

There isn’t an actual competition, the goal is more to make contacts easier, everybody can take benefits to participate to this market place :

The buyers have a global access to news products and buyers, the research costs are reduced, the suppliers can create an electronic storefront which is accessible by a large group of buyers. Moreover, it can be the tool to establish a durable co-operation and to develop news products.

The supplier:

He has to complete the registration forms and submit it to administrator for approval. The administrator verifies the references of all suppliers to ensure quality and security. If he prequalifies the supplier, he registers him and send a confirmation.

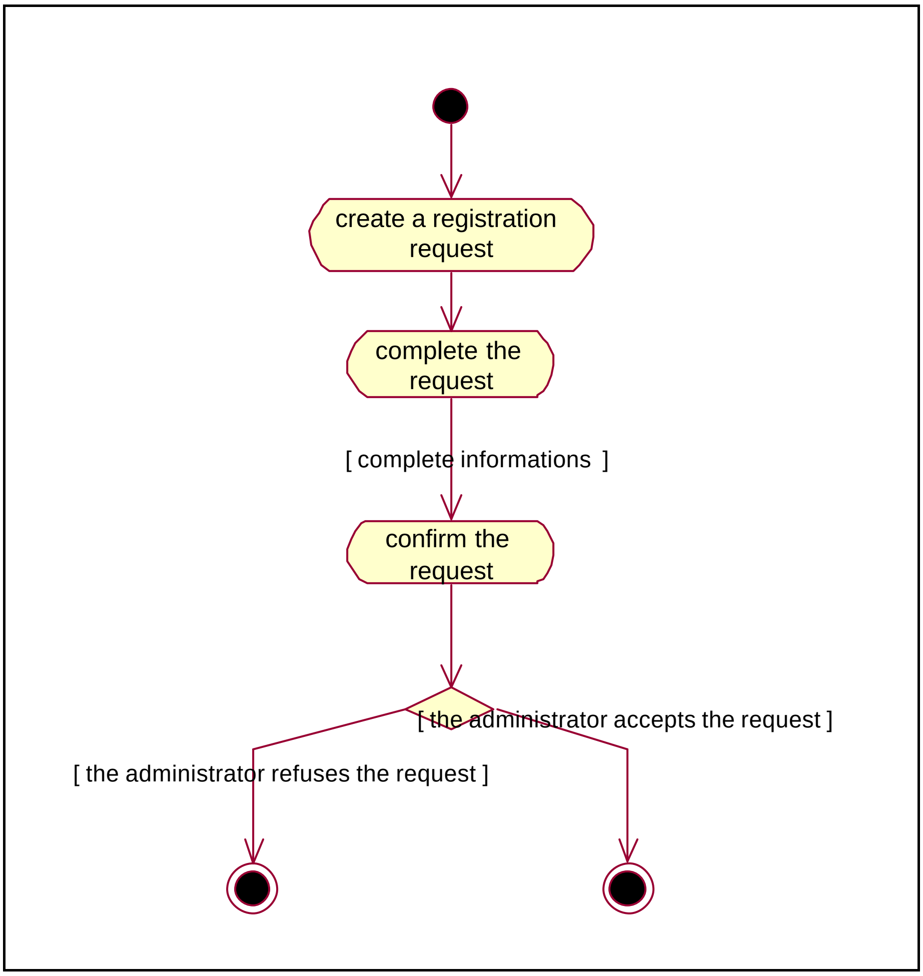

create a registration request
complete the request
[complete informations]
confirm the request
[the administrator accepts the request]
[the administrator refuses the request]

Figure 67: Activity Diagram – Procurement (“registers as Market Place supplier”)

The supplier creates his virtual showcase: he enters the references and information of his offer according to the configuration of market place model.

He can modify his showcase: add, modify or delete product.

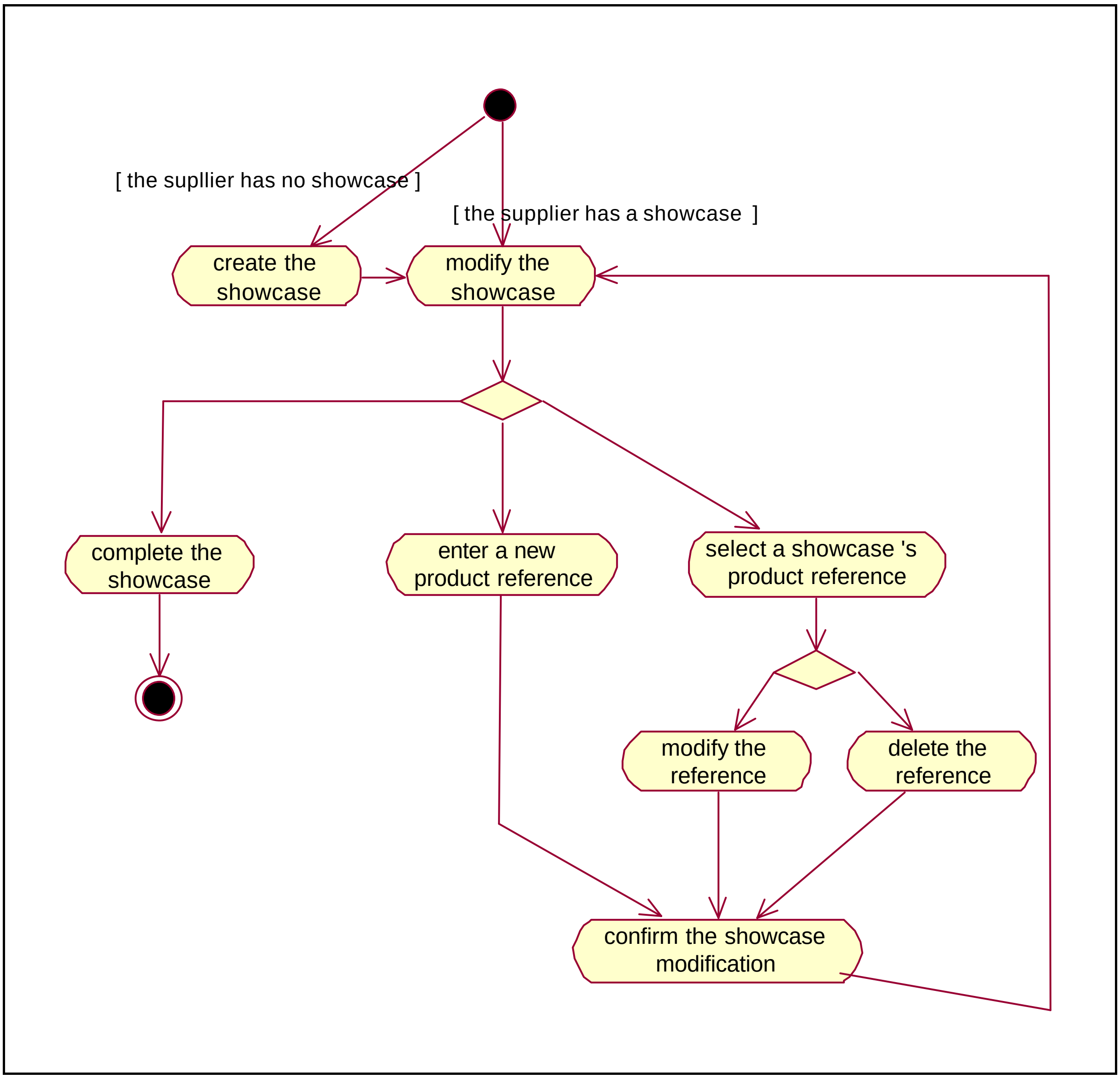

[the supllier has no showcase]
[the supplier has a showcase]
create the showcase
modify the showcase
complete the showcase
enter a new product reference
select a showcase 's product reference
modify the reference
delete the reference
confirm the showcase modification

Figure 68: - Activity Diagram – Procurement (“dresses his showcase”)

The buyer:

The buyer shops on market place ; he create a shopping list that he can modify, and finally he validates it, the suppliers of shopping list products receive automatically an order request.

The buyer and the supplier have to finalise the terms (shipping, timeframe and payment terms).

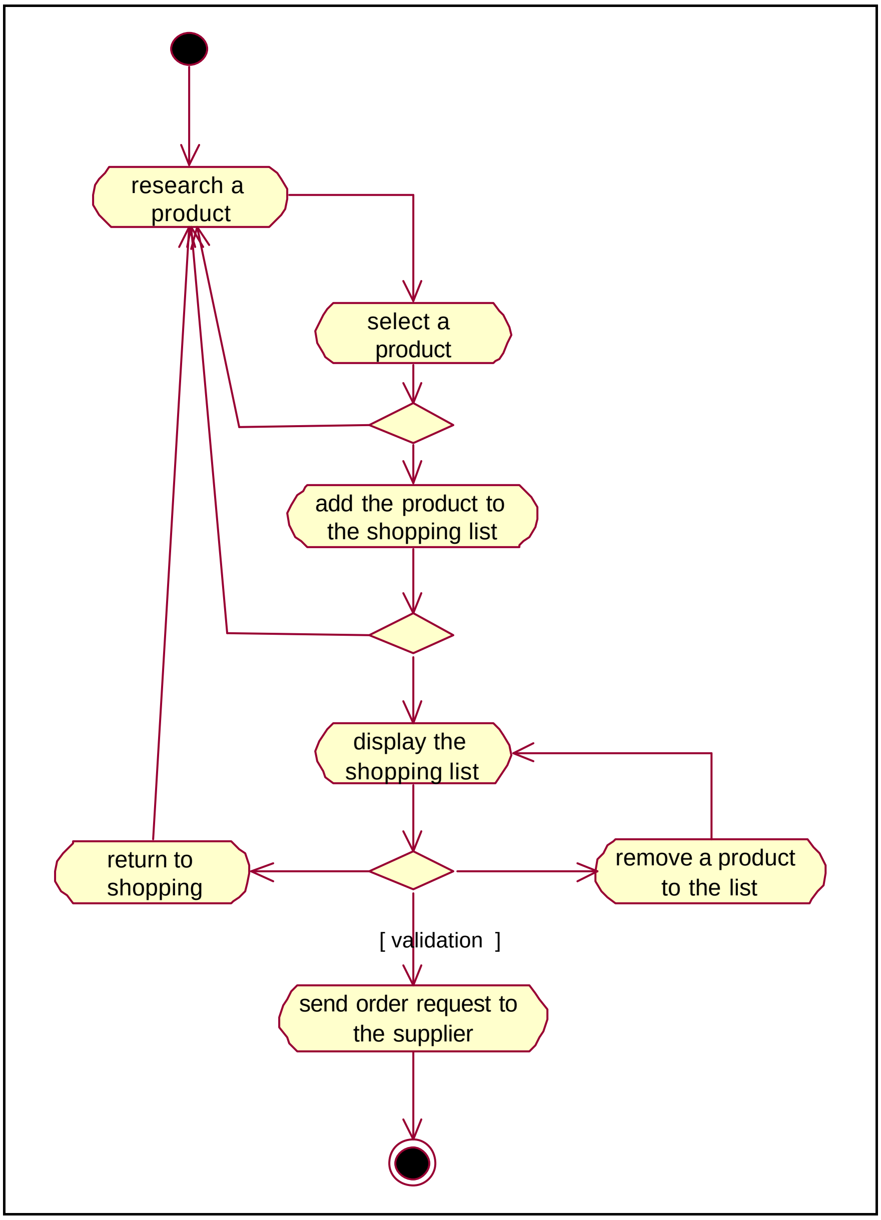

research a product
select a product
add the product to the shopping list
display the shopping list
return to shopping
remove a product to the list
[validation]
send order request to the supplier

Figure 69: Activity Diagram – Procurement ("places an order")

5.5.4.2.2 *Delivery Invoice*

Functionally, the e-Shop Museum is composed of :

- The sale unit
- The delivery unit
- The stock unit
- The accounts unit

These different units communicate together.

The functional field of “Delivery Invoice” concerns the delivery unit: preparing parcels and invoices and sending them. No verification is executed about the authorisation for a payment cart, the cheque reception or the order’s settlement.

The delivery unit intervenes after the sale unit, it takes in charge all the new orders registered in the sale unit on customer’s request. The delivery unit send messages to the stock unit (updating items quantity) and the accounts unit (sent invoices).

The business workers of “Delivery and Invoice” are identified (ONTHOLOGY). They login via the REGNET Portal which gives us access to the Delivery functionality.

They are:

- **The Logistics responsible:** He manages the Delivery and Invoice service : parcels sending and return.
    - He makes sure of enough and appropriate stock of packaging.
    - He handles all customer’s question or complaint about order’s delivery.
    - He defines the return conditions and informs the customer about him on the REGNET Site and on the invoice which is sent.
    - He contacts the shipping companies and arranges the terms of transport contract. Some shipping company proposes free services : packaging as pack, box, shipping documentation as waybill, tracking label, shipping reports (with tools to integrate into REGNET Site – tracking).
    - He has to make sure of the shipping rates (last updating).
    - All the shipping companies have their rates which are established according to the weight of the parcel. Consequently, it could be relevant that the customer has his shipping cost calculated according this same criterion; the weight of each item of eShop’s catalogue must be recorded , the customer must have access to this information which justifies the transport cost.
    - He verifies the invoice of the shipping company and make sure of its settlement.
- **The shipping clerk:** He takes in charge the parcels preparation and he makes sure of the collection by the shipping company.

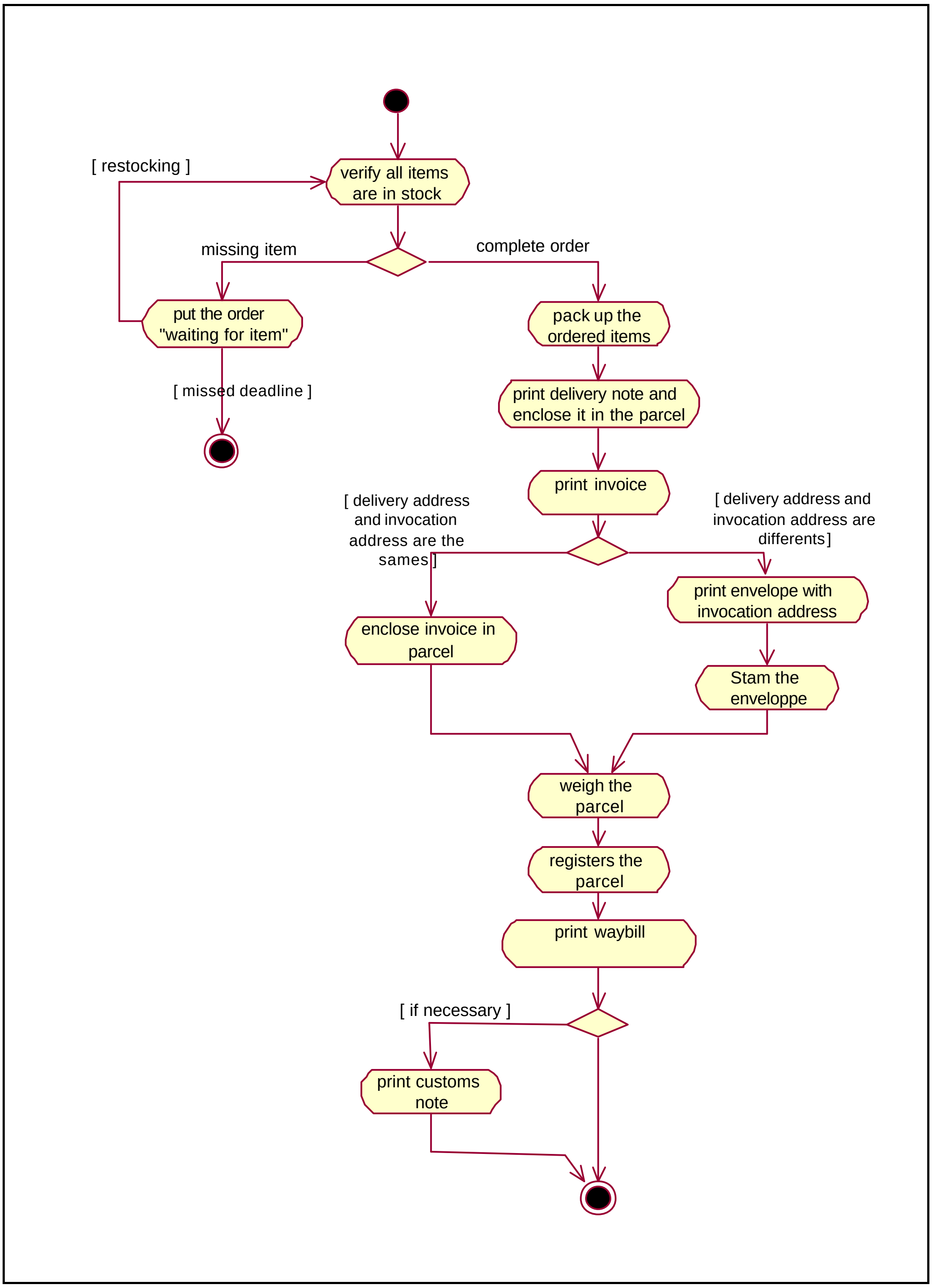

[restocking]
verify all items are in stock
missing item
complete order
put the order "waiting for item"
[missed deadline]
pack up the ordered items
print delivery note and enclose it in the parcel
print invoice
[delivery address and invocation address are the sames]
[delivery address and invocation address are differents]
enclose invoice in parcel
print envelope with invocation address
Stam the enveloppe
weigh the parcel
registers the parcel
print waybill
[if necessary]
print customs note

Figure 70: Activity Diagram – Delivery (“prepares the parcel”)

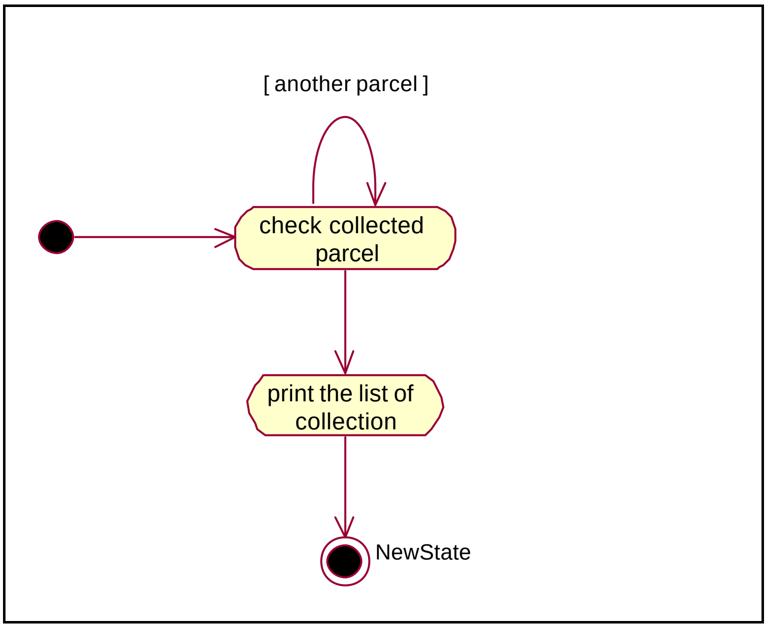


Figure 71: Activity Diagram – Delivery ("sends parcel")

- **The return clerk:** He registers the returned parcels. He checks them (condition, delay,...).

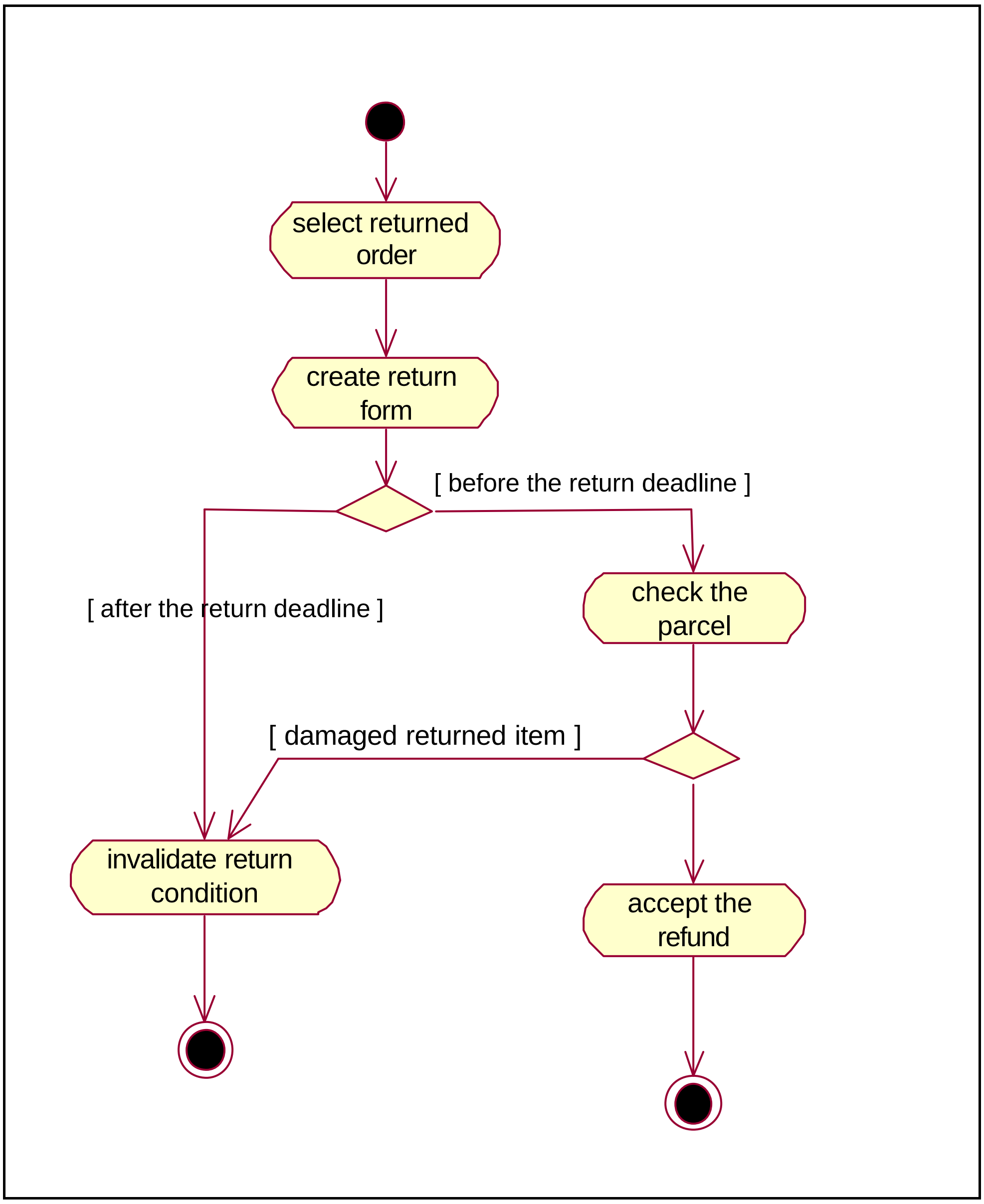

select returned order
create return form
[before the return deadline]
[after the return deadline]
check the parcel
[damaged returned item]
invalidate return condition
accept the refund

Figure 72: Activity Diagram – Delivery ("registers the returned parcel")

Modify in order to consider that stock is manage at the CIO level or outsourced.

The services for the customer:

- The customer who wants to know his order's progress, can track it. He logins on REGNET Portal and asks to consult his account. He must sign in (identification and password). The System proposes the list of his open orders or all orders. When he selects an open order, the System displays the detail and the step of the order. Moreover, the System can automatically send to the customer an email to notify the sending of the parcel.

5.5.5 Knowledge Base Access

The Ontology System node is central to the REGNET architecture, as it controls, to a great extent, the integrity and the logic of the system. This means that it provides the necessary meta-data and tools to check the validity of user input, native REGNET data, and actually the system's output.

The Ontology System can be perceived as a backbone to the entire system, as a common knowledge base and set of rules that the other subsystems should apply. As such it is connected directly to all REGNET subsystems. The end user of course, has no direct access to the Ontology System.

Knowledge Base can be considered as the intermediate link between the user access subsystems (like Data Generation, Search & Retrieval, e-Business) and the content related subsystems (like Reference System, Product Catalogue Management, Procurement & Delivery, Electronic Publishing). Knowledge Base could also be connected with external registries too.

The task of the Ontology System is the management of the Knowledge Base and the handle of different kind of data like:

- *User profiles*
- *Terminal profiles*
- *Thesauri*
- *Authority information*
- *DTDs and XML-Schemas for metadata formats*
- *Query definitions*
- *Binary Data*
- *Style-Sheets*
- *Topic Maps*
- *Location and availability of repositories in the REGNET network*
- *Administrative information*

Every subsystem of REGNET is connected to the knowledge Base of the Ontology System. The following diagram shows only the Knowledge Base dependencies to other nodes within one REGNET-site.

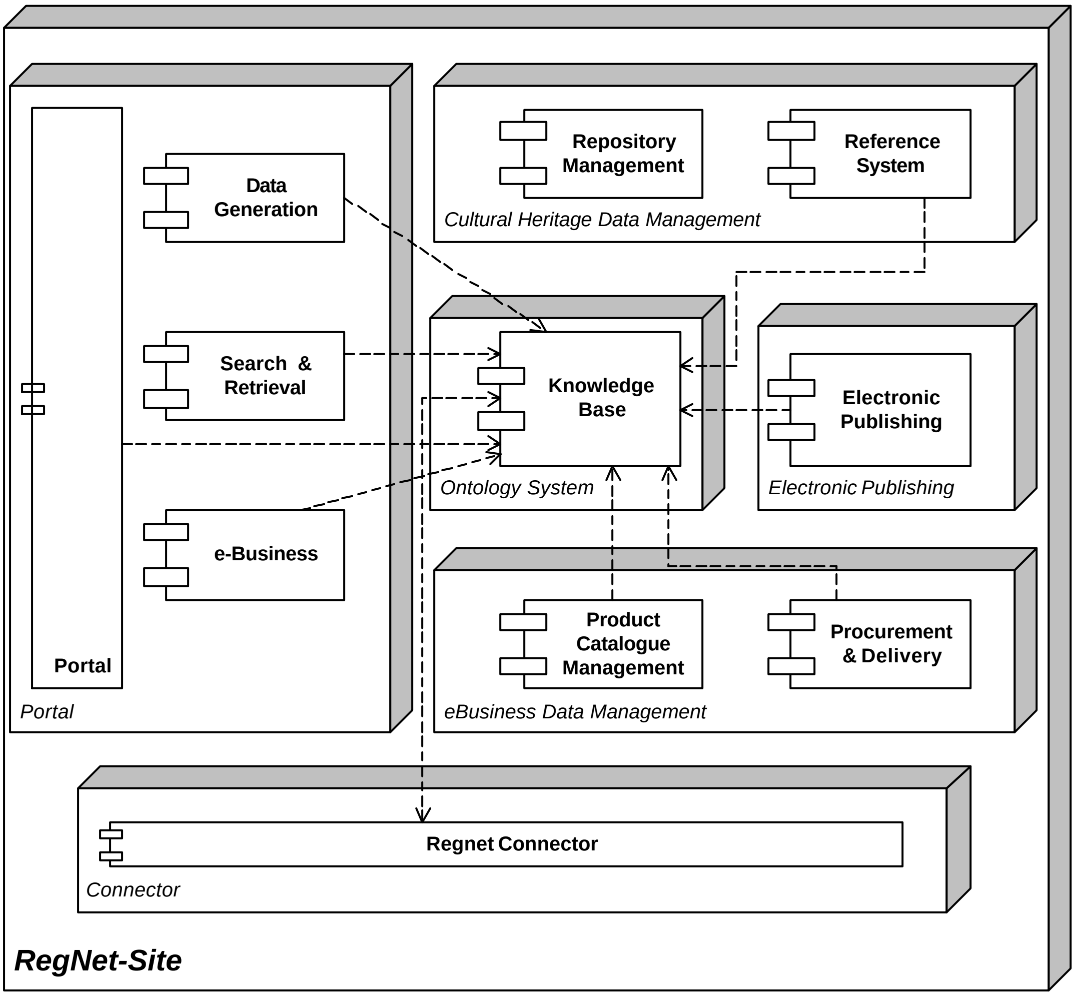


Figure 73: Deployment Diagram - Knowledge Base Access

The following diagram shows in more detail the interconnections between the Ontology System and the Portal, Electronic Publishing.

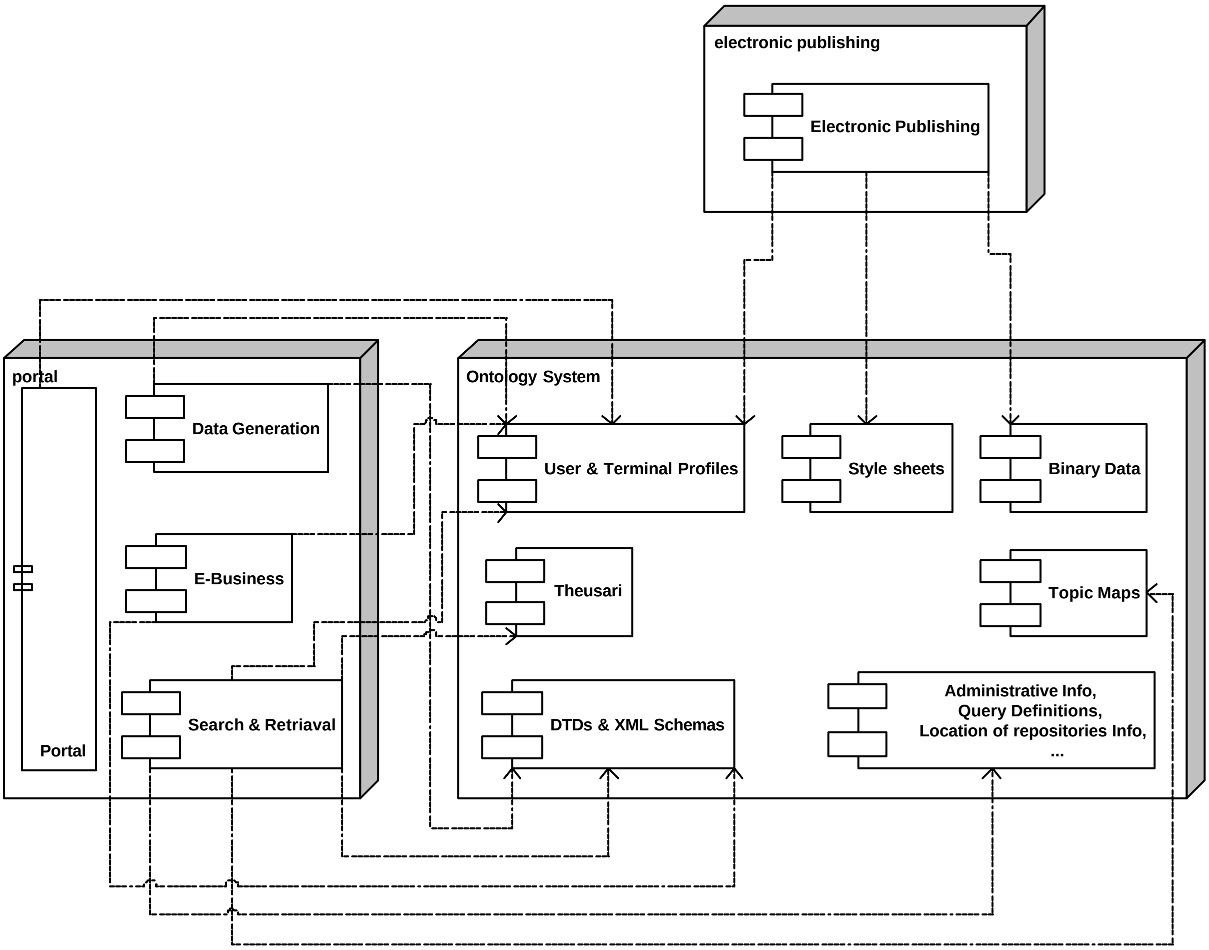


Figure 74: Deployment Diagram – Ontology System (connection to Electronic Publishing and Portal)

The following diagram shows in more detail the interconnections between the Ontology System and the e-Business Data Management, Cultural Heritage Data Management.

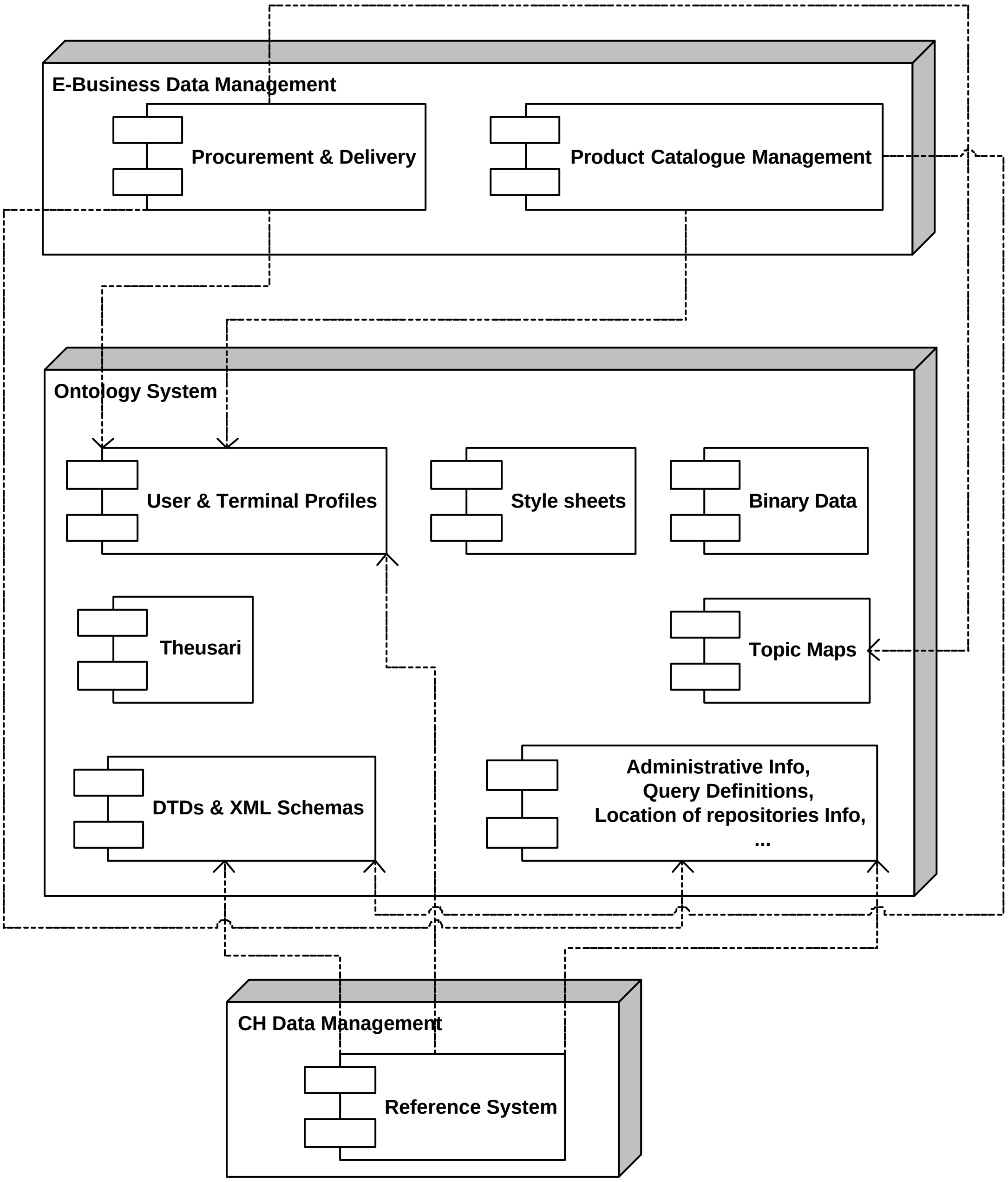


Figure 75: Deployment Diagram – Ontology System (connection to e-Business Data Management and CH Data Management)

All connections between the various nodes and the Ontology System are based on SOAP or RMI.

5.5.6 Electronic publisher

5.5.6.1 Purpose of the subsystem

The main objective of the Electronic Publishing Subsystem (subsystem 8) of the REGNET System is to provide an easy to handle tool for the creation of publications based on the search results gained by a query within the Search System of the REGNET Portal.

The User who wants to do the publishing will be guided through the publication process, which leads either to a final output or to a publication prototype which then can be further refined with editing tools for final production.

This final output or publication prototype will be a document representing the artefacts the user has chosen. This document can be used as template or suggestions for further processing or refinement, which can be done by an expert publisher.

Growing amount of data demands for being organised. Richard Saul Wurman founded out that every information can be organised by one of five categories: by **L**ocation, by **A**lphabet, by **T**ime, by **C**ategory, (i.e. the way department stores are organised), and by **H**ierarchy, (from the largest to the smallest of something, from the reddest to the lightest red, from the densest to the least dense, and so on). These are summed up by the acronym "**LATCH**". Good Information architecture should imply this principle therefore the publishing prototype will support organising data in this five ways. Adding a storyboard to the ResultSet implies structuring data with LATCH. Two of the categories: Time and Location will be implemented in the "Choose storyboard()" functionality. The C category will be implemented using the XTM 1.0 specification of Topic Maps whereas topics are equal to themes in this context.

5.5.6.2 Overview of the required functionalities

By analysing the user requirements and use cases (See 7.5.7) done by the content group we found the following main functionalities that should be supported:

- Create a catalogue of the selected records (a collection)

  Example: A museum wants to have a printable version of it's latest collection.

  Example: An artist creates a catalogue of all his art work for merchandising.

- Publish a theme

  Example: A CD-ROM of a special theme (created by a content provider) should be produced

- Create an exhibition

  Example: A museum wants to produce a catalogue about a specific exhibition

- Create a virtual gallery

  Example: A museum creates a Web-site prototype or even a SMIL presentation of its most famous artefacts to enhance its publicity.

- Create a printer friendly version

  Example: A museum creates a printable version of selected artefacts.

5.5.6.3 Users of the EP module

Users of the Electronic Publishing Subsystem will be all Content Providers participating in the REGNET System (e.g.: Museums, Archives, Libraries, Art Galleries and Artists). The functionality of the Electronic Publishing component should not be restricted to special products, but should provide a broad variety of the different needs of all user groups.

Both a novice should be supported to get simple outputs, as well as more experienced users should be able to use the Publication prototype for further processing.

5.5.6.4 Publication process

The publication process is a complex workflow consisting of combined tasks that finally leads to the required publication. This publication can either be an end product, that will not be further edited or a prototype that can then be sold as new good in a shop. If users want to add new products to their shops in this solution of REGNET, they will have to enter them via the Data Generation Subsystem. Nevertheless the architecture of the EP module should allow a direct workflow with automatic reduction from the Electronic Publishing System to the repository of the shop to be implemented easily.

In the process more subsystems of the REGNET system are involved and are co-operating.(See table below) Tasks like searching for the objects to be published, getting some refined ResultSets according to the user's preferences are functions provided by other components. The management of the subsystems is done by the Electronic Publishing System and by the Portal, which have the overall control of the whole publication process.(e.g. Session Management, User tracking, etc.)

Business Process	Workflow	Subsystems
Publish	Search, Edit, Create	Search and Retrieval, Ontology Subsystem, Electronic Publishing, Repository Management

The Electronic Publishing System is delimited from the other subsystems through it's interfaces and its capsulated functionality.

5.5.6.5 Detailed description of required functionalities

According to the functional requirements we define the following detailed functions for the Electronic Publishing Subsystem.

5.5.6.5.1 Select records

The records, each represented by a unique identifier, are displayed in a window, and the user can choose the objects to be published.

The records can be scrolled up and down and the user can jump to the beginning and the end.

5.5.6.5.2 Add records

The users can go back to search functionality and do another search and add this ResultSet to the already received ResultSet. The ResultSet is also sent to the Electronic Publishing System and appended to the original ResultSet.

5.5.6.5.3 Select elements of a record

The fields of each record type (AMICO, EAD, TEI) that is supported will be shown and can be selected by the user. If the users select a field, they want it to be included in the publication prototype.

5.5.6.5.4 Choose storyboard

In order to sort all records or to provide a navigational aid, the user can choose a storyboard. A story board according to the LATCH definition can be chosen. The following storyboards will be supported directly:

- A theme based storyboard
- A time line storyboard
- A geographic storyboard

If the user has chosen to publish a theme, the Topic Map takes over the role of the storyboard. The objects are then categorised according to the Topic Map. A timeline and a geographic storyboard will be provided according to the underlying data.

A timeline will be supported by a scrollbar at the bottom of the HTML file, where the objects that the user wants to see will appear or be selectable, whenever the right time period has been chosen.

In a printable version the objects will be sorted by time of creation.

To support the users with a geographical storyboard the objects will be represented by single spots on a map, which can be zoomed in and out. For the first version we suggest to provide a map of Europe.

5.5.6.5.5 Create web-site (prototype)

An HTML file can be created, where the user can than navigate through either by a timeline or select objects by clicking on spots on a map. The Electronic Publishing System can also provide documents of a publishing prototype that can be further edited by other tools.

5.5.6.5.6 Create printer-friendly version

The creation of a PDF output will be provided, that can printed or be viewed by Adobe Acrobat.

5.5.6.5.7 Create CD-ROM prototype

A simple HTML page can be created.

5.5.6.5.8 Return to e-Business

If the users want to end their publication activities, they can return to the main e-Business screen and proceed with further activities.

5.5.6.6 Optional functionality

These functionalities will not directly be implemented within this project but should be considered when developing the detailed system architecture.

5.5.6.6.1 Save ResultSet and selected records (optional)

The ResultSet and the chosen records and record fields can be saved.

5.5.6.6.2 Export (optional)

The publication prototype can be exported to the shop and used for further e-Business activities.

5.5.6.6.3 Import (optional)

Saved ResultSets and chosen record and record fields can be imported for further use.

5.5.6.7 Required Functions of other Subsystems

5.5.6.7.1 Search for artefacts

The User can create a query to find the desired objects. All museums, archives and libraries connected to the REGNET system are searched through. This functionality is provided by the Search and Retrieval Subsystem. All settings and enhancements will be done there.

5.5.6.7.2 Search for themes

The User can search for themes. In order to retrieve the structure and all the objects included in a theme a special search mechanism must be provided by the Search System. The theme result will consist of the Topic Map structure in XTM 1.0 format followed by the ResultSet.

5.5.6.7.3 Refine search result

If the ResultSet is to rough, imprecise or to big, the user can refine his search by setting some parameters or restricted his current ResultSet. This functionality will also be provided by the Search and Retrieval Subsystem.

5.5.6.7.4 Refine with personal profile

The ResultSet can be refined or enhanced by using the Personal Profile of a registers user managed by the Ontology Subsystem

5.5.6.7.5 Get Suggestions from Ontology system

The ResultSet can be refined or enhanced by asking the Ontology system to suggest related objects or domains.

5.5.6.8 Scenario

The following scenario describes the whole publication process (See 7.6.6.4)

The user can start the publication process through the e-Business system.

In the search form of the Search and Retrieval System the user can perform a REGNET query. If the results of the query are not satisfying, the user can send a request to the REGNET Ontology Subsystem. Via the Ontology Subsystem another REGNET query is being performed and the results are added to the ones from the proceeding query. Having finalised the search process the results of the query are saved and handed over to the Electronic Publishing Subsystem.

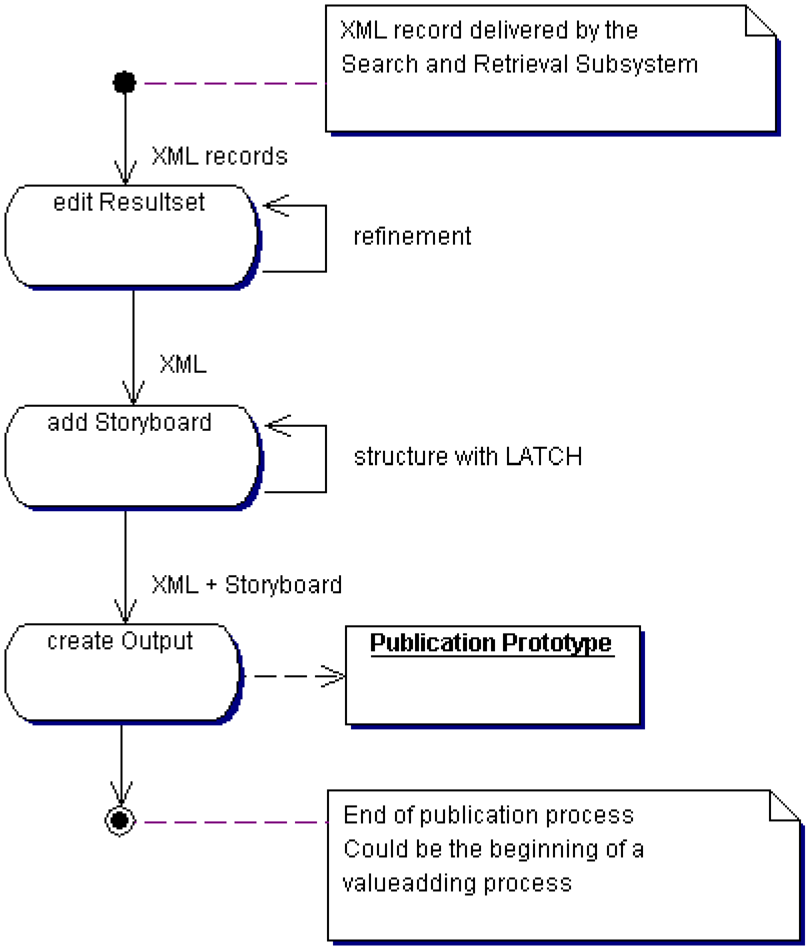


Figure 76: Activity Diagram - Electronic Publishing

The above figure shows the main tasks of the EP Subsystem, that follow the tasks done by other Subsystems (Search and Retrieval, Ontology)

The metadata of the located artefacts are displayed in one or two lines for each record. The results can be scrolled through and edited in a simple way. In the following, the term "artefact" is used for the digital surrogates of 'primary' (real world).

By the help of a "storyboard" the artefacts can be edited. In "storyboard"-mode thumbnails are representing the artefacts. The thumbnails are created already during data generation and saved with the REGNET metadata. These thumbnails can be drawn of the storyboard, placed between two other thumbnails respectively placed in a pool, from which they can be fetched again.

Optionally the storyboard supports a geographical view. In geographical view spots are arranged on a map depending on the location of their repository (the location where one has to go for visiting the artefact in nature). With this option a user could get a quick overview of how the artefacts are spread geographically. If there are too many results in one location, only some representatives and the quantity of results is displayed. The map can be scrolled in so that one is able to get a more detailed view of the query results.

Besides editing the geographical view can also be the basis for navigation through the produced publication (e.g.: a CD ROM, a Web Site).

The storyboard mode offers several options for pre-setting the way the artefacts are sorted. It is possible to sort the results according to a timeline and to the distance of their repositories from a geographic point.

Selected Filter-options, Storyboard and publication options can be saved for easy reuse. (Will not be provided in the first version)

If refinement of the data is finished the user generates the preferred publication. The main presentation forms will be included. Technically this will result in either a SMIL- presentation, a PDF-doc or an HTML page. More advanced users will be able to receive a publication prototype (which consists either of an XML and the corresponding XSL file or an HTML file), which can be used for further processing.

5.5.6.9 EP Component Framework

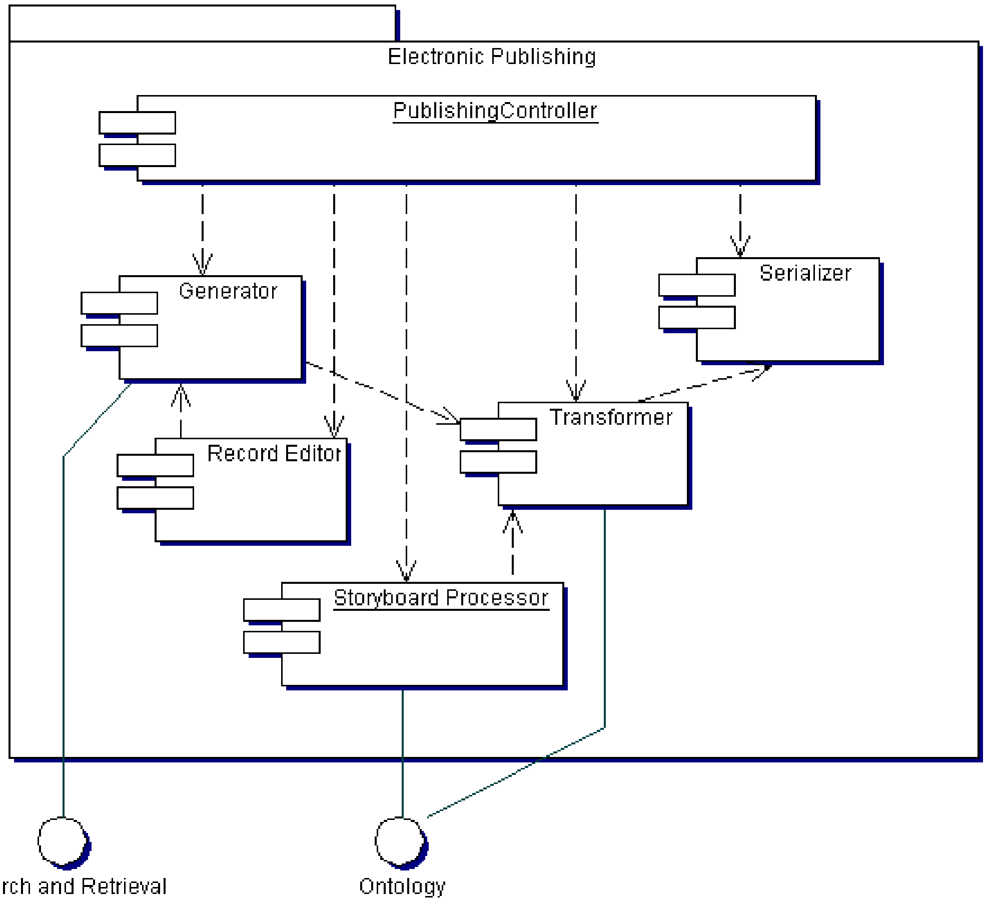


Figure 77: Component Diagram – Electronic Publishing

- Generator Component:

The Generator component generates an XML structure from the input source (Resultset of the Search and Retrieval Subsystem).

- Record Editor

The Record Editor component manages the editing of the single records to be published.

- Storyboard Processor

The Storyboard Processor retrieves the storyboard from the Ontology Subsystem and structures the result.

- Transformer

The Transformer is the core component handling the XSLT transformations also using the Ontology Subsystem as Stylesheet repository.

- Serializer

A Serializer is used to render an input XML structure into some other format like HTML or PDF.

- Controller Component

The Controller Component handles the overall connection and the control of all other subcomponents.

5.5.7 REGNET Connector

The REGNET connector provides functionality to connect Cultural Institution / Organisation (CIO) and Cultural Serices Centers (CSC) with one another. This connection is done according to ebXML initiative that standardise data and message between stakeholders.

Multiple protocols can be support by this connector in order to address all functional need for communication. Nevertheless ebXML standards will be used as far as possible and available. In particular Z39.50 will be provide for distributed search.

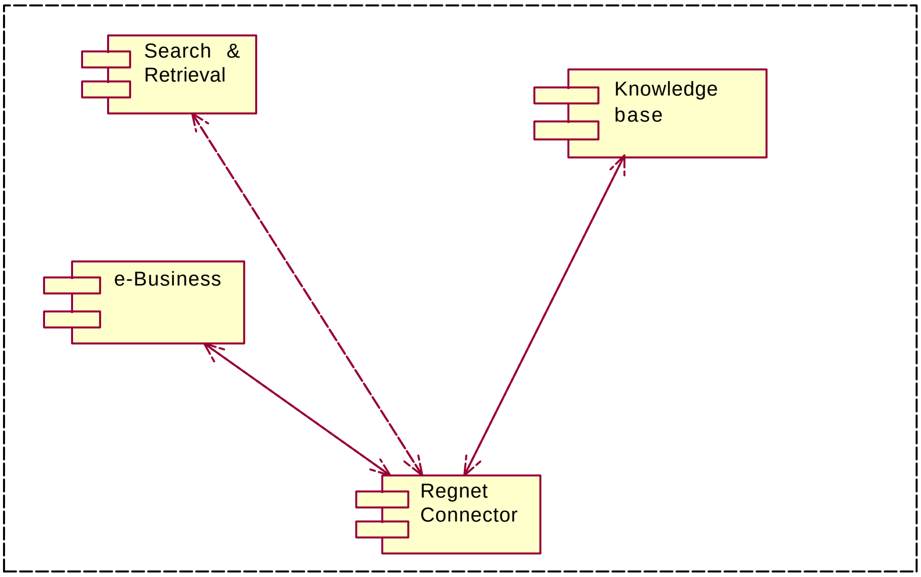


Figure 78: Component Diagram - REGNET Connector

The link between Search & Retrieval and the REGNET Connector is used to search distributed CH data catalogues, managed by cultural organisations (CSC or CIO).

Links between e-Business and the REGNET Connector are used for communication between e-Shops or procurement systems. An example of such a communication is the negotiation for exchange of materials.

Links between the Knowledge base and the REGNET connector are used by the connector to access user profiles. These profiles store information about CIOs and interested users. Moreover profiles store security information which is propagated by the connector.

5.6 Technical architecture

The technical architecture represents software building blocks that are used in order to implement the suggested functions (they can be seen as a projection of the functional architecture). Two aspects are address by the technical architecture. They define the components of a system and their interaction. The technical architecture consist of:

- the decomposition of a complex system into building box.
- the construction of the REGNET system from the building box.

Decomposition: building a complex system from simple components:

> "At the heart of each functional complex system, you will always find a simple system that works... Not only do complex systems designed *ex nihilo* never work, but you cannot patch them to make them work. You have to start from scratch, from a simple and effective system."
>
> -- J. Gall, *How Systems Work and How They Fail*

Integration: building a system which acts as a whole:

> "Conceptual integrity is the essential factor in the design of systems. It is better to have a solid system which is lean on functionality than a more complete system based on disparate and inconsistent ideas. Employing a system architect is a decisive step toward attaining conceptual integrity." -- Frederick Brooks, *The Mythical Man-Month*

This part describes the REGNET technical architecture from both, the engineering and technology point of view. It's aim is to address technical aspects of the REGNET design. It describes technologies and tools which will be used in order to implement the functionalities listed before. The engineering point of view represents an architecture which is technology independent whereas the technology point of view describes the software tools used.

This part builds on state-of-the-art technology which is detail in the annex. Technologies must be chosen carefully weighting the risks of the adoption and integration of new technologies. Technical risks associated with the architecture are given as well as corrective measures which are detailed in the software development plan.

In all considerations we have to distinguish between the architecture of the client and the architecture of the server.

5.6.1 REGNET network architecture

As described before, REGNET is a Network of Regional Cultural Service Centre.

The ontology subsystem is at the heart of REGNET network architecture. It embeds the common knowledge useful for all REGNET nodes. There is only one Ontology database which is managed by the REGNET master node. This database is periodically replicated from the master node to other REGNET nodes in order to share the knowledge across the network. Moreover, only the master database may be modified.

According to the observer design pattern (see state-of-the-art for details), relationship between Ontology system ("model") and other subsystems ("views") must respect constraints:

- Views modify model in a synchronous way.
- Model notify views of a modification in an asynchronous way.

5.6.2 REGNET server architecture

REGNET server architecture must provide an open environment in order to integrate technologies and tools. It must be based on standards in order to facilitate change management.

A first schema displaying the dispatch of functional modules on a distributed system is shown by the above figure:

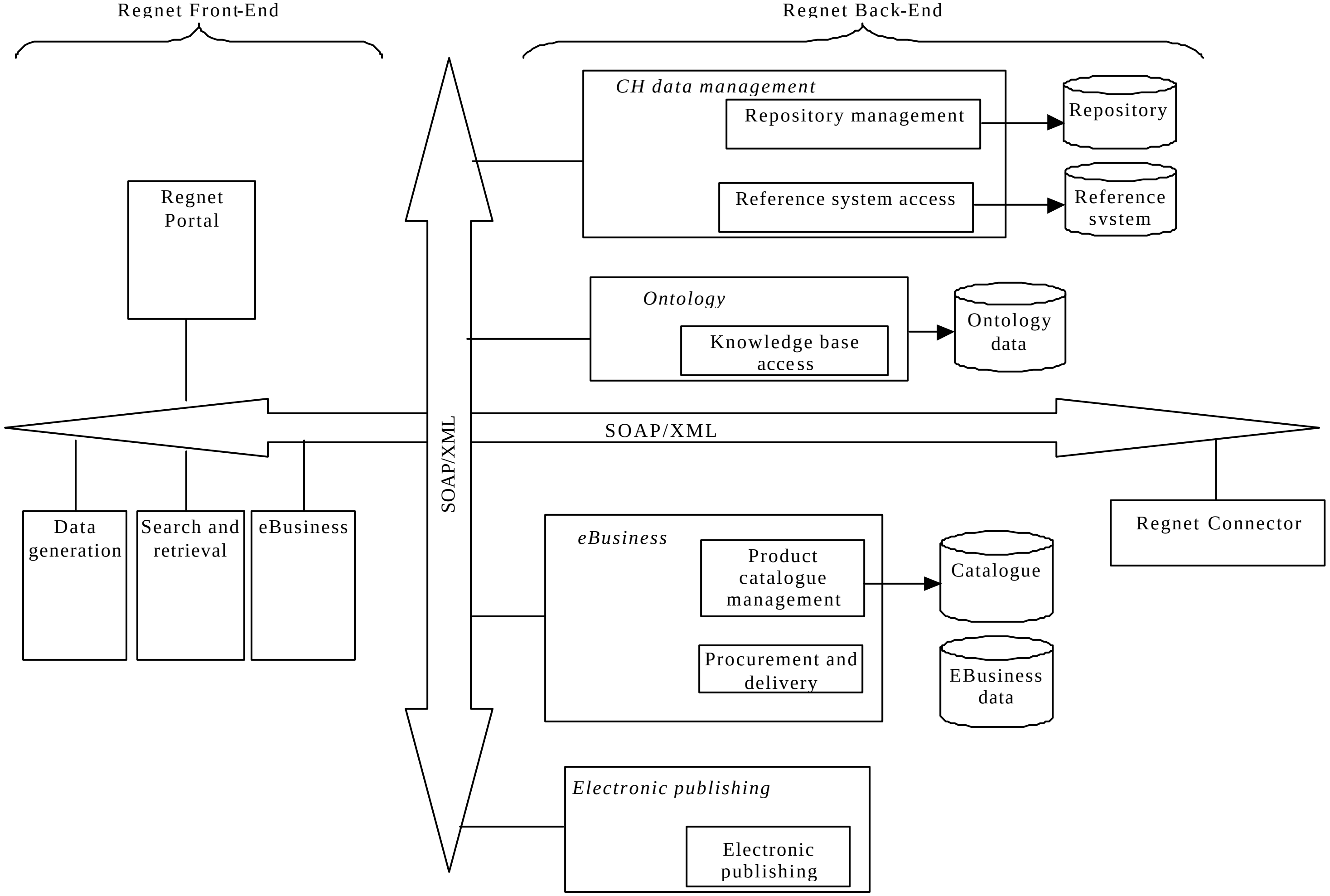


Figure 79: REGNET Server architecture

This schema illustrate REGNET overall technical architecture. This architecture is a Web Services approach:

- REGNET Front-End contain functionalities which are user accessible. These functionalities provide access to the back end through the middleware.
- Middleware is based on SOAP/XML. This tools which are discuss in the state-of-the-art part provide the necessary glue between REGNET subsystems and between REGNET nodes.
- REGNET Back-End provides data centric functionalities.
- REGNET connector is the necessary component allowing communication between nodes.

A critical aspect of REGNET architecture is Integration. As mentioned before, we have to integrate heterogeneous technologies and human skills into a distributed architecture. This architecture must provide good capacities in terms of evolving, scaling and management. In order to build such a system we must based REGNET on a component based framework which accept heterogeneous components. Web services approach provide such framework. The set of concepts and technologies that are related to Web Services are:

- XML: to describe information.
- UDDI: to find the necessary services.
- WSDL: to describe how Web Services work.
- SOAP: to remotely execute Web Services.

Each module provides a WSDL interface which allow others modules to request services. Services are invoked through SOAP protocol.

Detail about Web services are given below. Tools are available for PHP and JAVA. This approach provides necessary technologies in order to build REGNET middleware. However we must take care that these components share data. This aspect is relevant for the choice of the database technology which must support transactional mode.

Following sub paragraphs will detail each element.

5.6.3 Connections between modules

Following tables illustrate connections between modules.

Abbreviations:

Nodes

Nr	Name	abbreviation	responsible partner
1	Repository Management	RM	SPAC
2	Reference System	RS	AIT
3	Knowledge Base Access	KB	SI
4	Data Generation	DG	SPAC
5	Search & Retrieval	SR	AIT
6	e-Business	EB	ZEUS
7	Product Catalogue Management	PCM	ZEUS
8	Electronic Publishing	EP	SR
9	Procurement & Delivery	PD	VALT
10	Portal	PO	MOT
11	REGNET Connector	RC	VALT

Table 46: Abbreviations for system modules

Connections:

A cross between a line and a column indicate that the line component request a service from the column component.

		1	2	3	4	5	6	7	8	9	10	11
		RM	RS	KB	DG	SR	EB	PCM	EP	PD	PO	RC
1	RM											
2	RS			x								
3	KB											x
4	DG	x	x	x		x		x				
5	SR		x	x				x				x
6	EB			x		x		x	x	x		x
7	PCM		x	x								
8	EP	x		x		x						
9	PD			x					x			
10	PO	x		x	x	x	x					
11	RC			x		x	x					x

Table 47: Connection between system modules

5.6.4 REGNET client architecture

From the functional and technical requirements, the need for different kinds of clients can be derived:

- Heavy client:
    - Web browsing
    - Acquisition
- Light client:
    - Web browsing
    - Wireless client (WAP, J2ME enabled)
    - Clients enabled to Voice-Access-To-Web technology (wireline and wireless phones). Note that the technology to integrate these kinds of browsers will be presented in sections below for the sake of completeness, but it will not be integrated into the demonstrator due its low requirement priority.

As far as possible, common functionalities between these four categories will be factorised in the server side. However, there are some specificities notably for acquisition client.

5.6.4.1 Web acquisition client

This client is a part of the data-generation module. It provides tools in order to get data and metadata from user and use them into the Regent Server. These information are used by the e-Business module in order to initiate an auction or by the CH module in order to add new elements.

We have to take care that, as far as an upload operation is necessary to carry information from Internet into the REGNET system, security mechanisms must be placed in order to verify data.

Schema of such a client is given below:

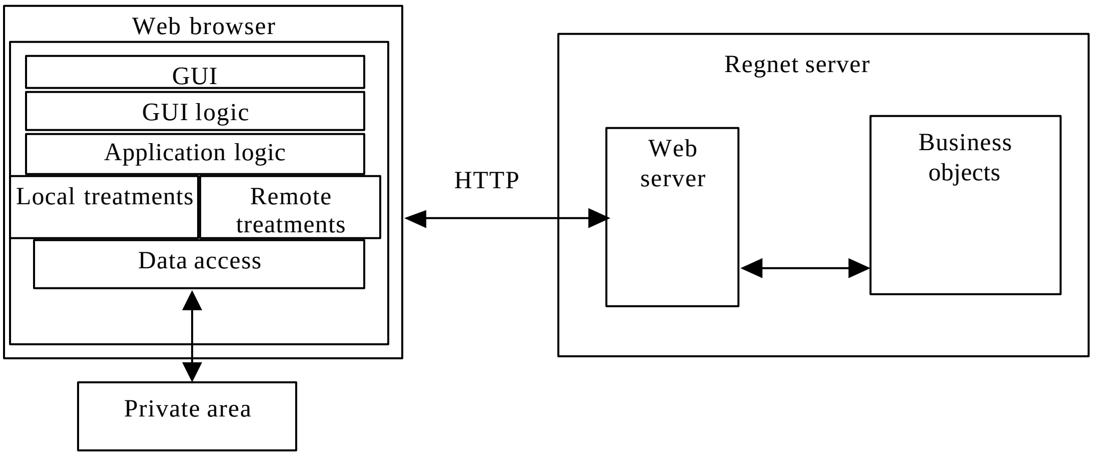


Figure 80: Acquisition Web client architecture

Explanation about business object (figure 80)

This client is a heavy client so we have to take car not to mixed presentation and applicative treatment. So a logical tree-tiers architecture must be define for the client side.

This client, access to a private space which is local to the computer. User manipulate data from this space and activate the upload only when data are ready. This architecture allow users to work offline the network.

This client uses HTTP in order to communicate with the REGNET server through the Web server. Despite of its low performance, use of HTTP based protocol (SOAP) allows firewall skipping.

This client can be easily develop in JAVA as an Applet. JAVA provides very sophisticated APIs (Swing, JAVA 2D and JAVA 3D) allowing to develop portable automatically deployed client GUI. Data can be upload to the Web server as XML serialised JAVA objects.

Risks associated with this architecture are:

- Heavy client: we have to avoid classical risks associated with client/server architecture. It means that we must embed into the applet only the necessary business logic. Furthermore we have to define a three tier architecture for the client in order to separate GUI and application logic.
- JAVA default security sandbox, prevent Applet to access local resources (e.g. Disk). In order to allow disk access for the applet, we have to use trust (signed) applet and modified local java security parameter.
- Applet download require time and local installed JAVA Virtual Machine with the right version. This problems are solve by using the JAVA plugin (http://java.sun.com/products/plugin/index.html) or JAVA WebStart (http://java.sun.com/products/javawebstart/index.html). These technologies are available for free from Sun.

JAVA plugin allows to use any version of JVM instead of the web browser's default virtual machine and provide Applet cache.

JAVA WebStart allows to build JAVA client which can automatically:

- Detect, install, and use the correct version of the JAVA Runtime Environment for the application.
- Launch the application from the browser or the desktop.
- Download newer versions of the application as they become available.
- Cache classes used by the application locally for fast start-up.
- Run as either an applet or an application.

- Download native libraries if necessary.
- Use local resources, such as the file system, in a secure way.
- Locate and load external dependencies.

5.6.4.2 Light client enabled to Web services

Light client allows navigating across REGNET Web site in order to access all functionalities except acquisition. This client interacts with the Search system (subsystem-5), E-business (subsystem-6) through the Portal, and with Knowledge Base Access (subsystem-3) to create/modify/delete its user profile.

5.6.4.2.1 Web browsing client

Web browsing client is connected to the REGNET system through a HTTP based Web server. As secure transactions have to be guaranteed in e-Business services, also HTTPS is used. Besides, to protect confidential data (credit card number, amount to be charged, etc.) some cryptographic process has to be defined.

In the following figure, it is depicted a web client – server architecture.

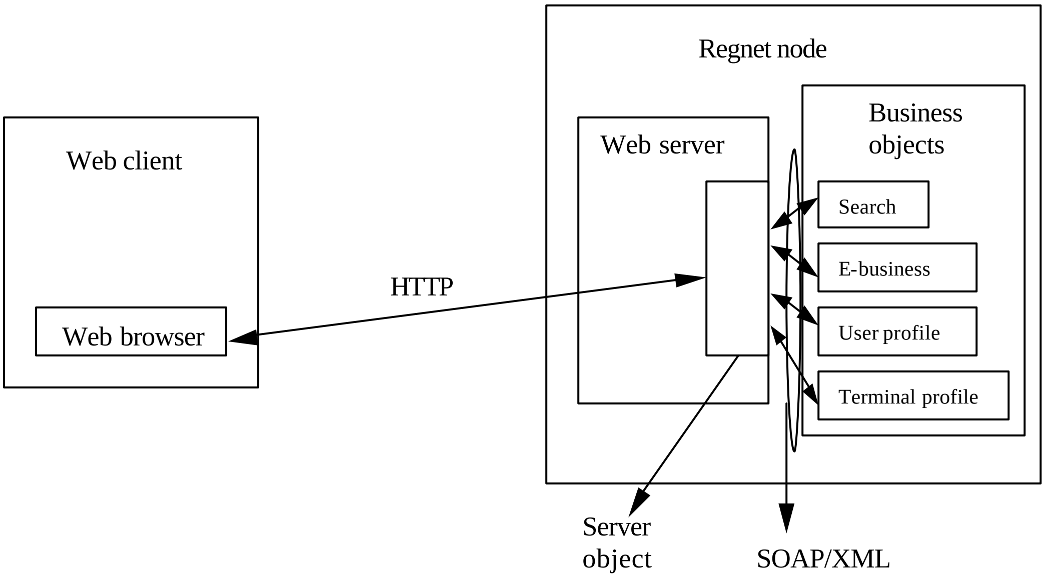


Figure 81: Web client-server architecture

This client is based on a standard web browser and it navigates HTML pages dynamically generated by the server, containing some client-side scripting code. Advantages of such architecture are obvious in term of deployment and standardisation.

Risks with this kind of client are:

- To use non-standard client-script language functionalities (e.g. JavaScript outside ECMA standard - available only for last browsers version).
- To mix business rules in client-side scripting routines.
- To make presentation logic complex in such a way that it becomes very hard to maintain.

5.6.4.2.2 Wireless client

This kind of client is characterised by all the wireless terminals that can access Web services: mobile phones, PDAs, etc.

There are mainly two technologies in which this client can be deployed:

- WML
- J2ME

In each framework, the client is able to access a REGNET node: it connects to a web server and makes requests to a server object. The web server is responsible to dispatch the requests to the business objects communicating via SOAP and XML.

Below, the two frameworks, applied to REGNET, are depicted.

1. WML

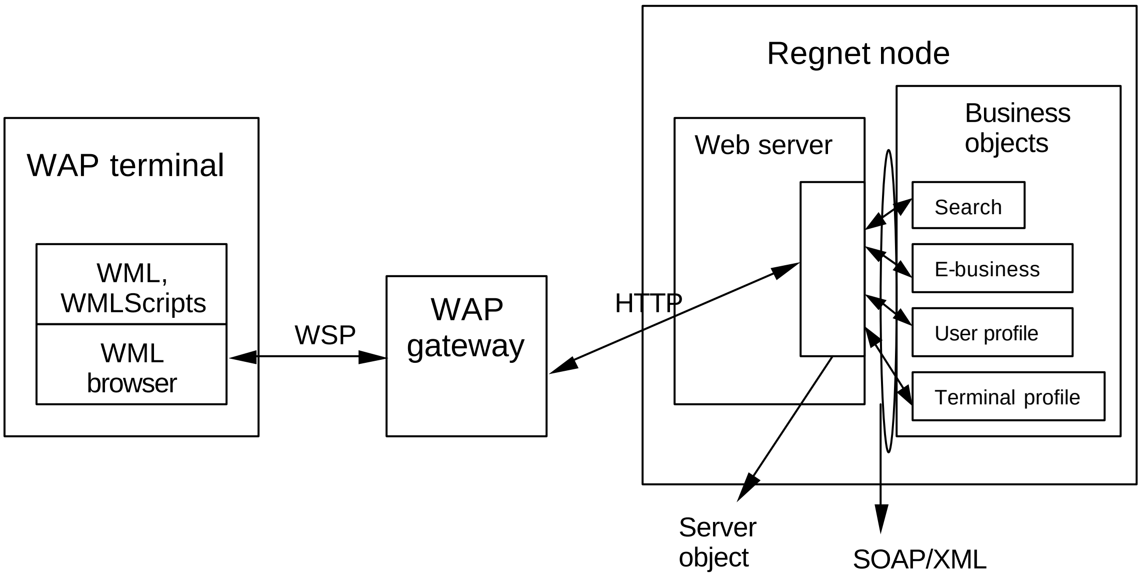


Figure 82: WML client

The user-agent connects to a REGNET node, and the system provides him some contents shown in a WML deck. The communication between user-agent and WAP gateway is based on WSP requests that are translated in HTTP requests towards the REGNET node, in order to be processed by a server object inside the Web server.

The pros are that the user interface is very light and fast, and cons are that it has limitations due to a scripting language like WML: few GUI resources (only lists, hyperlinks), no colours.

2. J2ME

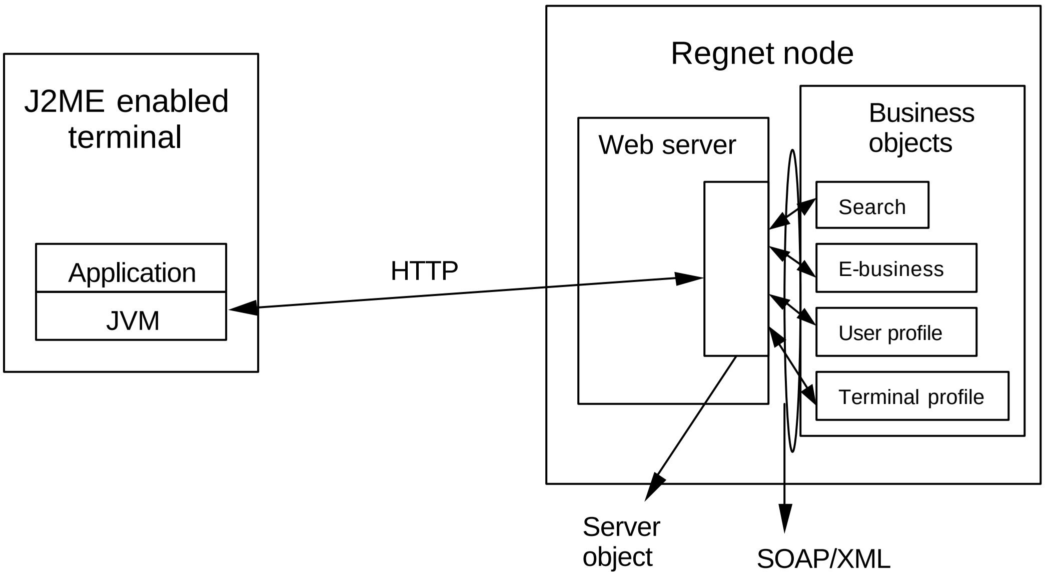


Figure 83: J2ME client

A JAVA application is installed on the terminal, and the client makes connections to a REGNET node. The communication is based on HTTP requests, using JAVA API, that are processed by a server object inside the Web server.

The pro's are that on the terminal there is an actual application, so the user interface provides more sophisticated GUI resources (check-boxes, radio buttons, buttons, etc.). Con's are that, like all java applications, they are less light and require more dynamic memory.

5.6.4.2.3 Voice-Access-To-Web client

The design of REGNET architecture let to integrate a Voice Access system. While this open the opportunity to access the offered services with the most common kind user terminal (a voice telephone, both wired and wireless), it complicates the infrastructure needed in the server side. For the sake of completeness, and to show further potential of the system, this technology is described below, even if it will not be part of the demonstrator.

A voice-access-to-web client is able to make a call from any phone (wireline or wireless) and retrieve information from the Internet or perform transactions. It is based on voice application technology.

The communication between the client and the system is based on a sequence of interaction dialogs:

- Dialogs to recognise the user (user profile)
- Dialogs to make the user able to ask information to the system
- Dialogs to make the user able to use e-Business services
- Dialogs to render the response from the system

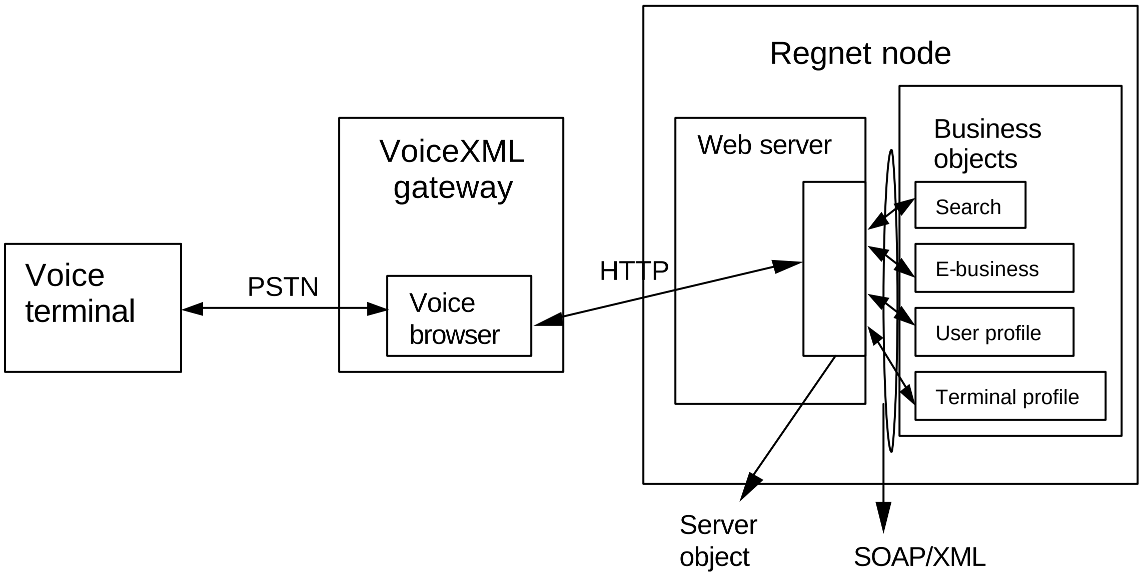


Figure 84: Voice-access-to-web client

The client uses the voice to ask requests to a REGNET node. The call is answered by a computer system at a VoiceXML gateway site. The gateway system retrieves the initial VoiceXML script from a VoiceXML content server, which can be local to the gateway or located anywhere on the World Wide Web. A portion of the VoiceXML gateway called the interpreter parses and executes this script playing prompts, hearing responses and passing them on to a speech recognition engine that is also part of the gateway system. When the script has collected all the necessary responses from the user, the interpreter assembles them into a request to the VoiceXML content server. The content server replies with a dynamically generated VoiceXML page containing the information requested by the user. The process can be repeated indefinitely to produce the appearance of a conversation between the user and the VoiceXML server. Substantially, the gateway serves as an HTTP client sending messages and receiving VoiceXML pages from the web server. Communications between the gateway and the content server follow standard HTTP protocols. Outgoing requests are in the form of an HTTP "get" or "post" command.

The pro of this technology is that it uses the most natural means of communication of the mankind: the voice. Moreover, in order to access to voice applications is sufficient any phone (wireline or wireless). No specific equipment is needed (e.g. WAP browser).

In spite of this, there are some weaknesses due to:

- Different kinds of pronunciation, the system does not always recognise the user speech;
- This technology has not yet achieved a naturally way of speak with the user;
- Voice applications offer information to the users in a serial way: isn't possible to present information to the user in a parallel way (e.g. forms containing text fields, etc.). This could annoy the user;
- Voice applications haven't persistence. On the other hand, in visual applications once presented, data remain on the display until replaced by new data or cleared by the user.

5.6.4.3 Security

Regarding security issues in communication between the user-agent and the REGNET node, the major distinctions are made based on the kind of access, whether it is wired or wireless, and based on the presence of a gateway.

5.6.4.3.1 Wired Access

This is the case of the classic Internet access, where the security is demanded to HTTPS protocol (this will be the solution adopted by REGNET). Additional mechanism to improve security shall be added at application level. See connected business object specifications for further details.

5.6.4.3.2 Wireless Access

GSM and GPRS radio access supply the same mechanism of security protection over the air of user identity and data. UMTS provides security functionalities as well. The WAP stack includes the security layer WTSL, but it utilisation depends from network operator service offer so REGNET will not use it in the first phase.

5.6.5 REGNET Portal

Portal provides interface between end user and REGNET system. It belongs to the presentation layer of the application.

Communication between the Portal and clients is SOAP based.

Communication between the Portal and others nodes is:

- SOAP in heterogeneous context (e.g. JAVA – PHP or PHP – JAVA)
- RMI in homogeneous context (JAVA only).

Data transmit between Portal and back-end elements are XML encapsulated. Portal uses information provided by Ontology system in order to generate presentation according to user profile.

5.6.5.1 Presentation Layer

It is the front-end for client request and system services. User agents generate requests that contain different kind of information, among which terminal characteristics (not mandatory), user preferences, and user data (e.g. POST). This data are collected, and the appropriate resources are identified to serve the requests. From a functional point of view, we call these presentation resources as Presentation Modules.

Client Requests

- URLs
- Data entry
    - Update REGNET repositories (data generation)
    - Supply form information (search system, e-Business)

Web Server Responses

- Generic pages – not rely on the three main Portal subsystems
- Services specific pages - rely on the three main Portal subsystems (data generation forms, search interfaces, results pages, etc.)

Presentation Modules include the capability to parse XML data arriving from the other server-side modules, and to translate this information in the client specific language. This task is performed in a combined approach based on client-specific Navigation Modules, and client-generic Navigation Modules that rely on XML Stylesheet Transformation. They perform also filter functions to tailor contents according to user profile and terminal profile. Presentation Modules contain only the mechanism to achieve filtering and tailoring of contents, whereas the capability of information to be modified in such a way must belongs to content itself. Namely, the information itself must contain the redundancy to address various terminals and user preferences.

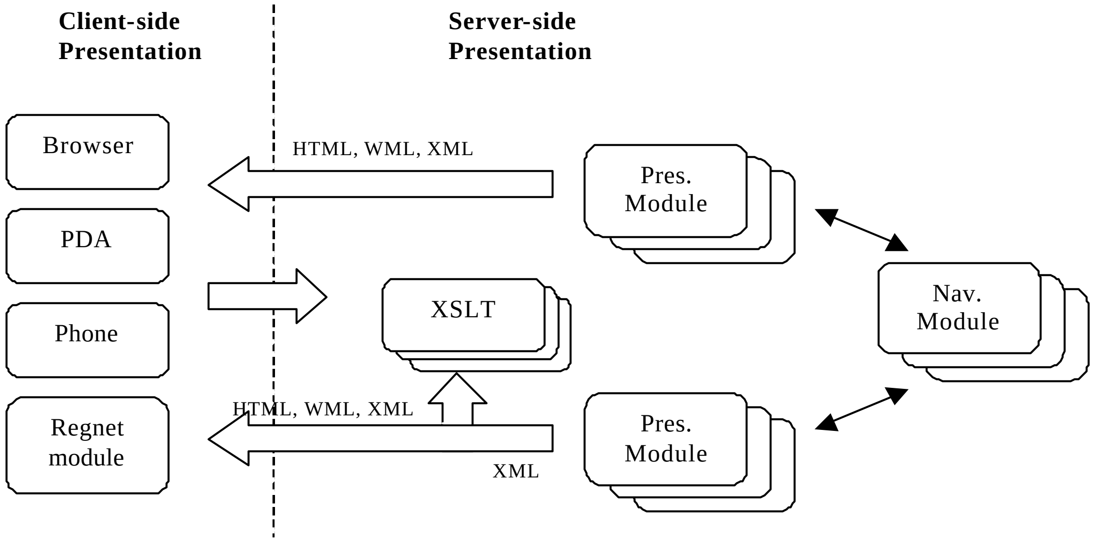


Figure 85: Portal Presentation Layer

Navigation Modules contains the logic to bind the pages presented to the user. This approach permits to separate the navigation information from contents (e.g. “back” is content related, “back to a specific URL” is a navigation information) thus simplifying the maintenance. However, more important, it allows a differentiation of navigation paths depending on user preferences and terminal capabilities.

5.6.5.2 Applicative Logic Layer

This layer contributes to create a consistent environment for user navigation, at a single navigation level, and a subsequent navigation level.

- Single navigation
  - Detection of the user terminal capabilities and configuration (Session Manager). This information is communicated to User Terminal Manager to identify the more matching profile among those handled by the system
  - The Session Manager detects user’s regional attributes
  - The Session Manager stores all the information generated during the navigation
- Subsequent navigation
  - The Session Manager interacts with User Profile Manager to exchange user preferences
  - The Session Manager recalls all persistent information generated in previous navigations (e.g. outstanding orders)

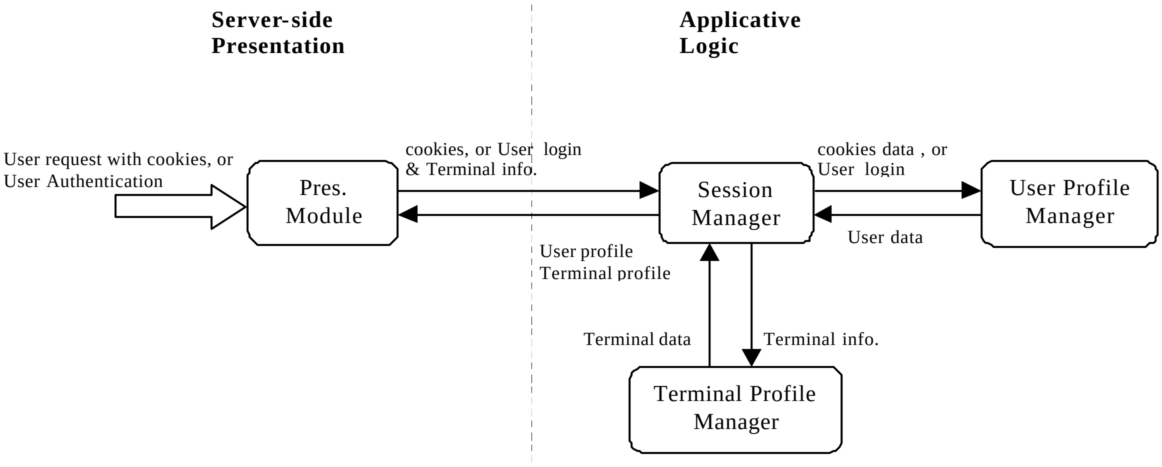


Figure 86: Portal Applicative Logic Layer

The user identification may occur after an explicit authentication procedure. The Session Manager always adapts contents to the user according to user terminal, regional, and preferred attributes. Moreover, it allows the user to work in standalone mode, that means the user client can interrupt the link to the server and connect again as it was a unique session. This is essential for wireless clients that may go out of coverage during a session.

The User Profile Manager stores user's regional attributes and all users' preferences, gathered during successive navigations. User profile may strongly affect the contents delivered to the user, in terms of:

- Filtering
- Translation

The User Profile Manager may rely on a User Profile module outside the Portal node, since a User Profile module could serve also other component of the system, for monitoring tasks, information exchange with external entities not through the Portal, etc.

The Terminal Profile Manager stores all the profiles of the terminals correctly handled by the system. It may be possible to conform another type of terminal to one in the predetermined set, if there is a good degree of compatibility. Profiles may be added along the time, with minimal impact on other system modules. Terminal profile may strongly affect the contents delivered to the user too, in terms of:

- Filtering
- Translation

5.6.5.3 Business Logic Layer

Three main modules reside in this layer: Data Generation, Search System, and eBusiness. They interface with other server-side Portal's modules and other REGNET's node in the system. Their main functionalities are illustrated in the following sections.

Since these modules can be implemented with different technologies, it is essential to take into account an interface design that can deal with them, moreover that is not limited to a fixed number of technologies, being open at future extensions. This will be obtained using WSDL paradigm; the data exchanged are XML based, and consist in:

- Information to build user interface for each service
- Information from the user making use of the service
- Information generated to serve the user

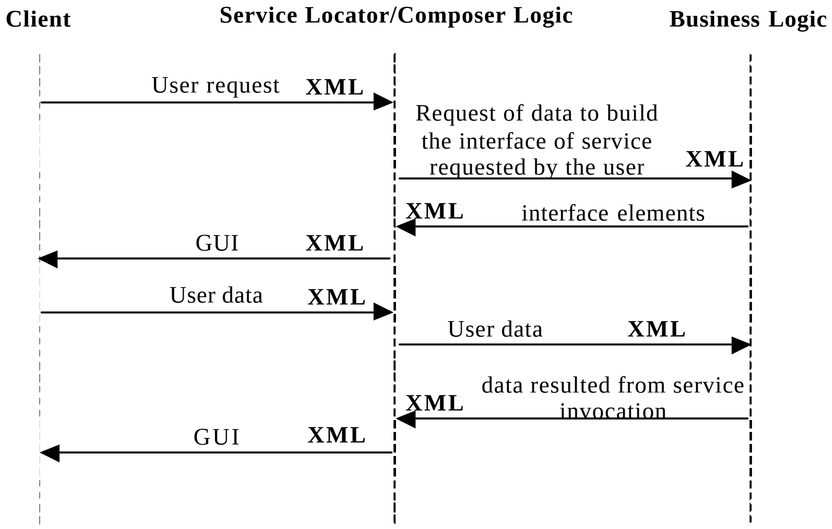

Client
Service Locator/Composer Logic
Business Logic
User request XML
Request of data to build the interface of service requested by the user
XML
XML
interface elements
GUI XML
User data XML
User data XML
XML
data resulted from service invocation
GUI XML

Figure 87: Interaction Client-Portal-Business Logic

It is also possible that, for the sake of improving performances or reducing the deployment complexity, the Portal contains specific interface's components for some Business modules. An example is the communication between Portal and Search & Retrieval module. In this case, the Portal supplies the search query to the S&R module. This implies that the Portal "knows" the syntax required by the S&R Query Translator.

5.6.5.4 Data generation

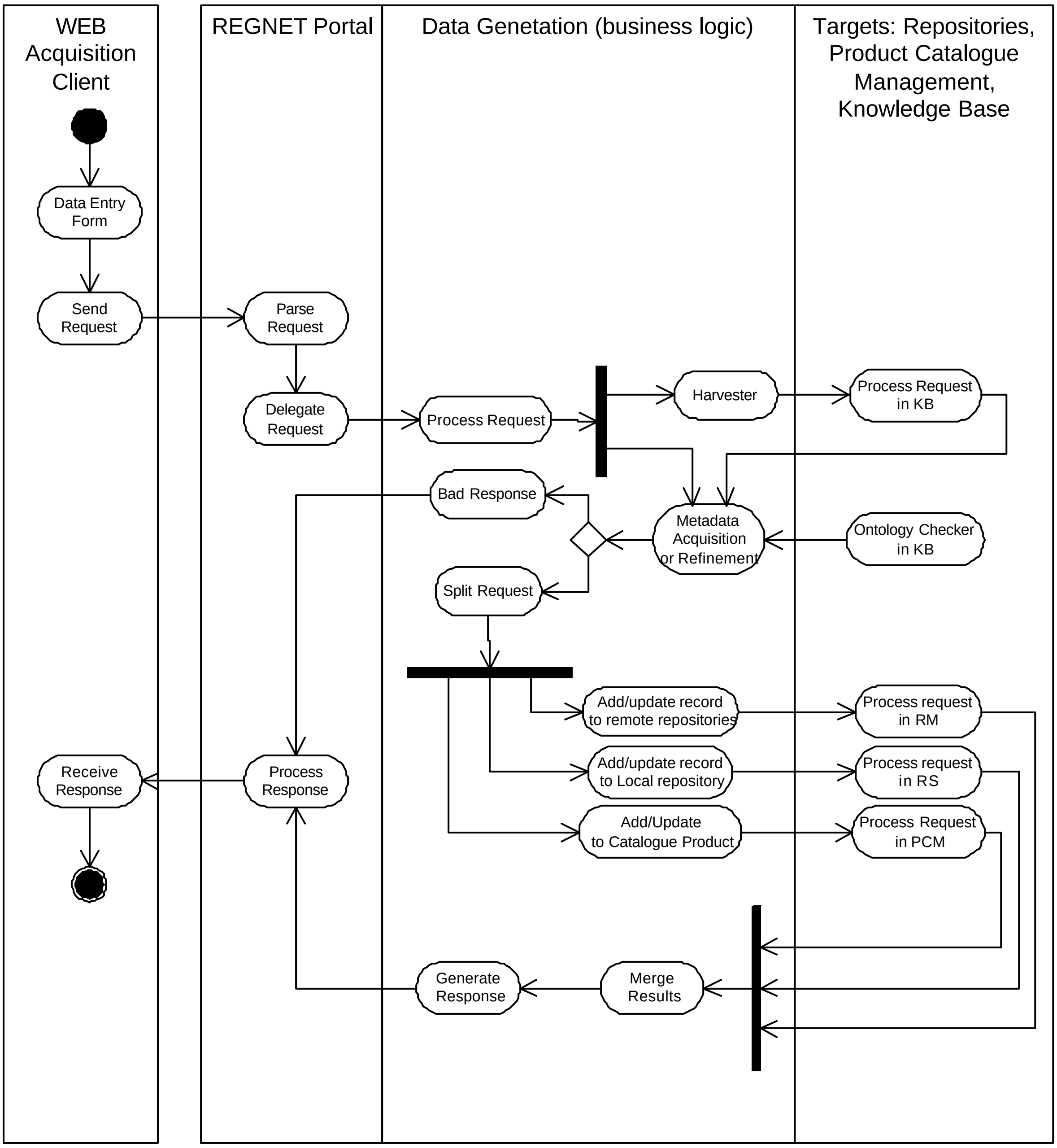


Figure 88: Activity Diagram – Data Generation

The Data Generation (subsystem-4) is a component belong to REGNET Portal node. Like others entry point of the REGNET's Portal its aimed at support REGNET's metadata contributor.

Creating correct metadata is like library cataloguing, except the creator needs to know more of the technical information behind the data in order to properly document them.

The form for maintaining metadata will depend on a number of factors:

- the size of the data holdings
- the size of an organisation
- the patterns of data management within an organisation

Probably REGNET partners who produce contents to be treated in the system, not be worth their time to fully learn a metadata standard. Instead, they might be asked to fill out a less-complicated form or template (Data Entry Form) that will be rendered in the proper format by a data manager or cataloguer who is familiar with the subject and well-versed in the metadata standard.

Following term lists and recommendations, the Data Entry Form permits the creation of high quality metadata to carry into XML documents.

The Data Entry Form displays a specific set of metadata elements with the available metadata values. Some mandatory elements could include Identifier, Title, Description and Subject. Where metadata is not present, values for these elements will be obtained according to the default rules that will be established.

The generation of the Data Entry Forms will be driven by XML-Schemas/DTD's (defining the data formats used within the REGNET System). Additionally support from the thesaurus system located in the Ontology System may help the users with their task of creating meta data.

When the request, coming from the WEB acquisition Client, reach the Portal node (on the WEB Server side), it is processed and the appropriate module is called: in our case the Data Generation component.

The request is processed and with the help of the Knowledge Base system his validity is checked.

If the request is valid it is subsequently split and sent to the appropriate system that execute the adapted action (for instance add a new record in the local repository), and getting back the positive response.

As the data holdings become larger and the access to the data becomes distributed, then organisations would look at more advanced methods for maintaining metadata of their data holdings.

These advanced tools may consist of commercial or self-developed systems that may extract aspects of the metadata automatically from the data itself: this is the approach of most of the various harvesters such as AltaVista and the like.

The harvester visits a document, reads it, and returns back resource description, such as every weeks or months, to look for changes.

The resource description continuum represents compromises among cost, ease of creation and maintenance, and utility.

At one end there is the virtue of simplicity: harvesting is done with no concern for the domain, the structure, or the purpose.

The scale of the collection and delivery of the information is challenging, but the description model is simplistic: word-level indexing. Search result sets are generally very large; high recall, but typically very low precision. Even with such coarse-grained retrieval, the harvester results can be useful.

5.6.5.4.1 Themes

The data generation part of REGNET has many facets and varies from scanning and OCR of paper based collection items, over getting data via on line Z39.50 connections, to specific fragment data entry and topic map generation.

- Object images and descriptions can be generated from three sources and delivered to REGNET in two ways. The sources are: scanning of paper based descriptions and images, OCR-processing and translation into Dublin Core of descriptions, extraction out of existing collection management systems with eventually translation into Dublin Core and direct dgital production of data (word processing and digital camera). There are two ways to deliver these data into REGNET: indirectly via storage media (floppy, CD, DVD) or directly via uploading (FTP).

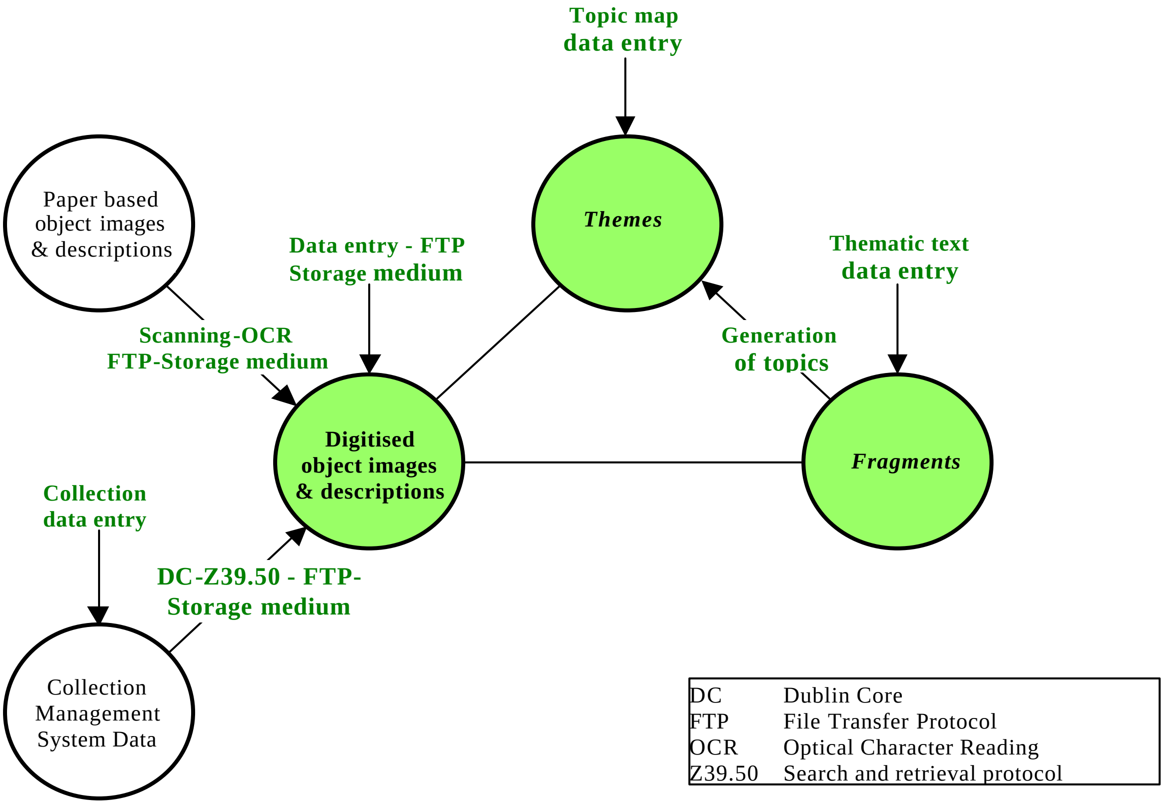
- Whereas collection objects images and descriptions pertain by their nature strictly to the objects themselves (see the equivalence with the standard collection management systems fundamentals), fragments are newly set up texts delivering supplementary and contextual information related to an object or to a set of objects. Moreover, these fragments are set up in a way that they can be reused for different contextual views and situations. The content of the template has to be XML-tagged for further exploitation.

Fragment name	title
Language	EN for English, NL for Dutch, …
Audience level	generic, children, students, scientific, …
Size level	short or full
Author	
Contribution	
Date	
Modified by	
Modif. date	
Modif. description	
Copyright	
Cost	
Content	Text with embedded references A reference is a number: e.g. (1) = reference to link 1
Links	(1): description of an action and/or URL of resource referenced by (1) (2): description of an action and/or URL of resource referenced by (2) …..
Primary keywords	most relevant
Secondary keywords	second order
Number	digital object identifier

Table 48: Fragment data set lay out

A particular characteristic of fragments is that they can also be used for object descriptions by applying the Dublin Core notation within the content field of the fragment. This Dublin Core-XML –tagging will then be embedded into the Fragment-XML-tagging.

A theme is a collection of thematic texts and/or objects that have one or more characteristics (“keywords”) in common. One can easily understand that with this approach the theme concept is a quite flexible one and that a single thematic text can show up in different contexts. Themes can either refer to thematic texts only, or to collection objects only or, in most cases, to both, thematic texts and collection objects. The right use of context and scope attributes can narrow the result to the exact level that a user requires. The topic map paradigm seems to fit extremely well to fulfil this requirement. In order to generate topic maps, a special data entry tool is needed to do this manually. One could also consider to automate this process by referring to the keywords entered in the fragments.



DC	Dublin Core
FTP	File Transfer Protocol
OCR	Optical Character Reading
Z39.50	Search and retrieval protocol

Table 49: Cultural Heritage data entry

5.6.5.5 Search and Retrieval

As already mentioned there are several components that can access the search functionalities in the Search & Retrieval component. In general we have three functional groups:

a) search: all kinds of search requests, scan/browse functionality, use of thesaurus

b) save: saving queries and refining them

c) information: get information about the S&R component (search status, statistic of usage, …)

To handle the requests from the portal (and other sources) a light layer of applicative logic will be used to distribute the requests to the right sub-component. This layer might be implemented with the JAVA Servlet technology.

To obtain an easy interface for all clients it is necessary to use protocols and data formats which is platform and implementation independent. The proposal to use SOAP was accepted by the technical partners and it satisfies both goals because SOAP uses HTTP as protocol and XML as data format.

Nevertheless the Search & Retrieval component should be accessible via RMI also as the Portal and some other components will be implemented in JAVA and RMI is a more natural way to communicate between pure JAVA applications.

For processing the requests a business logic layer will be used. The main components of this layer will certainly be a "Request-Processor" component and a component for administrating queries and result sets. The business logic layer will be completed with some components for information handling (point c) above). The plan is to implement the business logic components with Enterprise JAVA Beans technology.

5.6.5.5.1 Search Request (Search)

Next a first draft of a search request XML-Schema will follow. This is subject of change for the end of the Development of Concept phase (WP1) and the beginning of the Implementation of the System phase (WP2).

The search request will be structured in 3 parts:

- A General section: information about the user/component that send the request, interface language, …
- The Query section: specify the targets (repositories) and of course the query
- The Response section: preferred result format, sorting criteria, number of records to return, …

General

This section holds general information about the request source. One can specify the UserId, the Source of the request (e.g. Data Generation, Portal, Site-CSC Austria, ...), the SessionId, the Result-Set-Id (if no Result-Set-Id is specified in the Request a new Id will be created by the Search & Retrieval component; Result-Sets than can be accessed via this Result-Set-Id) and an interface language.

Query Format

The Query Format should be the same independently from which source it is. That means that it must be general enough to cover the needs for search request for all components which can access the S & R component (Portal, Data Generation, e-Business, Electronic Publishing, REGNET Connector). It would be possible use the Z39.50 protocol for this reason but that puts heavy implementation effort to each subsystem and requires better knowledge of the protocol for each component provider. The other thing is that the Portal – as presentation layer – should not deal with such a protocol as this task belongs to a business logic part.

Nevertheless parts of the protocol (the BIB-1 query) can be used to put on the query format on a common standard.

Result Format

The Result Format can be specified in one part of the search request. In general there will be two result formats available:

- native (that means that the native records with their own format will be returned)
- cross domain (e.g. Dublin Core)

Further it might be useful to have different options for each result format defining handy sizes for the result. Let us assume that there are 3 sizes available: short, medium, complete.

That means if a client wants to receive Dublin Core, short he will get only DC-Title and DC-Creator and DC-Subject fields (DC stands from Dublin Core), but if he wants to get Dublin Core, complete all Dublin Core fields will be returned. The same applies to other native formats. Of course a mapping has to be done which tells the system what fields to take from one format to deliver the right size.

SearchRequest XML-Schema:

```
<?xml version="1.0"?>
<xsd:schema targetNamespace="http://www.REGNET.org"
                           xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                           xmlns="http://www.REGNET.org"
                           elementFormDefault="qualified">
       <xsd:include schemaLocation="xer.xsd"/>
       <xsd:element name="SearchRequest">
              <xsd:complexType>
                     <xsd:sequence>
                            <xsd:element ref="General"/>
                            <xsd:element ref="QueryDef"/>
                            <xsd:element ref="Response" minOccurs="0"/>
                     </xsd:sequence>
              </xsd:complexType>
       </xsd:element>
       <xsd:element name="General">
```

```
                <xsd:complexType>
                        <xsd:all>
                                <xsd:element name="UserId" type="xsd:string" minOccurs="0"/>
                                <xsd:element name="Source" type="xsd:string" minOccurs="0"/>
                                <xsd:element name="SessionId" type="xsd:string" minOccurs="0"/>
                                <xsd:element      name="ResultSetId"      type="xsd:string"
minOccurs="0"/>
                                <xsd:element   name="InterfaceLanguage"   type="xsd:language"
default="en"/>
                        </xsd:all>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="QueryDef">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element ref="Targets"/>
                                <xsd:element ref="Query"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="Targets">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element ref="databaseNames"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name="Response">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element name="sortSequence">
                                        <xsd:complexType>
                                                <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                                                        <xsd:element         name="SortKeySpec"
type="SortKeySpec"/>
                                                </xsd:sequence>
                                        </xsd:complexType>
                                </xsd:element>
                                <xsd:element name="NoOfResultsRequested"
                                                                 type="NoOfResponseRecords"
default="10"/>
                                <xsd:element name="PreferredResponsStruct">
                                        <xsd:complexType>
                                                <xsd:sequence>
                                                        <xsd:element      name="ResponseFormat"
type="ResponseFormat"/>
                                                        <xsd:element        name="ResponseType"
type="ResponseFormat"/>
                                                </xsd:sequence>
                                        </xsd:complexType>
                                </xsd:element>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name="NoOfResponseRecords">
                <xsd:restriction base="xsd:nonNegativeInteger">
                        <xsd:minInclusive value="1"/>
                        <xsd:maxInclusive value="50"/>
                </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name="ResponseFormat">
                <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="cross-domain"/>
                        <xsd:enumeration value="native”/>
                </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name=”ResponseType”>
                <xsd:restriction base=”xsd:string”>
                        <xsd:enumeration value=”short”/>
                        <xsd:enumeration value=”medium”/>
                        <xsd:enumeration value=”complete”/>
                </xsd:restriction>
        </xsd:simpleType>
</xsd:schema>
```

SearchRequest.xsd

5.6.5.5.2 Search Response (Retrieval)

A SearchResponse.xsd should be developed to fix the structure of a response to a search request. In the search response the following data may be included (discussion):

- ResultSetId (specified in the request or generated).
- HitCount (total and per repository).
- The original query (optional: for displaying the original query and the corresponding results).
- The result records.
- Additional information (e.g. failed search requests for some repositories, system failures, …).

5.6.5.5.3 Other Request (Retrieval)

For other requests belonging to the functional group a) the same procedure as for the search request has to be done. Other requests are:

- Sort: resorting the result set.
- Present: present the next records in the result set (i.e. next and previous buttons).
- Scan: scan repositories for provided term-lists.

5.6.5.5.4 Request Processor

The following activity diagram shows the principle activities for a group a) request (search, sort, present, scan). The requests will all be delegated to the Request Processor which is located in the business logic swim lane.

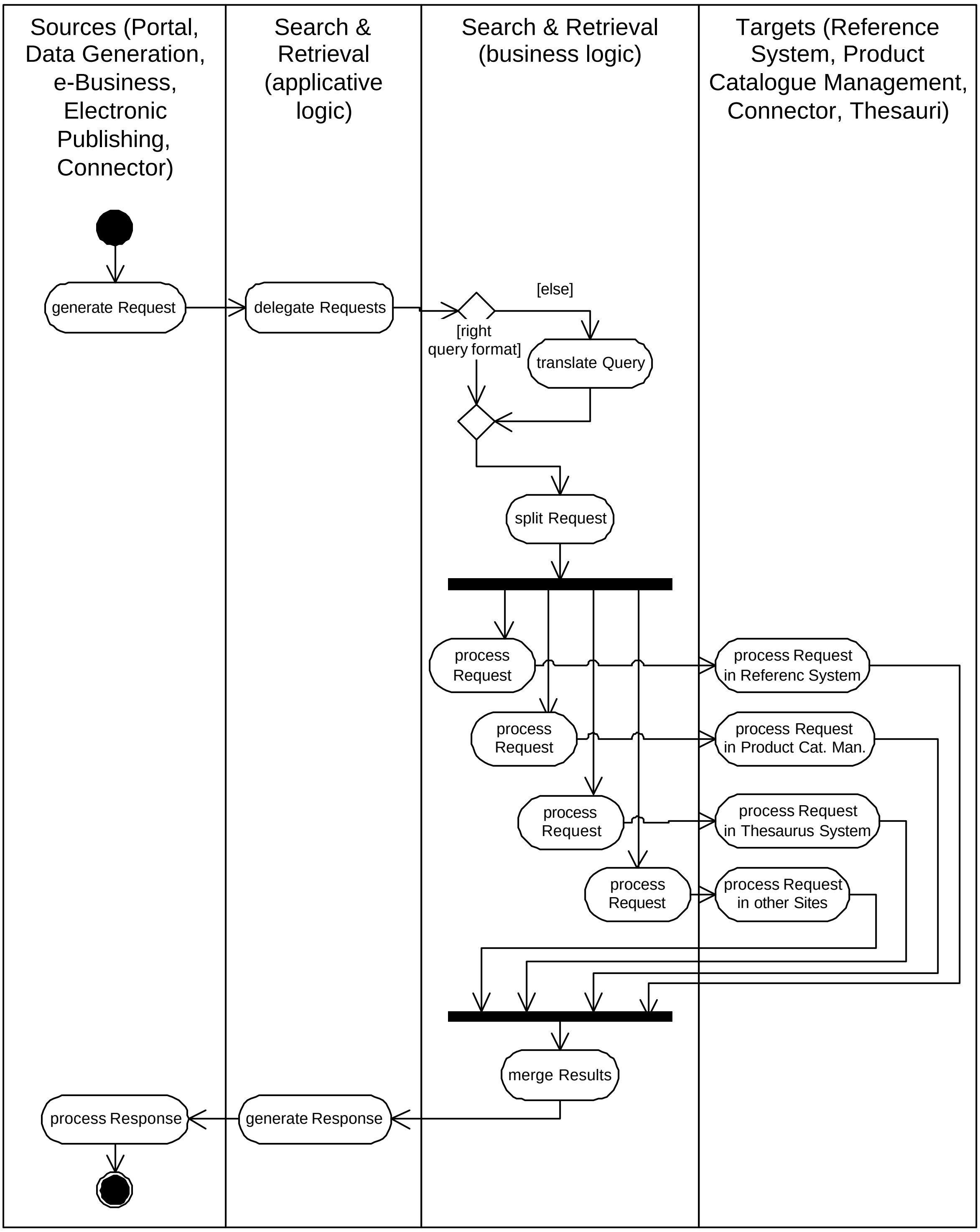

Sources (Portal, Data Generation, e-Business, Electronic Publishing, Connector)
Search & Retrieval (applicative logic)
Search & Retrieval (business logic)
Targets (Reference System, Product Catalogue Management, Connector, Thesauri)
generate Request
delegate Requests
[else]
[right query format]
translate Query
split Request
process Request
process Request
process Request
process Request
process Request in Referenc System
process Request in Product Cat. Man.
process Request in Thesaurus System
process Request in other Sites
merge Results
process Response
generate Response

Figure 89: Activity Diagram – Search Requests

5.6.5.5.5 Save Requests

Save requests are requests for:

- saving queries: this has to be done implicit for providing a query history and users can store queries also to a user space.

5.6.5.5.6 Informative Requests

These are all kind of requests to retrieve additional information about the Search & Retrieval component. A special sub-component will track all the statistical data and information about available repositories and responds to these requests.

5.6.5.5.7 Themes

The search and retrieval actions can be originated on three levels: objects, fragments and themes. Once one of these levels is entered, further searches can be done within the same level or shifted towards all other levels where more appropriate searches can be carried out. If the user has no preference, the search will be done on all levels and presented in the following priority:

- Themes
- Fragments
- Objects

In some cases, a specific search in the individual collection management databases could be offered by some of the content providers.

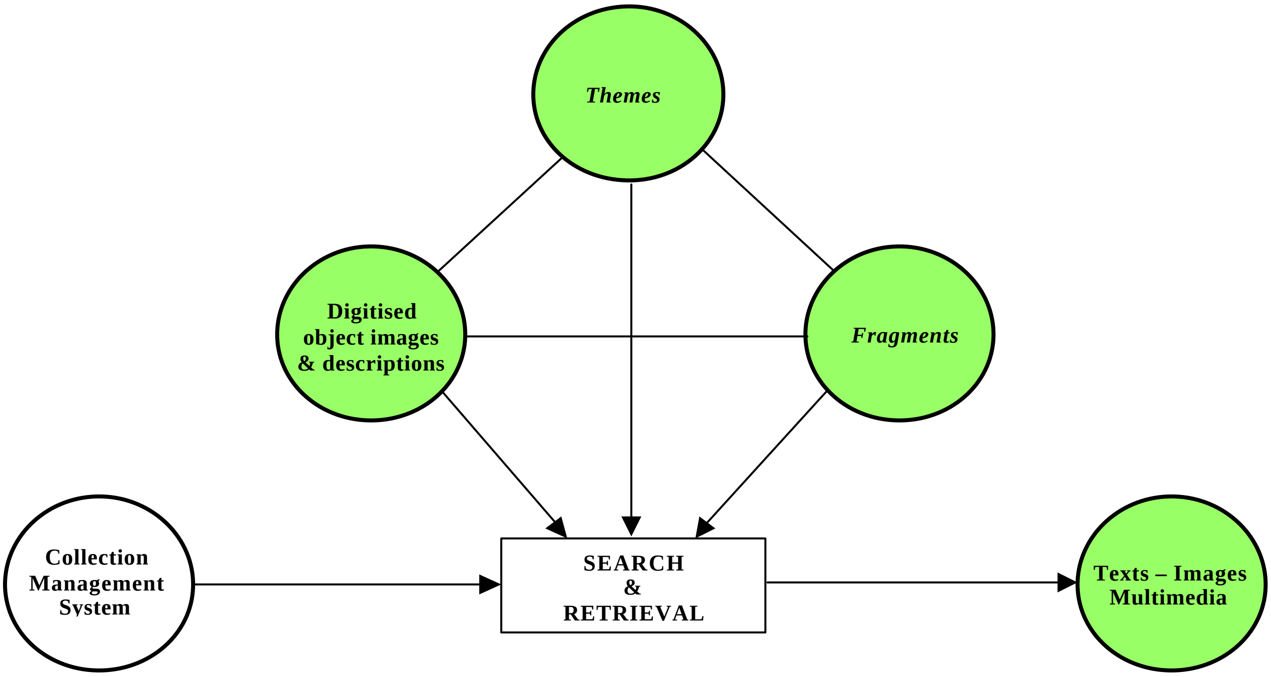

Themes
Digitised object images & descriptions
Fragments
Collection Management System
SEARCH & RETRIEVAL
Texts – Images Multimedia

Figure 90: Cultural Heritage data search and retrieval

5.6.5.6 E-business

The E-Business subsystem is a part of the portal node. Therefore it is obvious that most of the technical requirements of the specific subsystem will be derived by the requirements of the portal.

Even in this case there some points that need to be pointed out.

The E-Business part of the REGNET system is required to provide services to the users of the Internet. Therefore, the system will be implemented in an Internet oriented technology like PHP or JAVA. The reason for the specific constraint is the continuous interaction of this node with the end user and the inner part of the REGNET system.

The end user exploits the several functionalities that are being provided by the specific subsystem and the inner part of the system satisfies the requests from the E-Business subsystem.

In order to clarify everything the E-Business subsystem will be based on two levels of the system architecture. The interactions with the user will be satisfied by the Presentation layer of the REGNET system, but the core functionalities of the E-Business subsystem will be realised on the business layer.

The interactions of the E-Business subsystem will be clarified by the Web Services Description Language that will be used for the descriptions of the interfaces.

In terms of analysis and functional overview there is a clear distinction of the components that constitute the E-Business concept. The main difference lies on conceptual level where the user is able to take advantage of the E-Commerce system and the B2B system.

The following diagram gives a description of technical modules uses by the e-business subsystem:

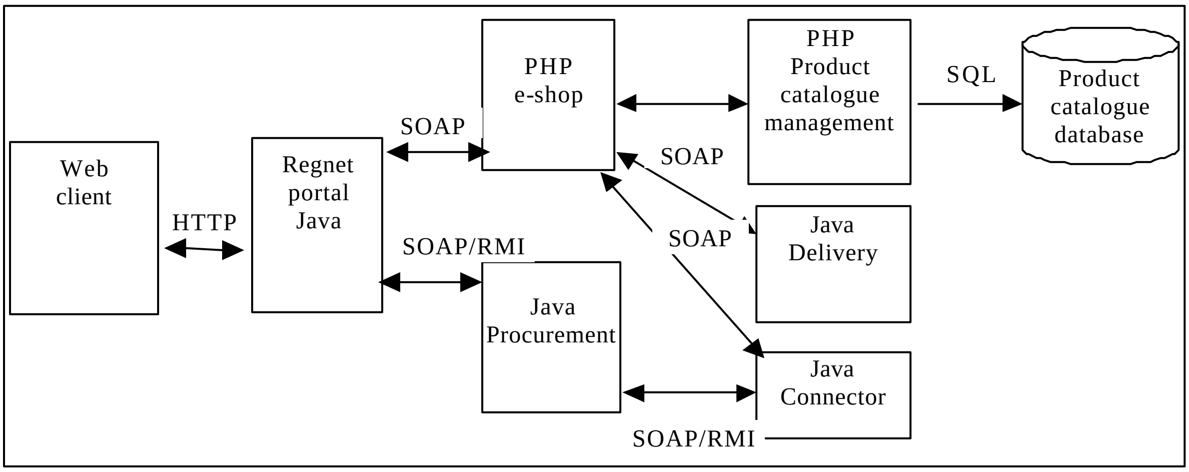


Figure 91: e-Business technical building blocks

5.6.5.6.1 Themes

In the eBusiness function of REGNET, the fragments and object images and descriptions can play two direct roles. The first one is that of material that can be sold, i.e. individual images and thematic texts and the second one as being a supporting medium for selling pieces of art and replicas. Indirectly a number of fragments and object images and descriptions will be sold through the paper based and electronic publications where they take part in. Flexible copyright measures have to be taken into account in all those cases.

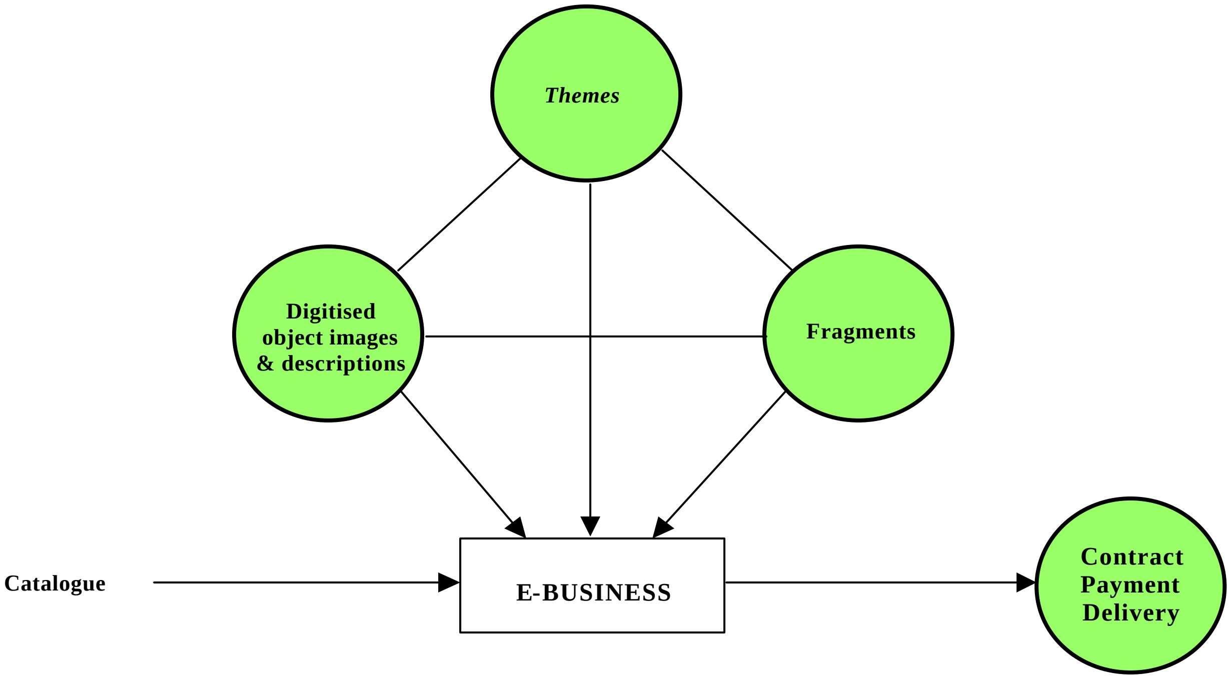

Themes
Digitised object images & descriptions
Fragments
Catalogue
E-BUSINESS
Contract Payment Delivery

Figure 92: Cultural Heritage data for e-Business

5.6.6 Cultural Heritage Data Management

5.6.6.1 Repository Management

According to Figure 63 the Repository Management is composed of 3 main components:

- Data Parser

  Interprets the incoming commands and executes the right actions on the repository. For instance it's the responsible for (e.g. add new data, update existing data, delete data etc.)

- Data Integrity Checker

  This component verify the consistency of data (XML format) to be inserted into repositories.

- Repository

  The repository will simply be a file system storage. Access to the repository is provided via web - server (and the HTTP protocol). The repository must not be searchable from the first as the designated way to access the repository is via the links provided within the meta data stored in the Reference System. If thereinafter it seems to be useful to have a searchable repository for the digital surrogates, the repository can be changed to a database system. In this case a search/retrieve interface has to be provided for accessing the data.

5.6.6.2 Reference System

According to the Reference System Component Diagram, 4 components can be identified in the Reference System. The services provided by this subsystem will be described with WSDL/SOAP.

Special attention has to be turned to the fact that the Reference System has similar functionalities than the Repository Management System (for data management) and the Catalogue Data Management System and the Search & Retrieval System respectively (for search functionalities).

5.6.6.2.1 Meta Data Management

The tasks for this component consists of commonly known functionalities for databases. As meta data repositories have to be managed by the Reference System there is a need to create and delete repositories and to insert, update and delete data in these repositories. The Repository Management and the Catalogue Data Management include similar functionalities so it would be desirable to combine development activities for these functionalities.

The interface to the Reference System can be based on Z39.50 extended services. Evaluation of this protocol will show if it is applicable and useful in this context.

The Meta Data Management has to ensure data integrity in the repositories. For this purpose it uses a component which checks the data structure (see 5.6.6.2.3). The data transfer is XML based so the checking mechanism can be done with standard components already available (i.e. XALAN2, XERCES,...).

For accessing the databases an abstraction Layer which provides a consistent interface to the Meta Data Management that is independent of the actual database system used in the background.

5.6.6.2.2 Search Component

The interface to the Search Component is based on the Z39.50 protocol. The main task of this component is to split up the search requests and distribute it to the single repositories included in the Reference System. The results must be merged together to present a homogenous result to the client. This will include transformation to cross domain formats and transformations according to user selections (short, medium, complete result presentation). The transformations will be done by applying stylesheet transformations to the results retrieved from the repositories. The stylesheets are retrieved from the Ontology System which encapsulates the knowledge for data transformations. In further versions a query translator can be added to make the search services available to clients which are not familiar with the Z39.50 protocol.

For really accessing the databases the same abstraction layer as for the Meta Data Management component is used.

5.6.6.2.3 Checking data integrity

This component uses XML-Schemas (from the Ontology System) to verify the data to be inserted (updated) into repositories. As the data to be verified is also in XML format this is a standard task and tools for verifying are already available (Sun's Multi-Schema XML Validator (MSV), XML Schema Validator (XSV), …). This validation service must also be used if and external component wants to insert (update) data in repositories.

5.6.6.2.4 Database Access

The Database Access is an abstraction layer for accessing databases within the Reference System. It provides consistent interfaces to the Meta Data Management and Search Component.

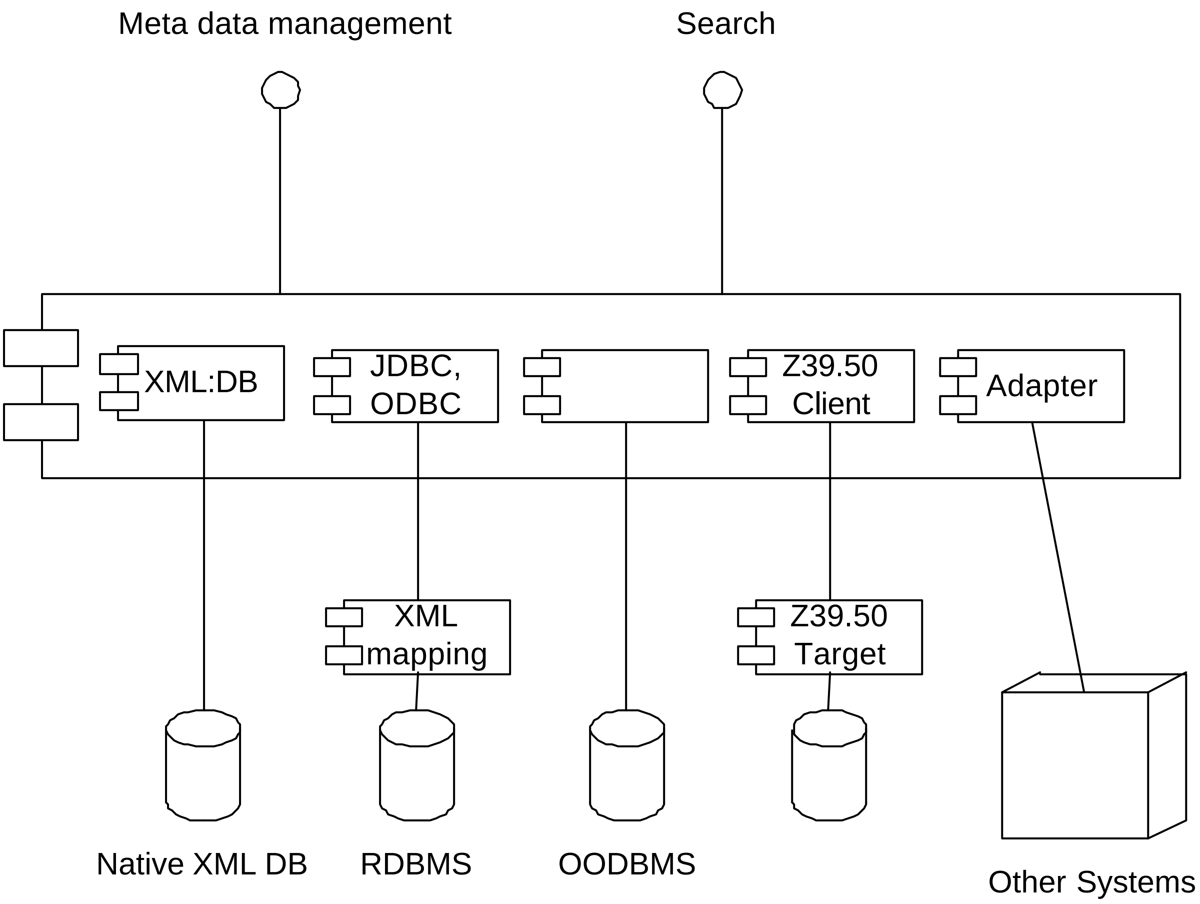


Figure 93: Component Diagram – Database Access

The task for this component is to transform all request for a specified target (Database System or other Information Management Systems). There are some groups of data management systems to identify:

- Native XML databases (e.g. dbXML, Tamino, Textml, …)

  This is the most natural choice of database because no mapping between storage format in the database and data format used in the system is necessary. To get reusable interfaces they should be based on existing interface standards as much as possible (XML:DB, XPath).

- RDBMS (Microsoft SQL Server (with XML interface extensions), MySql, …)

As the internal data structure used within the REGNET System is XML based, XML – mapping tools have to be used to transform data stored in relational database management systems. There are already tools available for some databases (e.g. Shiloh for MS SQL Server). In the case that there is no mapping tool available for a database it might be developed.

To get reusable interfaces they should be based on existing interface standards as much as possible (JDBC, ODBC, SQL).

- OODBMS

With the appropriate adapter it will be possible to use object orientated database management systems as well.

- Z39.50 Servers

The connection to Z39.50 Servers (which are widely used in the Cultural Heritage field) is an important point for the system. With such an interface the system opens up for many existing data sources.

- other Information Management Systems

As there are other systems which are capable of some kind of cross domain searching and information management, it is a good idea to be able to connect to such data sources as well. As one example a connection to the *"index+"* environment (used by the clustering partner OpenHeritage) will be established.

Most likely native XML database systems will be used. In order to support other database systems interfaces will be integrated in further versions of the project. These interfaces will be very useful for integration of existing repositories and they open the Reference System to use the database product of choice for each content provider.

5.6.7 e-Business Data-Management

The e-Business Data Management facilitates the management of data related products and services. This node is connected to the procurement and delivery of goods and services provided by cultural organizations and consists of two components:

- Product Catalogue Management (subsystem-7)
- Procurement, Delivery Invocation (subsystem-9)

5.6.7.1 Catalogue Management

As it is necessary for us to store large data volumes and to provide multi-user business logic of their processing, the system will globally consist from Data Source, Business Logic (Middle tier) components, Protocol layer components, which are described by the tree tier architecture

Any system of multi-user shared access to the data should provide security access ability.

Concerning to Data Source we should use the standard DC, which one on the present time is in development t means that the repository should allow work with a level meta information, which is present by the standard DC. The term meta will be used in relation to the external information saved in repository. This information should correspond to structure, described in meta information Model.

Main task of the meta information Model:

- The ability to save links, tags description, qualifiers of the standard DC, with help ones will conduct check of a correctness of structure and types of the loading information
- To enable fast attachment of the new data and new links
- To correct (modify) the old data and links
- To allow to realise a set of the main operations on the graph

The information, which is recorded in repository, nor is linear. It is the hierarchically introduced dataflow, which one should correspond to structure, described in meta information Model. The volume of these data is enough large, therefore it is meaningful to carve out data and their structure (links).The data will be presented in the Information Model, links between them in the Dependencies Model.

Main task of the Information Model:

The primary goals of Information Model limit by capabilities of storage, entry, updating and extraction of the data. Time information of last check of conformity of the data their source in internet also should be saved.

Primary goals of the Dependencies Model:

- An ability to save large volumes of the structured data (by the way of graph)
- Instead of the data, placed in the Information Model, to place tags of these data
- To provide fast search and extraction of the data
- To be open to upgrade
- To allow realize a set of the main operations on the graph

The access to Data Source can be realized from business level components, allowing multi-user access and executing primary goal of coordination of the inquiries and check of the rights users on inquired operations

All system operations and errors should be mirrored in the event log.

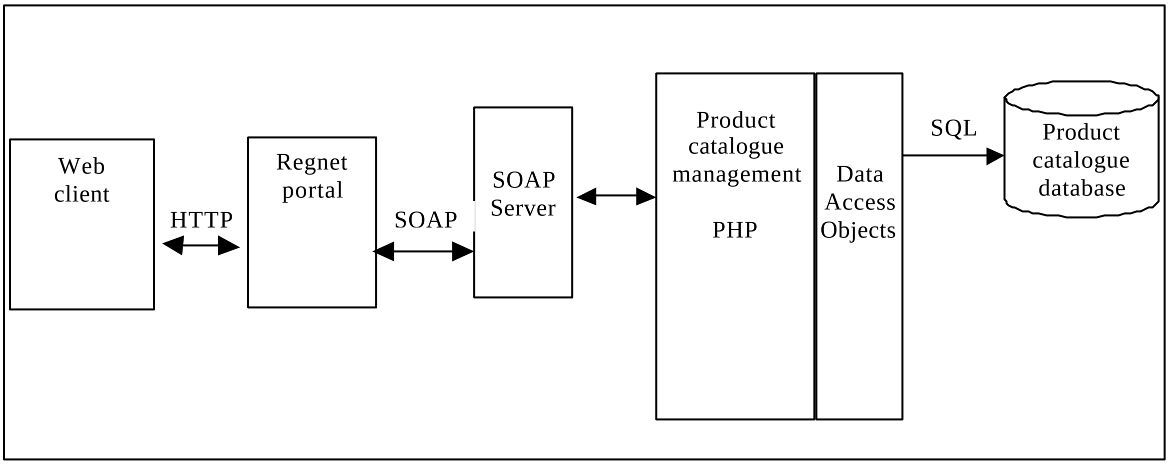


Figure 94: Technical architecture of product catalogue management

The above features will constitute the basis that will support the activities of the PCM.

5.6.7.2 Procurement and Delivery

Procurement subsystem is dedicated to the collaboration between users and suppliers. In the model described in the functional part, suppliers are in charge of the delivery process. REGNET Delivery is dedicated to the e-Shop.

5.6.7.2.1 Procurement

This part describe the technical architecture of the procurement subsystem. This architecture is illustrated by the following schema.

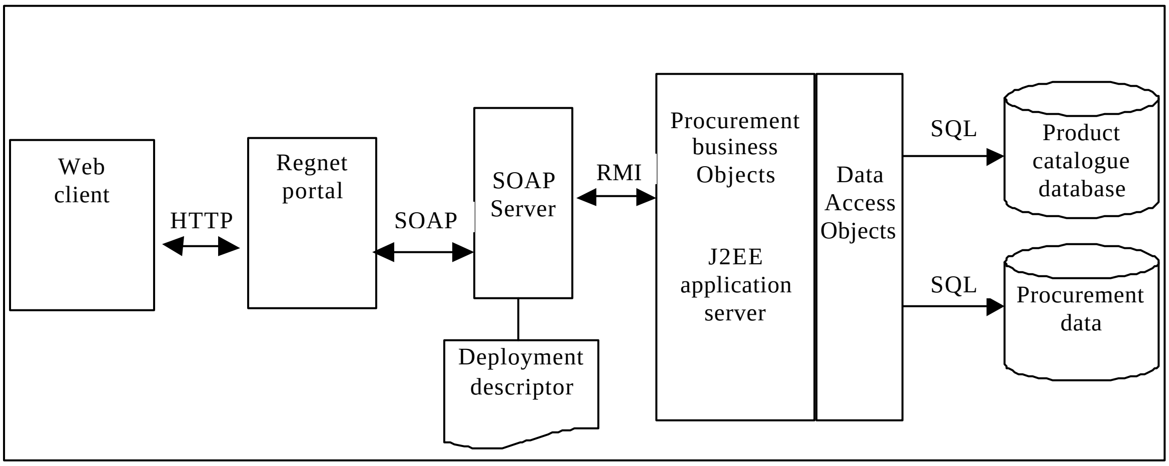


Figure 95: Technical architecture of procurement system

Business objects are managed by a J2EE container. A detail description of this technology is given by the technical annex. This container provide functionalities in order to manage Enterprise JAVA Bean components. These components implement Procurement Use Cases describes previously.

Soap server publishes business interface that different clients are able to call. This services are mapped in the deployment descriptor. Example of such services are:

- searchProducts
- getUSerCommand
- createCommand
- getProduct
- getAllProducts
- addProductToCatalog
- removeProductFromCatalog
- getSupplier
- getAllCommands
- getCommand
- getArtistCommand getAllSuppliers
- searchSuppliersToBeValidated
- subscribeNewSupplier
- getListOfType
- getListOfCategories

This services can be describe in WSDL and store in the UDDI registry.

5.6.7.2.2 *Delivery*

Delivery subsystem manage order coming from the e-Shop subsystem. It is based on a J2EE container.

It must be integrated with the stock management ERP (Entreprise Ressource Planning) of the e-Shop if existing. Integration technology depend on the available connector.

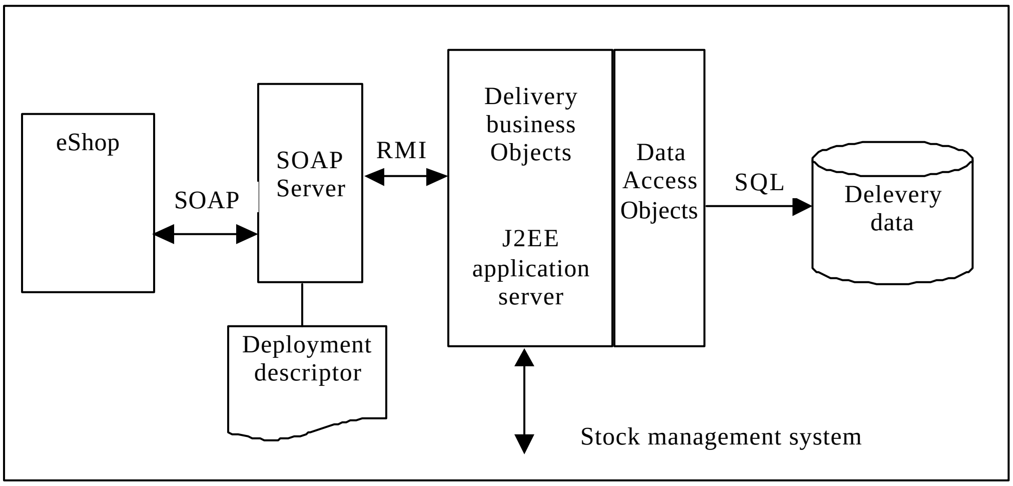


Figure 96: Technical architecture of delivery system

5.6.8 Certification

In order to handle the correct identification and copyright issues of the concerned objects treated within the REGNET environment, two techniques are envisaged: watermarking and unique number issuing.

Watermarking

Delivering texts, images, audio and video in electronic form to customers is rapidly increasing. The distribution of these multimedia elements over networks poses a great problem, namely the protection of the Intellectual Property Rights (IPR) of the concerned elements. In the case of REGNET, digital watermarking offers a solution to this problem for the images and the texts that will be produced.

Different watermarking techniques exist: visible-invisible, low-high number of bits, interoperability with other watermarking techniques, etc.

As far as images are concerned, REGNET, in the first phases, will primarily contain still images. Candidates for the watermarking of this kind of images are:

- Digimarc, a watermarking technology, currently used by one of the REGNET-partners
- Emerging new commercially available watermarking techniques (to be studied)
- Outcome of other European projects such as the currently ongoing IST-project TRADEX

The most viable way for REGNET will be to adapt to a de facto standard for watermarking that will be established in the near future.

Unique number issuing - Digital object identifiers

REGNET wants to introduce a unique number for the electronic versions of the texts, images and other multimedia elements that will be produced.

Whereas a specific REGNET numbering scheme can be envisaged, a digital object identifier (DOI) system would be preferable.

A DOI system:

- identifies and exchanges intellectual property in a digital environment
- provides the framework for managing content in any form and granularity
- links customers with content providers
- facilitates e-Business
- enables automated copyright management

In practice, the following sequence has to be followed:

2. Decide on the content
3. Obtain a DOI Prefix
4. Choose the numbering scheme
5. Define the Metadata and DOI preparation
6. Register DOIs with a Registration Agency.
   - Some agencies:
   - - CrossRef, USA
   - - Content Directions, USA
   - - Enpia Systems, Korea

5.6.9 Electronic Publishing

The main objective of the Electronic Publishing (subsystem 8) of the REGNET System is to provide an easy to handle tool for the creation of publications based on the search results gained by a query with the REGNET Search System and additional material coming from other sources.

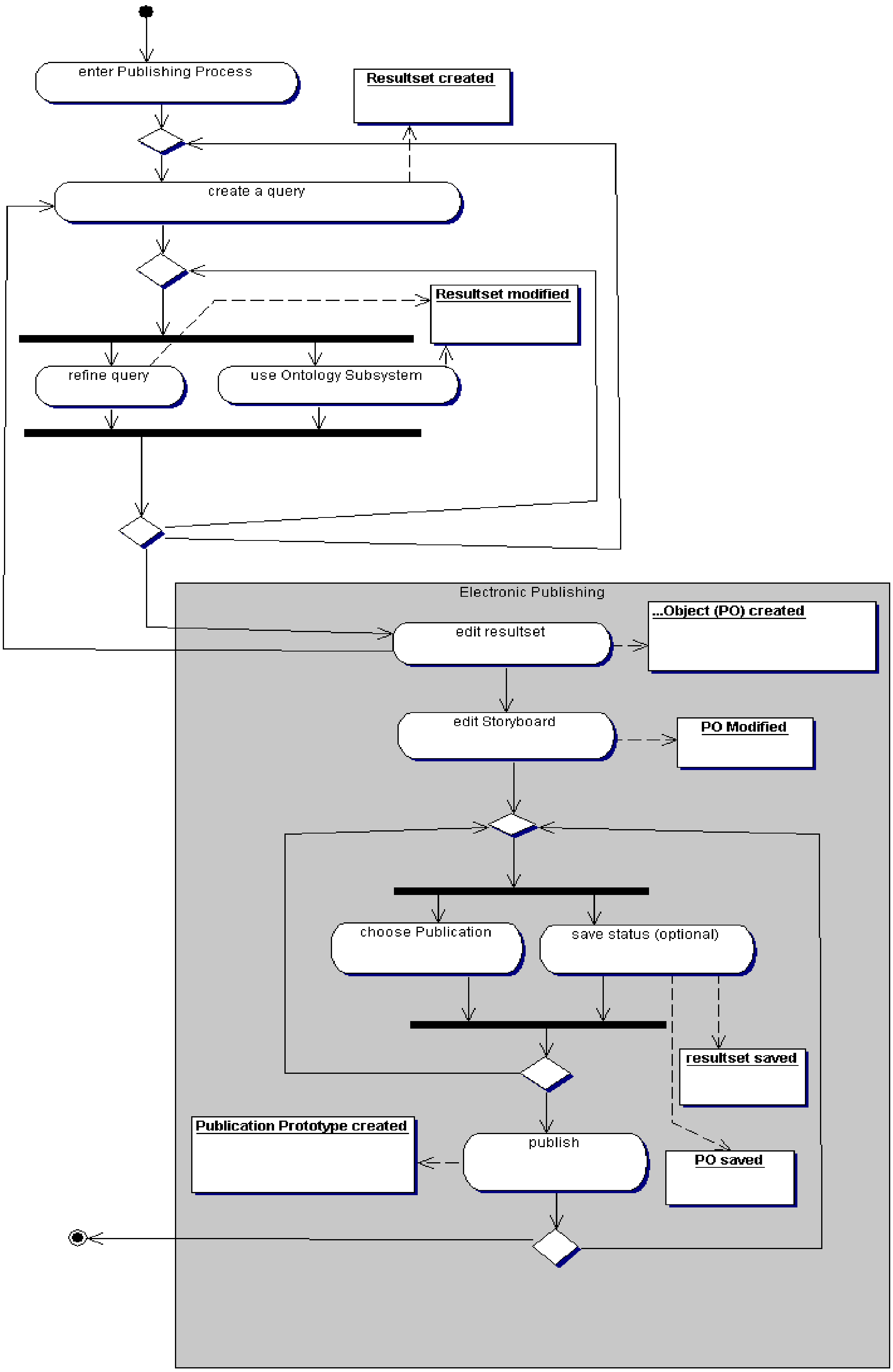


Figure 97: Activity Diagram – Electronic Publishing

The activity diagram shows how the publishing process might look like in more detail (See 7.6.6.4 for overview). Two Objects, the ResultSet and the publishing object are the main components used to manage the publication settings.

- ResultSet

The ResultSet will be the record set like described in External Interfaces: Search and Retrieval.

- Publishing Object

The publishing object is used to save all the settings the user made, when editing the ResultSets. (e.g. the records that will be published, the fields that will be embodied in the publication)

This object can also be used for further functions like saving the current status and get into the process again at any time.

Typically Web content generation is mostly based on HTML, but HTML doesn't separate the information from its presentation, mixing formatting tags, descriptive tags and programmable logic (both on server side and client side). Publishing frameworks offers a different way of working, allowing content, logic and style to be separated out into different XML files, and uses XSL transformation capabilities to merge them.

These aim to change the way web information is created, rendered and served. Their objective is a complete separation of the three layers, allowing the three layers to be independently designed, created and managed, reducing management overhead, increasing work reuse and reducing time to market.

To do this, the development of web content is divided into three separate levels:

XML creation - The XML file is created by the *content owners*. They do not require specific knowledge on how the XML content is further processed - they only need to know about the particular chosen "DTD" or tagset for their stage in the process. (As one would expect from a fully generic XML framework, DTDs are not required in Cocoon, but can be used and validated against). This layer is always performed by humans directly, through normal text editors or XML-aware tools/editors.

XML processing - The requested XML file is processed and the logic contained in its logicsheet(s) is applied. Unlike other dynamic content generators, the logic is separated from the content file.

XSL rendering - The created document is then rendered by applying an XSL stylesheet to it and formatting it to the specified resource type (HTML, PDF, XML, WML, XHTML, etc.).

The Electronic Publishing Subsystem is based on COCOON, the publishing framework from the Apache Cocoon project (http://xml.apache.org/cocoon/)

5.6.9.1 Publishing Framework

The server side of the application will be implemented using XSL-Stylesheets, XSP-Pages and Servlet-Technology. The client side of the Application will be implemented through dynamically generated HTML-Pages, certain functionalities will be implemented as JAVA-Applets. Applets will be embedded into HTML as objects using the JAVA-Plugin Virtual Machine. Thus problems with different implementations of the JAVA VM in Browsers can be avoided.

Communication with other REGNET- Components takes place via SOAP/HTTP, the communication with the web-clients via HTTP and dynamically generated HTML pages.

For details on the architecture of the Cocoon publishing framework and the description of the components see State-of-the-art.

5.6.9.2 Query REGNET System

The Electronic Publishing Subsystem submits a query to the REGNET System (in fact the search and retrieval subsystem) and gets back the results as a "ResultSet" in XML, consisting of Dublin Core and native Metadata. In both cases the answer is also a ResultSet. The ResultSet contains the matching data records each consisting of a description of a REGNET artefact (Dublin Core and Native Data) **and** a reference to the repository of this artefact. In the following, the term "artefact" is used for the digital surrogates of 'primary' (real world).

5.6.9.3 Access to Ontology Subsystem

If users want to extend the query they can call the Ontology Subsystem. The Ontology System can be accessed through the Search and Retrieval Subsystem, which refines or extends search results. If themes are searched, the result will consist of the XTM Topic Map representation of the theme and the data records.

5.6.9.4 ResultSet editing

For the publishing process a possibility for editing the ResultSet must be given. The ResultSet has to be scrolled through, and it must be possible to exclude records from the publication. It has to be distinguished between editing the Metadata and editing of artefacts. For the editing of the artefacts (if necessary) third party programs (for example MSPhotoeditor or Corel Photopaint) can be started. Thumbnails should already be generated at the time of data acquisition and come by reference with the Metadata so that the user can more easily decide between the utilisable records and thrash by viewing the thumbnails only in an early step of the publishing process. The editing is thought to be done either via HTML -forms or JAVA- applets. The modifications are stored as references to the ResultSet. Already for the first displaying of the raw data a chain of XSL transformations from the XML ResultSet into HTML is required.

5.6.9.5 Storyboard view

The Storyboard view allows easy editing of the data records. Therefore, if thumbnails are automatically placed on a board in one-dimensional order or on a map in two- dimensional order. The map is planned to be a SVG-Map, via drag and drop, cut, copy and paste the data can be edited. For Storyboard View an Applet, probably using BATIK for rendering SVG will be launched on the client side.

5.6.9.6 Creation of publications

The publication is created via XSL- Transformations or/and XSL- Formatting. The stylesheets for each type of publication are prototype stylesheets which can be accommodated by the user. XALAN is intended to be used for stylesheet transformation, FOP for formatting XSL-FO to PDF. As underlying framework Cocoon comes into operation.

The Formatting of XSL:FO to PDF probably has to take place on the client platform, because this process consumes very much performance.

5.6.9.7 External interfaces of Electronic Publishing

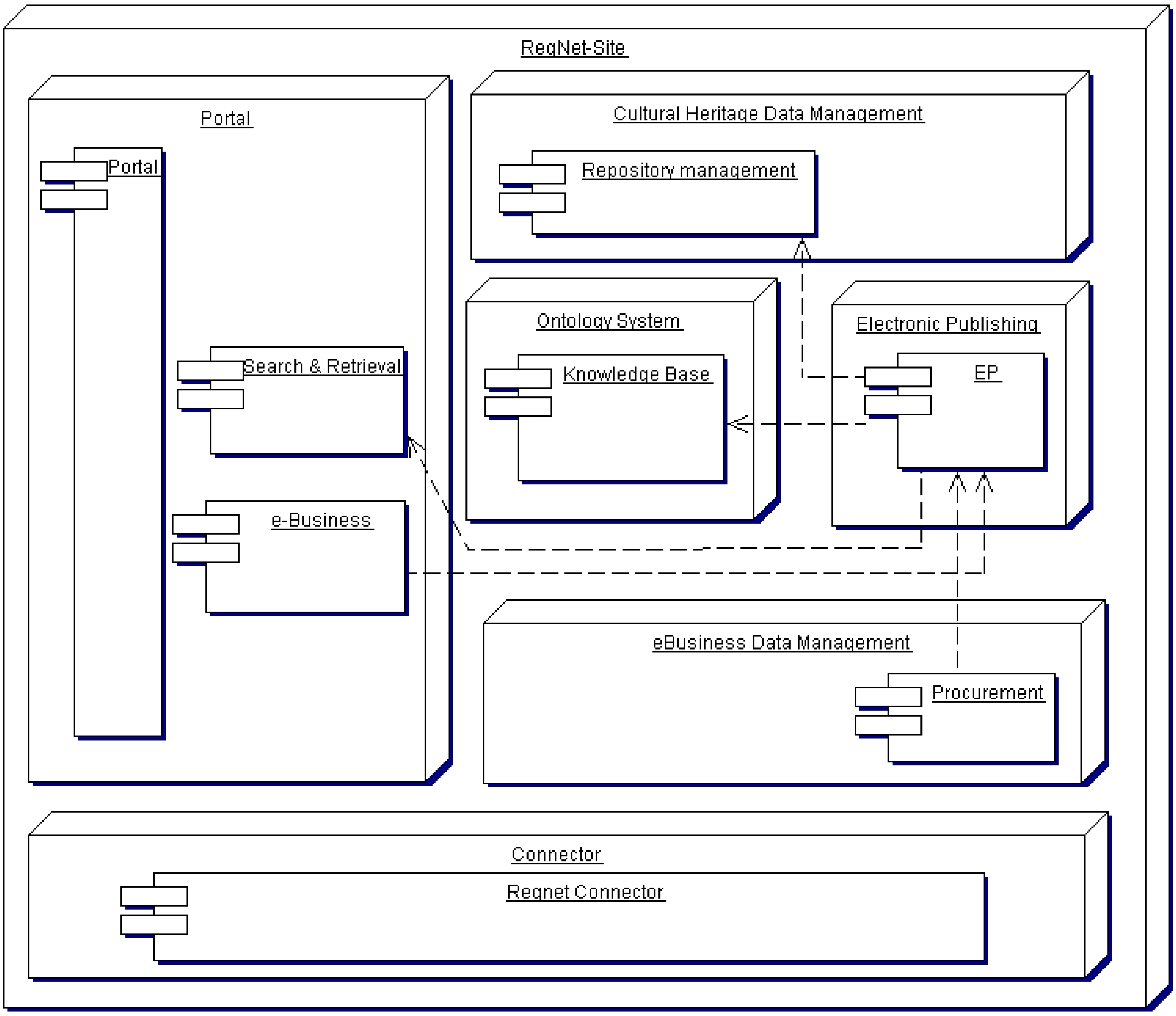


Figure 98: Deployment Diagram – Electronic Publishing

Besides their will be provided an interface to the Session, User profile and Terminal profile manager.

5.6.9.7.1 *Graphical user interface*

The graphical User Interface of the publication Prototype is a dynamically generated HTML- Page which is displayed in the GUI for the e-Business Subsystem. The minimum space required for the GUI of the publication subsystem is 750x550 pixel. This GUI is designed for lightweight web-browsing clients (IE, Netscape, Opera, defined in 8.2.1) which have installed the JAVA VM plugin. On other Clients (PDA, Handy- Cellphone) the functionality of the publication subsystem will be restricted.

5.6.9.7.2 *Search and Retrieval*

This is the main interface for communication with the REGNET System (in fact with the Search and Retrieval Subsystem). It has to provide a method for sending a query to the REGNET Portal which gives back the results as an XML document. If the results cannot be served as one document, methods for accessing blocks of data should also be provided (e.g.: "next ten esults"). The data records within the XML document are expected to having been standardised by another REGNET Subsystem already (mapping from native to Dublin Core format). Probably the ResultSet is enveloped in a XER format and contains additional information about User ID, ResultSet ID, search Request etc. In every case the ResultSet has to contain:

- **Metadata** in Dublin Core Format - are needed to provide the user of the Publishing System a preview of the result set.

- **Native Data** (EAD, AMICO..) - are needed to have a more precise data and have all the data needed for the publication. (The Metadata can be followed by the Native Data in the record or there can be a reference)

The result records can than be processed individually. The Native data must additionally have a reference to the electronic representation of the artefact. Metadata about the type of the object of the reference has to be provided, too. (JPEG, GIF, MPEG....)

Following Table shows a (shortened) ResultSet from the COVAX project which could be processed by the publishing prototype (DC not included).

Sample ResultSet:

```
<?xml version="1.0" encoding="UTF-8"?>
<Search>
       <presentResponse>
              <referenceId>1234567890</referenceId>
              <numberOfRecordsReturned>13</numberOfRecordsReturned>
              <nextResultSetPosition>14</nextResultSetPosition>
              <presentStatus>success</presentStatus>
              <records>
                     <responseRecords>
                            <Item>
                                   <name>BTH-BIB-RESEARCH</name>
                                   <resultSetRecId>1</resultSetRecId>
                                   <presentRecId>1</presentRecId>
                                   <record>
                                          <retrievalRecord>
                                                 <External>
                                                        <direct-reference/>
                                                        <encoding>
                                                               <CovaxNative>
                                                                      <dc
docType="mrcb"/>
                                                                      <mrcb format-
type="bd">
                                                                             <mrcb-main-
entry>

       <mrcb100 i1="i1-1" i2="i2-0">

       <mrcb100-a>Larsson, Sven-Olof</mrcb100-a>

       </mrcb100>
                                                                             </mrcb-main-
entry>
                                                                      </mrcb>
                                                               </CovaxNative>
                                                        </encoding>
                                                 </External>
                                          </retrievalRecord>
                                   </record>
                            </Item>
                            <Item>
                                   <name>SR-ARC-SAMPLES</name>
                                   <resultSetRecId>2</resultSetRecId>
                                   <presentRecId>2</presentRecId>
                                   <record>
                                          <retrievalRecord>
                                                 <External>
                                                        <direct-reference/>
                                                        <encoding>
                                                               <CovaxNative>
                                                                      <dc docType="ead"/>
                                                                      <ead
relatedencoding="MARC21">
                                                                             <eadheader
langencoding="ISO 639-2" findaidstatus="edited-full-draft" audience="internal" id="a0">

       <eadid systemid="MnCCHi" source="DLC" type="file"
```

```
encodinganalog="850">2468</eadid>

        <filedesc>

        <titlestmt>

                <titleproper>William Fonds Provenance</titleproper>

                <subtitle>An Inventory of His Papers at the Cupcake Corners Historical
Society</subtitle>

                <author>Inventory prepared by B.W. Moos</author>

        </titlestmt>

        </filedesc>
                                                                                        </eadheader>
                                                                                </ead>
                                                                        </CovaxNative>
                                                                </encoding>
                                                        </External>
                                                </retrievalRecord>
                                        </record>
                                </Item>
                                <Item>
                                        <name>SRF-MUS-FRONFESTE</name>
                                        <resultSetRecId>5</resultSetRecId>
                                        <presentRecId>5</presentRecId>
                                        <record>
                                                <retrievalRecord>
                                                        <External>
                                                                <direct-reference/>
                                                                <encoding>
                                                                        <CovaxNative>
                                                                                <dc
docType="am_object"/>
                                                                                <am_object>
                                                                                        <am_titles>

        <OTG_object_title_name>

        <OTN__object_title_name>Scherdegen</OTN__object_title_name>

        </OTG_object_title_name>
                                                                                        </am_titles>

        <am_media_metadata>

        <XID__dc_resource_identifier>
http://www.covax.at/content/mus/fronfeste/images/tn_hmnmktw_leder_i000004_jpg.jpg

        </XID__dc_resource_identifier>

        </am_media_metadata>
                                                                                </am_object>
                                                                        </CovaxNative>
                                                                </encoding>
                                                        </External>
                                                </retrievalRecord>
                                        </record>
                                </Item>
                                <Item>
                                        <name>BTH-TEX-SAMPLE</name>
                                        <resultSetRecId>6</resultSetRecId>
                                        <presentRecId>6</presentRecId>
                                        <record>
                                                <retrievalRecord>
                                                        <External>
                                                                <direct-reference/>
                                                                <encoding>
```

```
                                                             <CovaxNative>
                                                                    <dc
docType="TEI.2"/>
                                                                    <TEI.2
id="sheldon">
                                                                           <teiHeader>

       <fileDesc>

       <titleStmt>

              <title>Charles Sheldon and the BalticÂ´s first dry dock</title>

              <author>Harris, Daniel G.</author>

       </titleStmt>

       </fileDesc>
                                                                           </teiHeader>
                                                                    </TEI.2>
                                                             </CovaxNative>
                                                      </encoding>
                                                </External>
                                         </retrievalRecord>
                                  </record>
                           </Item>
                    </responseRecords>
              </records>
       </presentResponse>
</Search>
```

ResultSet.xml

If the user has searched for themes, the resultset also contains an XTM Topic Map representation of the theme.

5.6.9.7.3 REGNET Ontology System (Knowledge Base Access - Subsystem 3)

If the results of the query are not satisfying, the user can send a request to the REGNET Ontology Subsystem. (Related to the User Profile or the Ontology knowledge base). The Electronic Publishing Subsystem retrieves a new ResultSet from the Search and Retrieval Subsystem (same format as retrieved by a REGNET Query)

If there are several suggestions the process can be performed in an iterative way as long as the user thinks that the suggestions should be added.

All stylesheets and storyboards can be stored in and retrieved from the Knowledge Base of the Ontology System either in XML format or as binary file.

5.6.9.7.4 Repository Management(Subsystem 1)

Calls to the Repository Management Subsystem are necessary for retrieving thumbnails or the artefacts for example which are referenced in the ResultSet by an URL.

In addition the Repository Management Subsystem will store the external data sources for the publication process and make it accessible for the Electronic Publishing Subsystem. For this purpose a dedicated repository is needed.

5.6.9.7.5 e-Business (Subsystem 6)

Interface to provide access to the Electronic Publishing Subsystem for the e-Business Subsystem. Methods for retrievi ng information over the publication (number of records, publication type, etc.) will be provided. User registering and login session will be done by the e-Business Subsystem.

The handling of copyrights of individual objects and images will not be dealt with in the Electronic Publishing System.

There will be an XML File available with relevant information for further accounting containing:

- Number of publications
- Type of publications (Printable version, CD-ROM,..)
- Number of included objects
- Type of objects

5.6.9.7.6 *Procurement, Delivery Invocation (Subsystem 9)*

Interface to provide access for the Procurement and Delivery Subsystem. This will probably be methods similar to methods in interface "e-Business".

5.6.9.7.7 *Session Manager, User Profile Manager, Terminal Profile Manager*

The Session Manager, User profiles manager and Terminal profiles manager has impact on data filtering, translation, access rights etc. Additionally, the managers will demand the publication subsystem via this interface.

5.6.9.8 Additional Requirements from other Subsystems

5.6.9.8.1 *Search and Retrieval*

The Search and Retrieval System must provide functions to support the search for themes. A selected theme must then be exportable. As themes can be nested the structure and all the objects can be exported to the Electronic Publishing System. This structure can be mapped in a XML file.

5.6.9.8.2 *Data Generation*

If new objects are generated the Data Generation System must also make time/date and geographical data entry available. For geographical data entry we suggest that the longitude and latitude and the name of the town must be available.

In order to provide the user with preview thumbnails of each object the Data Generation System has to generate an upload of thumbnails to the Repository Management System and the Search and Retrieval System has to include a reference in each data record.

5.6.9.8.3 *e-Business*

In order to manage the publication session the eBusiness System has to provide the UID to the Electronic Publishing System. Information about number and type of publication will be made available to the e-Business System. Since there was a common decision to use SOAP as Communication Protocol between the REGNET Subsystems and the different subsystems can be implemented independently we will save all the relevant information in an XML file that can be accessed for further processing.

5.6.9.9 Themes

The publishing function within REGNET can make use of all types of data available in the system. Practically all the material delivered by REGNET can be used as a stand-alone information source that makes sense on its own. However, in most of the publishing cases the themes will be used as a guidance or (part) of a framework for a full storyboard or full scenario while the fragments and object descriptions and images can be considered as the raw input material for the publication. Additional texts and input from other sources will result in a real new coherent and consistent publication.

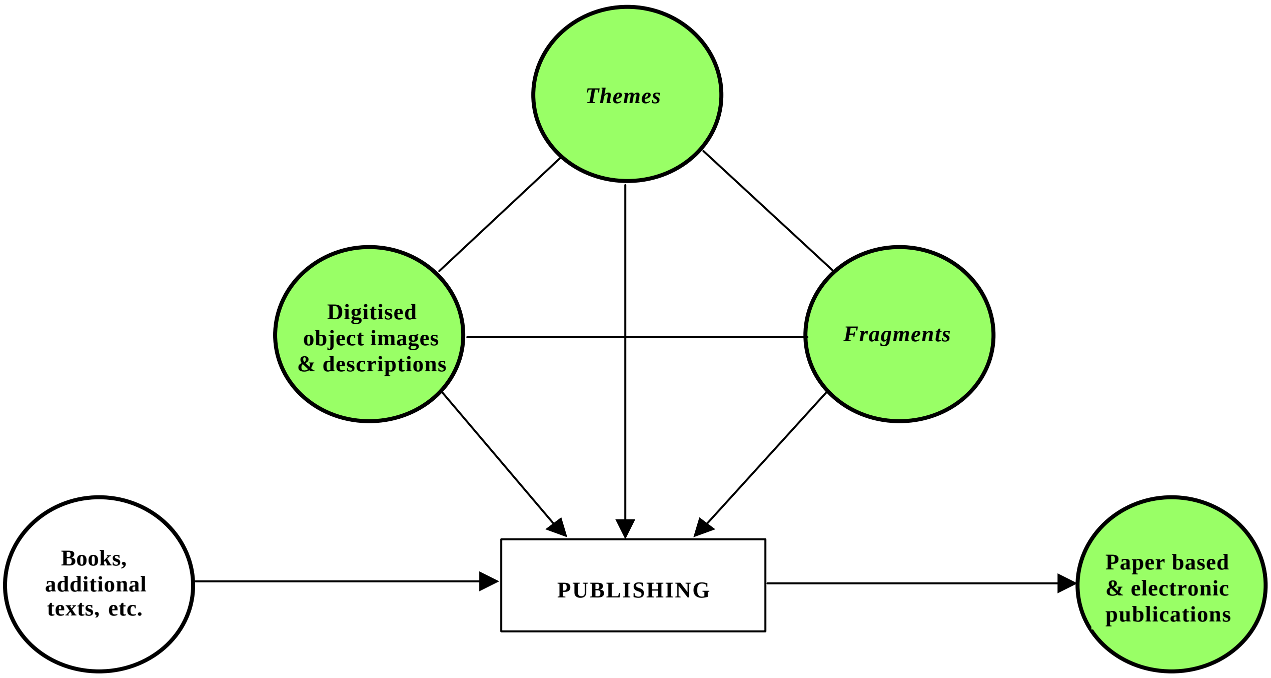

Themes
Digitised object images & descriptions
Fragments
Books, additional texts, etc.
PUBLISHING
Paper based & electronic publications

Figure 99: Cultural Heritage data publishing

5.6.10 Ontology system

The Ontology System is the backbone to the REGNET system. It consists of a *knowledge base* and a set of rules and tools that perform management on the knowledge base data. The System is responsible for the integrity and the even flowing of the REGNET portal.

Technically, it can be perceived as the entire system processor (along with the Reference System), where all the system metadata are stored, processed and sent out to the other systems. It is self-explanatory that the end-user doesn't have any access to it. Rather, the Ontology system is interconnected with all the other REGNET nodes, streaming in useful information, for example the suitable Style-sheet for the Electronic Publishing system.

Better structure

Knowledge Base

The Knowledge Base part of the Ontology System stores data pertinent to the integrity and validity of the whole system. Moreover, it contains the metadata for the REGNET portal.

a) **Topic maps** (for every node): they are used to store the metadata of the information found on every REGNET node. This means that the Ontology System will contain the metadata of the whole system, to be used for the museum data, the publishing and business system, and so on. The Topic Maps format that will be applied is XTM (XML Topic Maps).

b) **DTD for SHOE:** SHOE is in meaning and function very close to the concept of Topic Maps but it was decided not to be employed due to certain disadvantages it possesses. Nevertheless, its DTD will be used in conjunction with the Topic Maps DTD for the purpose of storing the metadata.

c) **Style-sheets:** they are used especially for publishing procedures. The format of choice here is XSL, the Extensible Style-sheet language.

d) **User and Terminal Profiles (for every node):** the profiles of each user logging into the system and each node accessing the REGNET subsystems will be coded in standard XML.

e) Organisation Profiles

f) **Binary Data (java classes):** They will be used especially for the "Electronic Publishing System", and they will be invoked every time some function needs to be performed for Electronic Publishing either for the creation of items (booklets, brochures, etc) to be posted, or for online presentations.

g) **DTD and/or XML-Schemas:** These are especially suitable for data integrity checking, that is if the data is in the correct format to use or not. The Reference System will perform this checking.

h) **Other data (thesauri):** such as thesauri, which could help the end user refine his/her search by typing words close in meaning to the original ones.

i) **Query definitions:** standard queries are defined and stored in the system. These are used to extract metadata from the Knowledge base and are formulated in a specific format for querying XML data, X-PATH.

j) **Location of repositories in the REGNET Network:** it stores information about the physical and network location of the repository nodes storing the museum/library data available in the REGNET Network. Moreover, if for some reason a node becomes unavailable this is noted in the Ontology System so as the other REGNET Systems to avoid transactions with it.

k) **Administrative information:** General information about the cultural and technical organisations collaborating with REGNET, the users, the privileges of each one, and so on, are also stored on the Ontology System.

Ontology System management

This subsystem consists of a set of rules and tools managing the data found in the knowledge base, and "transmitting" this data to the rest of the REGNET systems connected to the Ontology system.

a) **Database Management:** as a database Management tool it can perform all the usual database functions, like find, delete or modify topic maps and other metadata from the Knowledge Base. Furthermore it will send away all requests for such metadata.

b) **Interconnection:** the Ontology System will use a common and widespread distributed system protocol for connecting to all the REGNET systems. The protocol of choice will be RMI, and later on SOAP. SOAP is the more generic and simple one, but RMI makes more sense as a starting point and greatly facilitates the process, since it was specially designed to use with the JAVA programming language (in which most of the project code will be written).

5.6.11 REGNET connector

REGNET connector allows a REGNET system to collaborate with others REGNET system. This connector is mainly based on ebXML technologies but it also provides Z39.50 gateway in order to allow distributed search

Connector provide standard WSDL interface which is stored into the registry. This interface propose services which can be queried trough SOAP protocol. Security context must be propagated by SOAP messages.

Messages and data are ebXML compliant which imply they fit definition (DTD or Schema) stored into the registry and necessary core components.

5.6.12 Internationalisation and localisation

Internationalisation is the process of designing an application so that it can be adapted to various languages and regions without engineering changes. Internationalisation can be address in different way depending technology used.

The technical realisation of internationalisation and localisation within REGNET is based on several techniques. First of all, the operating systems, databases, web servers, mark up and programming languages that will be used within REGNET have to support UNICODE as character encoding system now or are undergoing actively developing efforts for it. Independent from the platform, the language and the application program, UNICODE offers a unique number for every character.

The second key element for realising internationalisation is the JAVA Internationalisation API. This provides following characteristics:

- through the addition of localised data, the same executable can run world wide;
- descriptive elements are not hard-coded in the code base but are located in separate areas and retrieved dynamically;
- addition of new languages does not require recompilation of the code and they can be implemented in a very quick way;
- culturally-dependent data presentations appear in formats that conform to the end user's region, language and profile;
- switching between languages, regions and profiles can be carried out in real time while the program is running.

A lot of JAVA-information and -tools are available at the web sites of respectively Sun Microsystems and IBM Corporation.

As far as vocabulary references are concerned, the latest multilingual developments will be incorporated in REGNET. Besides the own generated thesauri by REGNET, other standards such as the Art & Architecture Thesaurus (AAT), the Union List of Artist Names (ULAN), the Getty Thesaurus of Geographic Names (TGN) and others will be used in as much language versions as possible, depending on their implementation status and availability. However, on those topics there is still a lot of work to be carried out in order to attain a sufficient level of smooth integration of different European cultures and languages.

Concerning multicultural issues, especially languages, REGNET is delivering a substantial contribution to the practical implementation of solutions to this characteristic European property.

Whereas most systems are offering multilingualism on the guidance and presentation level, the content parts themselves lack the same feature. As far as collection management systems are concerned, this is mostly due to the fact that those systems don't offer an easy way to implement multilingualism.

Through the introduction of the newly to be made "thematic texts" in REGNET, the multilingual feature is build in at the data entry level (you have to specify the language of the content). Due to this, one of the important language barriers has fallen away.

Another important opportunity for a real multilingual implementation is the availability, via the participating institutions and organisations, of a large number of languages:

- Bulgarian
- Russian
- Swedish
- Dutch / Flemish
- German / Austrian / Swiss German
- Spanish
- Italian
- English / American English

Typical and subtle regional aspects (e.g. different names for the same item) can be worked out in all concerned domains. Even within families of languages, governed by the same syntax, those typical regional aspects play a major role. This is the case for instance with German, Austrian and Swiss German, also for Dutch and Flemish just as for English and American English.

5.7 System architecture and tools

This part detail necessary software tools for each module. It contain a list of tools which implement technologies described in the previous part of this document.

5.7.1 Web acquisition client

The Web acquisition client used a JAVA Applet or available plugin if necessary.

JAVA^{TM} Web Start is a new application-deployment technology gives the power to launch full-featured applications with a single click from the Web browser. User can now download and launch applications, such as a complete spreadsheet program or an Internet chat client, without going through complicated installation procedures.

5.7.2 Light client enabled to Web services

5.7.2.1 Web browsing client

Web browsers can be:

- Netscape Communicator v.4.78 or higher
- Microsoft Internet Explorer v.5.5 or higher
- Opera Cowified v.5.12 or higher

5.7.2.2 Wireless client

There can be two types of this client:

1. WML: the client consists of a series of WML deck. The *UP.SDK* is a tool to support the development of WAP 1.1 services written in WML and WMLScript. The UP.Simulator will execute content written in WML, WMLScript, and HDML.
2. J2ME: the client is a MIDlet. Jext is used as JAVA editor and J2ME Wireless Toolkit as terminal emulator.

5.7.3 REGNET Portal

All the modules in presentation and applicative logic are developed using the JAVA language.

Server side JAVA modules will process the client requests.

The interactions with other modules are based on:

- SOAP in heterogeneous context (e.g. JAVA – PHP or PHP – JAVA)
- RMI in homogeneous context (JAVA only).

Jext is a 100% pure JAVA programmer's text editor. It is lightweight software, which provides an open architecture and many different possibilities to manipulate text of any kind.

Ant is a JAVA based build tool. Instead of a model where it is extended with shell-based commands, it is extended using JAVA classes. The configuration files are XML based calling out a target tree where various tasks are executed. Each task is run by an object, which implements a particular Task interface.

Tomcat is a Servlet and JSP engine that can be integrated to Apache Web server.

5.7.3.1 Data generation

The Data Generation component has in principal 2 different aspects: direct data generation and harvesting. For what concern the connections to the other component, it will be based on the SOAP protocol. The data entry forms will be generated by the means of XSLT processing (Exstensible Stylesheet Language Transformation). The Stylesheets for all the data formats used within the REGNET System are stored in the Ontology System.

Data generation:

The Data Generation component will be developed using PHP/ASP and JAVA respectively. Which programming/scripting language is used depends on the type of client to be developed. For heavy clients a JAVA application running with JAVA Web Start may be used. For other client interfaces existing components may be used or developed (e.g. JAVA Applets, PHP/ASP applications).

Harvesting:

Concerning tools implementing the Harvest module we could adopt the Combine Harvesting robot (http://www.lub.lu.se/combine/): Combine is an open system for harvesting and indexing Internet resources. The Combine was initially developed as a part of the Development of a European Service for Information on Research and Education (DESIRE) project, which was funded by the European Commission within Telematics for Science Program.The Combine Robot was released only in 1998, and is currently used within the Nordic NWI services in Sweden and Denmark. It is also used in other contexts like the research information system Safari. Combine its useful for regional web indexing in general.

Another Harvesting tool which can be adopted to REGNET is the DC.dot Dublin Core Generator. This service retrieves a web page and automatically generate Dublin Core tags from the processed web page section. The generated tags can be edited and transformed into various meta data formats (USMARC, TEI Header, RDF, etc.) using the provided editing and transformation forms.

5.7.3.2 Search & Retrieval

The Search & Retrieval System will be implemented in JAVA.

The connectons to other modules are based on:

- SOAP in heterogeneous context (e.g. JAVA – PHP or PHP – JAVA)
- RMI in homogeneous context (JAVA only).

Datastructures for interfaces to the subsystems will be XML-based and defined with XML-Schema (or DTD's).

For internal data conversions available API's will be used as much as possible. The JAXP packages for example will be used as API for stylesheet transformation and for verification of XML documents against XML-Schemas and DTD's.

5.7.3.3 E-business

The E-Business subsystem is one of the main elements of the REGNET system. The infrastructure that will be used is the ebXML infrastructure .

As sawn before, tools that will support the E-Business system are PHP or JAVA. Connections to other subsystems will be based on the SOAP protocol. The interfaces to the subsystems will be XML-based and defined with XML-Schema (or DTD's).

For the e-Shop component, PHP language will be used in order to implement presentation and business layers. MySQL database will be the underling software tool in order to provide relational database.

5.7.4 Cultural Heritage Data Management

5.7.4.1 Repository Management

Access to the repository is provided over the web by the means of a web-server. The other 2 modules are in the Repository Management ("Data Parser" and "Check Data Integrity") are light components. For some of the tasks there are some extensions to web-servers already available. For own development in this area frameworks for easy extension of web-server functionality will be used.

The choice is to use Apache as web-server and Tomcat as extension framework for JAVA based development. If there are suitable PHP modules for some task already available the should also be used.

5.7.4.2 Reference System

As the Reference System has functionality similar to the Repository Management, the Catalogue Data Management and the Search & Retrieval System it is highly desirable to develop modules which are applicable in more than one of these subsystems. So the tools used will depend on these modular development and the tools used in the other subsystems. Likely the implementation will be in JAVA or PHP. Access to meta data repositories should use existing interfaces (shipped with the database products) as much as possible.

Databases that will be used in the first phase will be native XML databases. They are easier to integrate as there is no mapping between relational data tables and XML structures necessary. Possible products are for example: TEXTML-Server, dbXML (open source), Tamino, …

Goal is to be able to integrate various database systems. So in further phases also RDBMS (mySQL, MS SQL Server, …) and OODBMS can be used.

To get generic interfaces to the database systems, standards must be used to access the databases. These standards are:

- XML:DB for Native XML databases
- ODBC, JDBC for RDBMS
- ODBC for OODBMS
- Z39.50 Client protocol for Z39.50 Targets.

To connect to other (external) systems special adapters have to be implemented.

5.7.5 e-Business Data Management

5.7.5.1 Product Catalogue Management

The most important point that will constitute the basis for the implementation and the development of the PCM is the specific functionality and requirement of the PCM within the REGNET system. An small example of the PCM use case is the following:

The first step of the query is directing the SOAP request to a Query Translator (QT). SOAP request contain the all necessary parameters to immediate creation of the SQL query to the cashed data warehouse, and to enhance the process of the search QT also sends parameters received from the SOAP query to the module which has to form the distributed query. Then depending of the result of the search throw the cashed data, the following decision should be made:

If there is no records (or less than some determined number of records) in the result set, the program makes a signal to send the distributed query to remote databases. Then, receiving the response from the distributed query, a module named “Form SOAP response” which might be a part of the Decision Support should produce the SOAP response and send it back to the portal’s level.

The Modules of the PCM will be developed using **PHP** or **JAVA** language.

CatXML (www.catxml.org) will be test in order to determine its availability and usability for this component.CatXML is an open interoperable standard for catalogue information exchanges.

The goals of the initiative are to allow interoperable exchanges of catalog details using XML based syntax and developing an open interface API. The work of the participants is strongly aligned to the work of the ebXML initiative. They are committed to providing implementations of CatXML systems that are also ebXML compliant.

An initial online concept demonstrator is available utilizing the GoXML engine and conforming to version 1.1 of the CatXML specifications.

A backup solution will be to build this system on Z39.50 specification. This will imply development of an application profile dedicate to REGNET Product Catalogue Management system.

The tools that will support the distributed query will be the Z39.50 protocol, which will be supported by the **PHP/Yaz functions**. The basic features of the PCM, which are the repositories are going to be XML based repositories

Test of CatXML, backup solution: Z39.50 (development of an application profile) and data management system.

Database technology: an XML data base will be used, utilisation of dbXML will be address.

The dbXML Core is a native XML database. As such it is a database server intended to manage large numbers of small XML documents. The documents are stored in collections and the server provides the ability to query these collections using XPath. The server is lightweight, modular and suitable for embedding in custom applications or running as a standalone database.

The dbXML Core is written in JAVA and will run on any platform that has a Stable JAVA JDK 1.3 available. For JAVA developers it provides an implementation of the XML:DB XML Database API that makes developing XML database applications easy. For developers working in other languages the server exposes a CORBA API to make access available to any language that supports a CORBA binding.

5.7.5.2 Procurement and delivery

The J2EE application server that we propose to use is JBOSS (http://www.jboss.org). JBoss is an Open Source, standards-compliant, Enterprise JavaBeans Application Server implemented in 100% Pure JAVA.The JBoss organisation is working to deliver JBoss as *the* premier Enterprise JAVA application server for the JAVA 2 Enterprise Edition platform. JBoss will be delivered under the LGPL licence.

The SOAP server is Apache SOAP. Apache SOAP ("Simple Object Access Protocol") is an Open Source implementation of the SOAP submission to W3C. It is based on, and supersedes, the IBM SOAP4J implementation.

Apache SOAP general features are:

- Supports most of the SOAP v1.1 specification.
- Supports the SOAP Messages with Attachments W3C Note.
- Provides server-side infrastructure for deploying, managing and running SOAP enabled services.
- Provides client-side API for invoking SOAP services.
- Release includes full source under the Apache Software License.
- Supports three encoding styles: SOAP v1.1 Encoding, Literal XML and XMI.
- XMI encoding (available when using JAVA 1.2.2) supports automatic marshalling and unmarshalling of arbitrary objects.
- SOAP encoding: built-in support is provided for encoding/decoding primitive types, Strings, arbitrary JavaBeans (using reflection) and 1-dimensional arrays of these types. For other types user can hand-write encoder/decoder and register with XML-SOAP runtime.
- Literal XML encoding: allows one to send XML elements (DOM org.w3c.dom.Element objects) as parameters by embedding the literal XML serialisation of the DOM tree. No code needs to be written to support this (see the address book demo to see a sample use of it).
- Supports messaging and RPC over two transports: HTTP and SMTP.
- Supports authoring services in scripting languages.

5.7.6 Electronic publishing

The Electronic Publishing Module is a server-side application based on the Cocoon Framework (http://xml.apache.org/cocoon/index.html)

COCOON is an Open Source Publishing Framework and part of the Apache XML project. Cocoons most common use is the automatic Creation of HTML through processing XML files but Cocoon also allows XSL:FO rendering to PDF files or client dependent formatting such as WML for WAP devices. It integrates the Stylesheet transformator **XALAN**, an XSP Processor and **FOP,** a Processor for creating PDF.

For running Cocoon, a **JAVA Virtual Machine** (1.2.x or greater) and **Servlet Engine** (2.x) has to be installed on the system. This could be achieved through installing **Apache JServ** or **Apache Tomcat** for instance.

For displaying SVG the use of **BATIK** is planned. BATIK is the Open Source SVG- Project at http://xml.apache.org/batik/. Part of the project are SVG parsers, SVG DOM implementations SVG generator. The modules of BATIK can easily be integrated in other applications.

5.7.7 Ontology system

5.7.7.1 Tools

For a long while it was difficult to experiment with topic maps because tools did not exist to manipulate or visualise topic maps. This situation is changing and there are now a number of tools freely available to developers. Some of them are:

- **Ontopia Navigator**: allows you to navigate topic maps in a generic interface
- **XSLT stylesheets**: for the transformation of the numerous existing XML DTDs for topic maps. The stylesheets are designed to convert topic maps from the syntax used by tools from Cogitech, Empolis, Infoloom, Ontopia and Techquila.com into the XTM syntax.
- **MDF**: a combination of a simple approach to creating reusable modules for the processing of metadata and an implementation of that approach using JAVA". The framework has been designed to make it easy to trawl resources, extract and clean metadata and finally write them out as a topic map.
- **Tmproc**: is an open source Python topic map engine.
- **DTD of SHOE**: It will be used in conjunction with the DTD of the Topic Maps.
- **TM4J-API**: is an open source API that allows you to work with Topic Maps as JAVA objects, create Topic Maps, edit and merge Topic Maps, read and write XTM etc.
- **K42-API**: similar to TM4J-API but not open source

The tools that are going to be used in REGNET implementation are: The Ontopia Navigator, the Document Type Definition of SHOE and the TM4J-API.

5.7.8 REGNET connector

Regent connector will be implemented in JAVA language. Apache SOAP described before can be used to implement SOAP functionnalities.

REGNET business profile will be store in a public UDDI registry (XML.org for example).

A public UDDI registry indexes all publicly distributed Web Services. Every user sees these Web Services through an inquiry interface. Anyone may register with a public UDDI registry and populate it with their own Web Services through the publishing port. Every registration is automatically replicated between all public UDDI operators so that any public node contains an up-to-date record of Web Service offerings.

5.7.9 Synthesis

This table contain a synthesis of software tools necessary to the development and test of the REGNET system.

Function	Tool	Comment
Web server	Apache + tomcat	Open source
J2EE application server	JBOSS	Open source
Object / relational mapping	Castor	Open source
SOAP/JAVA	Apache SOAP	Open source
PHP		
SOAP/PHP		

Function	Tool	Comment
Client : Applet	JAVA Web start	Available for free from Sun
XML Editor		
XML Server	DbXML Textml	Open source
Relational Database	MySQL	Open source
Harvester		
Connector Z39.50		
Operating system	Linux	Open source
Development tool: JAVA IDE	Forte for JAVA	

Table 50: Development Tools

5.8 Deployment

To be referenced in the introduction

This chapter gives some examples how REGNET-Sites can look like and should show the way the REGNET System is deployed.

First a full REGNET-Site is shown. Then 4 sample scenarios for REGNET-Sites will follow. To get these different characteristics for the Sites a set-up tool has to provide a sophisticated configuration tool for a Site and the sub-nodes in the site respectively.

5.8.1 REGNET-Site

The following diagram shows all the components of the system, including their interaction (denoted with arrows). Such an connection means that a component uses functionalities provided by the other component.

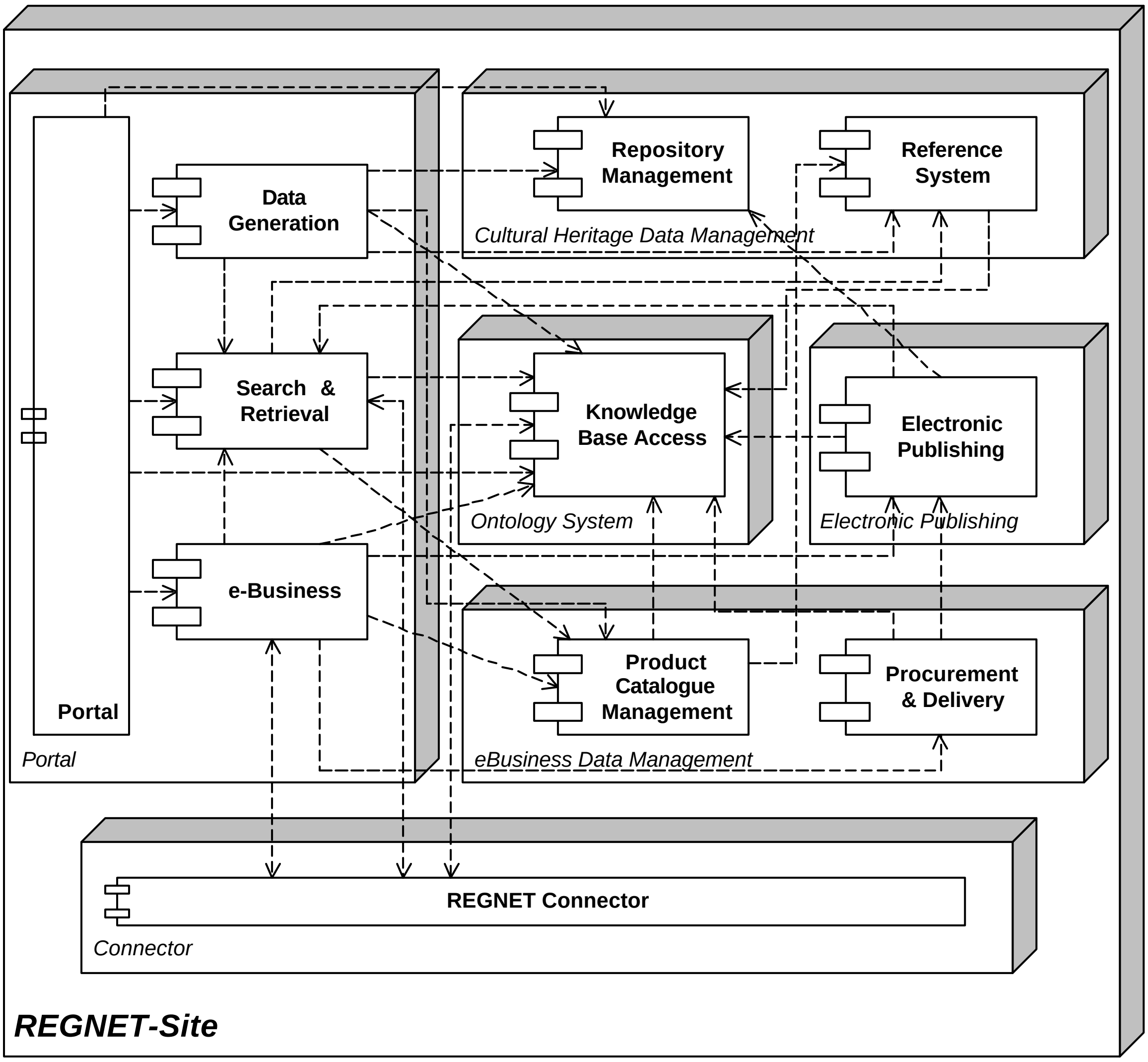


Figure 100: Deployment Diagram - Connections within one REGNET-Site

The full REGNET-Site consists of 11 components (or subsystems) and 6 nodes. Mainly the regional cultural service centres will run all the components in one Site (but also others can). For single institutions different configurations of the Sites will be possible. As example 4 scenarios will be illustrated.

5.8.1.1 The Ontology Master Site

There will be one Site which acts as Ontology Master Site. The main task of this site is to hold a "global" version of the Ontology System thus the Site consists only of the Ontology System and the Connector. The exact mechanism how the dher REGNET-Sites are updated with new data from the Ontology System and how data in the Ontology System is updated through other REGNET-Sites has to be clarified in WP2. Also its not clear how to manage the Ontology System. As it is a central Node for all Sites, an administration tool for this node will be necessary.

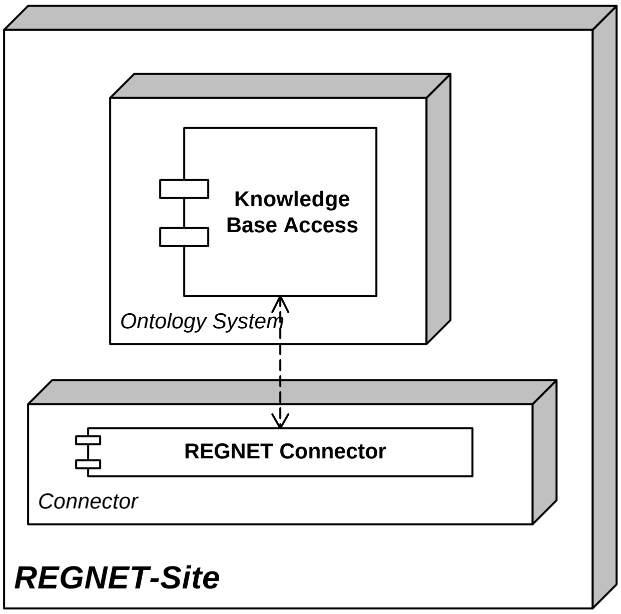


Figure 101: Deployment Diagram - Ontology master site

5.8.1.2 Union Catalogue

The Union Catalogue is a typical scenario for institutions with a few different collections. The union catalogue scenario provides data generation facilities (i.e. creating, updating and deleting data), the Cultural Heritage data management facilities (local storage of the institutions CH data) and local (through the Search & Retrieval component) and remote (through the Connector) access to the data.

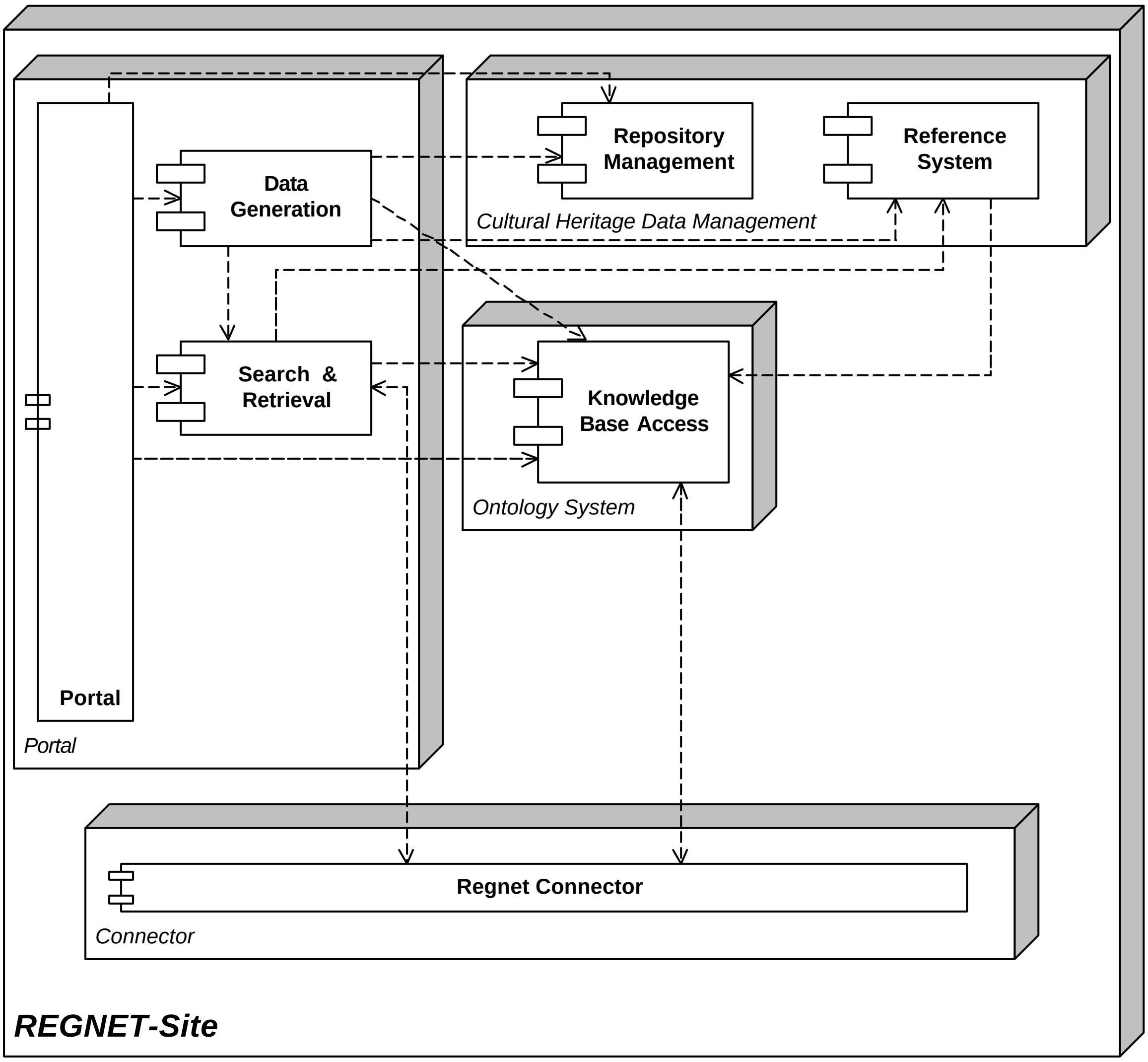


Figure 102: Deployment Diagram - Union Catalogue Site

5.8.1.3 The Metasearch-Model

The Metasearch-Model provides principally only a Search & Retrieval functionality. In this scenario the Cultural Heritage data is already available or is generated with external tools. An institution which has already data available and wants to join the REGNET "Network" is a possible user for such a REGNET-Site.

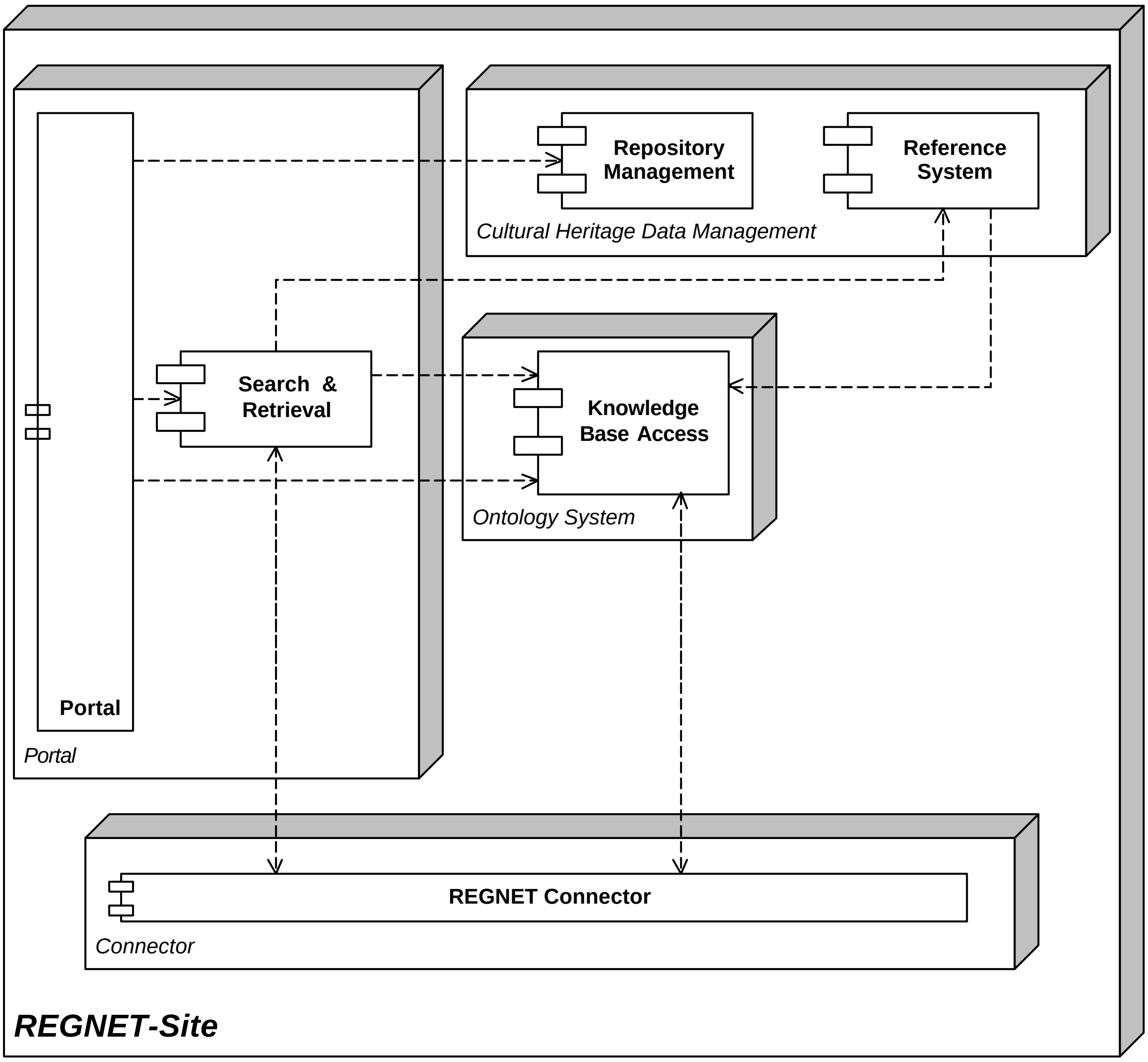


Figure 103: Deployment Diagram - Metasearch-Model

5.8.1.4 Museum Shop

This scenario can be supposed for a small museum shop. Mainly there are the eBusiness parts of the Site included. The Search & Retrieval component is needed for browsing the available Product Catalogue.

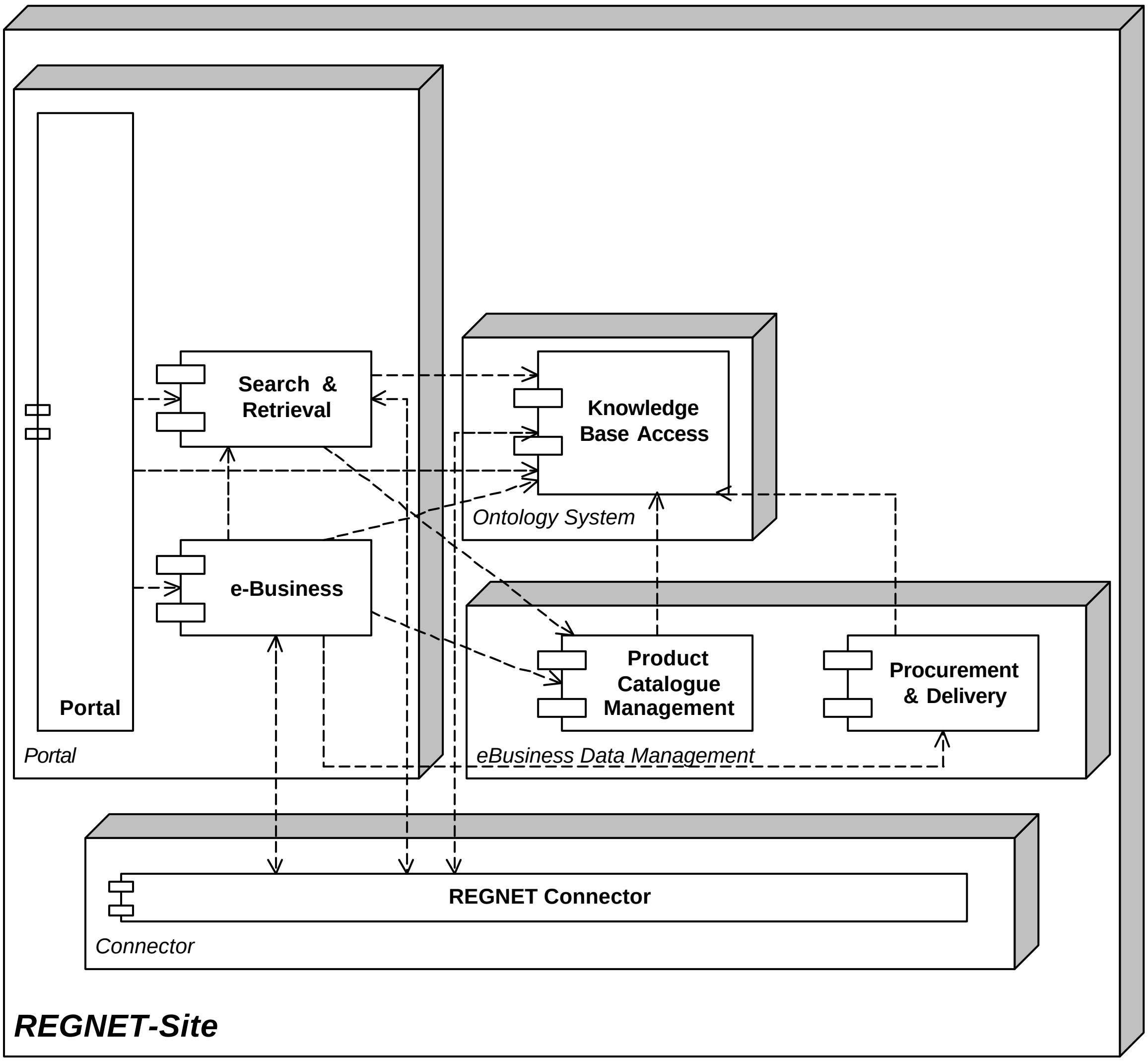


Figure 104: Deployment Diagram - Museum Shop

5.8.1.5 Publishing Node

This following scenario shows which subsystems are involved in a publishing process.

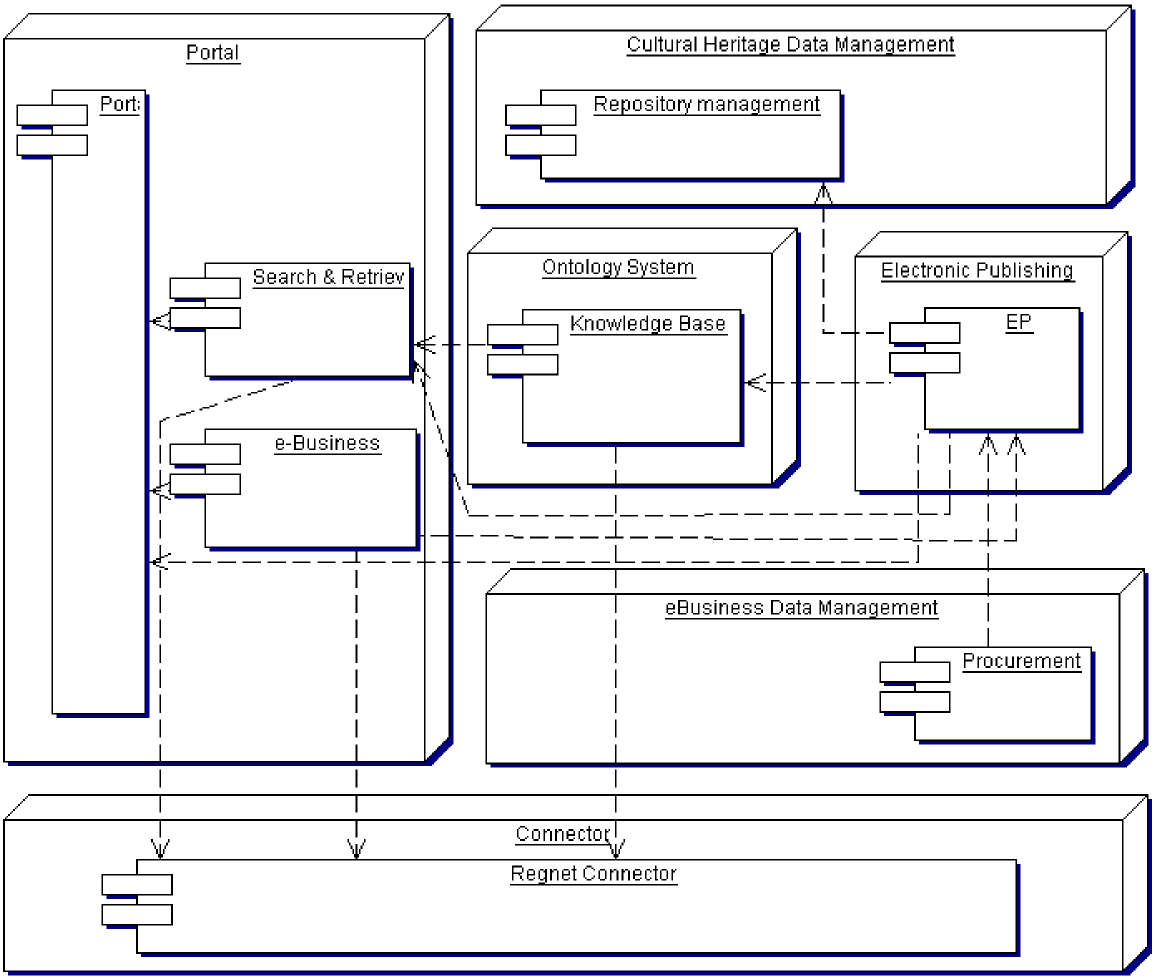


Figure 105: Deployment Diagram - Publishing Node

5.8.2 The REGNET Network

Above some of the possible scenarios for REGNET Sites were shown. All Sites have two things in common. The Ontology System, which acts as a general knowledge source for all components, and the Connector. The Connectors of all REGNET Sites are connected with each other and provide controlled access to the single Sites. The Ontology Master Site has a kind of exceptional position. It contains only an Ontology System and has no interface for users (there might be an interface for administrating the Ontology System). New Ontology data collected or entered in one of the REGNET Sites will be forwarded to the Ontology Master Site. The Ontology Master Site will then distribute the new data to all other REGNET Sites.

Also when installing a REGNET-Site an actual copy of the Ontology System can be provided by the Ontology Master Site.

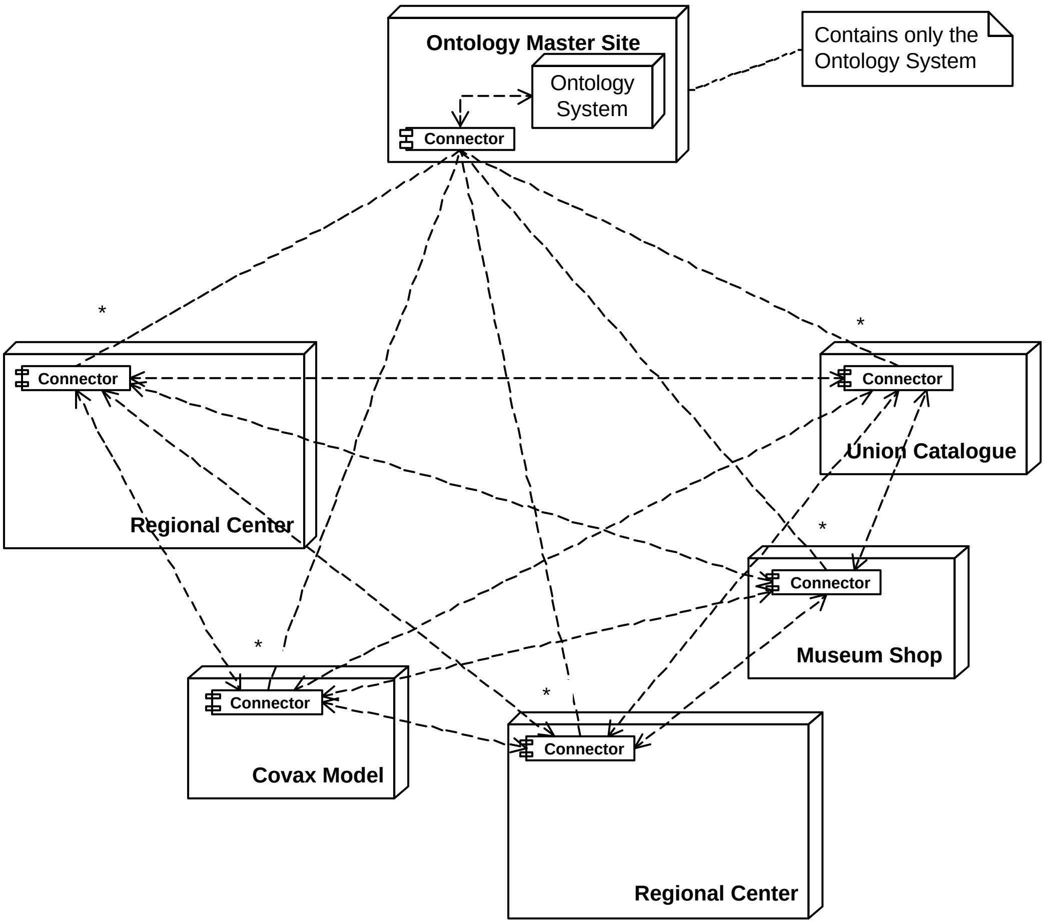


Figure 106: Deployment Diagram - Example REGNET Network

5.9 Interfaces

In order to specify the Interfaces of the REGNET nodes, it is important to clarify the exact interactions that exist between the several subsystems.

In the following table all the connections between the REGNET nodes are shown:

		1	2	3	4	5	6	7	8	9	10	11
		RM	RS	KB	DG	SR	EB	PCM	EP	PD	PO	RC
1	RM											
2	RS			x								
3	KB											x
4	DG	x	x	x		x		x				
5	SR		x	x				x				x
6	EB			x		x		x	x	x		x
7	PCM		x	x								
8	EP	x		x		x						
9	PD			x					x			
10	PO	x		x	x	x	x					
11	RC			x		x	x					x

Table 51: Connections between REGNET Components

Nodes

Nr	Name	abbreviation	responsible partner
1	Repository Management	RM	SPAC
2	Reference System	RS	AIT
3	Knowledge base access	KB	SI
4	Data Generation	DG	SPAC
5	Search & Retrieval	SR	AIT
6	e-Business	EB	ZEUS
7	Product Catalogue Management	PCM	ZEUS
8	Electronic Publishing	EP	SR
9	Procurement & Delivery	PD	VALT
10	Portal	PO	MOT
11	REGNET Connector	RC	VALT

Table 52: Abbreviations

Connections:

A cross between a line and a column indicate that the line component request a service from the column component.

The interaction between the modules are detailed in the following sections showing the information exchanged between the modules. This table will act as a reference in WP2 for defining the interfaces between the subsystems on code level.

The table describing the relationships is structured as follows:

- *Subsystem:* The subsystem mentioned in the headline communicates with the subsystem mentioned in this column.
- *Function:* Describes the functionality for which the information exchange is needed.
- *Input / Output:* Classifies the information exchanged either as input or output data.

- *Information Exchanged:* The data exchanged between the communicating modules.
- *Comments:* Comments for clarifying the purpose of the information exchange.

5.9.1 Portal

Subsystem	Function	Input / Output	Information Exchanged	Comments
Search & Retrieval	Search & Retrieval Requests	Input	DTDs, XML files	Create the base query
		Output	XML files	XML file containing the search results
Data Generation	Data Generation Requests	Input		Provide an access point for DG client
e-Business	e-Business Requests	Input	DTDs, XML files expressed in WSDL (Web Services Description Language)	Create the e-Business requests
		Output	XML files representing the outcomes	Results of the e-Business transactions
Knowledge Base	Knowledge Base Requests	Input	DTDs, XML files	Users and terminals profiles involved
		Output	XML files describing User & Terminal Profiles	Users and terminals profiles involved
Repository Management	Repository Management Requests	Output	XML files referencing the real objects	-

Table 53: Services (Portal)

5.9.2 Data Generation

Subsystem	Function	Input / Output	Information Exchanged	Comments
Knowledge Base	Knowledge Base Requests	Input	DTDs, XML files	Users and terminals profiles involved
		Output	XML files describing User & Terminal Profiles	Users and terminals profiles involved
Repository Management	Saving of Data	Input	Data (XML files, binary data)	upload of data, import of data, etc.
	Data Management	Input	Schemas	Administration of the Data Repositories
Reference System	Provide Access to Cultural Heritage Data	Input	query	The query format will mainly follow the Z39.50 protocol (with some additional REGNET specific information)

		Output	response document in XML	the response will be formatted as defined within the query structure
Search and Retrieval	Data Access	Input	DTDs, XML files	Create the base query
		Output	XML files	XML file containing the search results
Product and Catalogue Management	Product Management	Input / Output	Product Information	

Table 54: Services (Data Generation)

5.9.3 Search and Retrieval

Subsystem	Function	Input / Output	Information Exchanged	Comments
Reference System	Provide Access to Cultural Heritage Data	Output	query (following a XML-Schema to be defined in WP2)	The query format will mainly follow the Z39.50 protocol (with some additional REGNET specific information)
		Input	response document in XML	the response will be formatted as defined within the query structure
Knowledge Base	Manage User Data	Output / Input	user profile	user data
	Session Management	Output / Input	XML-Schemas/DTD’s/ Stylesheets	for checking and processing requests and responses
	User support through a thesaurus system	Output	query to thesauri / Topic Map Server	assistance for users with their searches
		Input	Topic map entries in XTM	
	Manage repository information	Input / Output	repository information	detailed information about repositories in the system. (to dispatch the queries to the right targets)
Product Catalogue Management	Access to e-Business data	Output	query to the Product Catalogue Management	for searches in e-Business related data
		Input	response document in XML	the response will be formatted as defined within the query structure
Connector	Access to other REGNET nodes	Output	query to other REGNET Sites	the queries can be distributed to all other connected REGNET Sites. these queries will be processed by Search & Retrieval components included within these other Sites

		Input	response document in XML	the response will be formatted as defined within the query structure
Data Generation, Connector, e-Business, Electronic Publishing, Portal	Access to distributed repositories	Input	query (following a XML-Schema to be defined in WP2)	a common query Schema will be developed which should be used by all components that want to send queries to the Search & Retrieval component
		Output	response to the query in XML	the response is built by merging all results from the different targets to one result.
Portal	Provide statistic and status information	Input	query statistical data of the Search & Retrieval component	data about usage of this component or about status of the currently running searches
		Output	statistical data	information for users
	Query management	Input	request for saved queries for a user	to provide refine function and a search history
		Output	saved queries	the queries a user has previously done (automatically saved) or saved (user defined)

Table 55: Services – Search & Retrieval

5.9.4 e-Business

Subsystem	Function	Input / Output	Information exchanged	Comments
Knowledge Base	Repository management	Input / Output	repository information	detailed information about repositories in the system. (to dispatch the queries to the right targets)
	User management	Output / Input	user profile	user data: necessary for checking rights (for create, update and delete actions)
Search & Retrieval	Search & Retrieval Requests	Input	DTDs, XML files	Create the base query
		Output	XML files	XML file containing the search results
Product Catalogue Manage-ment	Retrieve Product Information	Input	Product information	-
		Output	Terms, Conditions, Prices, Suppliers	-
Electronic Publishing	Product generation	Input / Output	Product information, Terms, Conditions, Restrictions	For parts and end product
Procure-ment and Delivery	Product generation	Input	Terms, Conditions	-
	Product selling	Output	Terms, Conditions, Costs, Restrictions	-
REGNET Connector	Cross selling / buying	Input / Output	Price, Terms, Conditions	-

Table 56: Services – e-Business

5.9.5 Repository Management

No services needed from other subsystems.

5.9.6 Reference System

Subsystem	Function	Input / Output	Information exchanged	Comments
Knowledge Base	Session management	Output / Input	XML-Schemas/DTD's/ Stylesheets	for checking and processing requests and responses
	Repository management	Input / Output	repository information	detailed information about repositories in the system. (to dispatch the queries to the right targets)
	User management	Output / Input	user profile	user data: necessary for checking rights (for create, update and delete actions)
Data Generation	Repository management	Input	Repository information	2 main functionalities are provided for repositories: create delete
	Repository management	Input	Item data (digital surrogates of real objects) – following a XML-Schema/DTD used in REGNET (e.g. AMICO, MARC, …)	3 main functionalities are provided to manage data within repositories: insert update delete
Reference System	Search	Input	query (following a XML-Schema to be defined in WP2)	The query format will mainly follow the Z39.50 protocol (with some additional REGNET specific information)
		Output	response document in XML	the response will be formatted as defined within the query structure

Table 57: Services – Reference System

5.9.7 Product Catalogue Management

Subsystem	Function	Input / Output	Information Exchanged	Comments
Knowledge Base	Repository management	Input / Output	repository information	detailed information about repositories in the system. (to dispatch the queries to the right targets)
	User management	Output / Input	user profile	user data: necessary for checking rights (for create, update and delete actions)
Reference System	Search	Input	XER request (query), user information	for distributed search

		Output	search result	
	Meta Data Management	Input / Output	DTDs, XML, XML-Schemas, Location of repositories Administrative info	create retrieve meta data

Table 58: Services – Product Catalogue Management

5.9.8 Procurement and Delivery

Subsystem	Function	Input / Output	Information Exchanged	Comments
Knowledge Base	Repository management	Input / Output	repository information	detailed information about repositories in the system. (to dispatch the queries to the right targets)
	User management	Output / Input	user profile	user data: necessary for checking rights (for create, update and delete actions)
Electronic Publishing	Product Information	Output	Description of product produced and content used for production; payment and transportation information	-

Table 59: Procurement and Delivery

5.9.9 Knowledge Base Access

Subsystem	Function	Input / Output	Information Exchanged	Comments
Search & Retrieval	Search & Retrieval Request	Input	DTDs XML, XML-Schemas Topic Maps	Create metadata
		Output	Thesauri information Themes (Topic Maps)	Respond to Search queries
Data Generation	Data Generation Requests	Input	DTDs XML, XML-Schemas Topic Maps	Create metadata
		Output	User & Terminal Profiles	Know the users and the terminals involved
e-Business	e-Business Request	Input	DTDs XML, XML-Schemas Topic Maps	Create metadata
		Output	User & Terminal Profiles	Know the users and the terminals involved
Product Catalogue Management	Product Catalogue Request	Input	DTDs XML, XML-Schemas Topic Maps	Create metadata

		Output	User & Terminal Profiles	Know the users and the terminals involved
Procurement & Delivery	Procurement & Delivery Request	Input	Topic Maps	Create metadata
		Output	User & Terminal Profiles	Know the users and the terminals involved
Electronic Publishing	Electronic Publishing Request	Input	DTDs XML, XML-Schemas Topic Maps	Create metadata
		Output	Style Sheets (XSL) Binary Data (java classes)	Publish document with the correct format
Reference System	Reference System Request	Input	DTDs XML, XML-Schemas Topic Maps Location of repositories Administrative info	Create metadata
		Output	User & Terminal Profiles	Know the users and the terminals involved
Portal	Portal Request	Input	DTDs XML, XML-Schemas Topic Maps	Create metadata
		Output	User Profiles Terminal Profiles	Know the users and the terminals involved

Table 60: Services – Knowledge Base Access

5.9.10 Electronic Publishing

Subsystem	Function	Input / Output	Information exchanged	Comments
Search and Retrieval	Search & Retrieval – Query	Input	Request Query	Search for objects to be published
		Output	Themes in XTM Resultset XER, AMICO, MARC	
Ontology	Ontology-Storyboard	Input	Request Query	Import of Storyboard
		Output	XSL or Binary data (e.g. JAVA class file)	
Ontology	Ontology-Stylesheet	Input	Request Query	Import of Storyboard

		Output	XSL	
Reference System	Reference System – artefact	Input	Request URL	Get picture of the artefact Get external resource input
		Output	Binary data (e.g. JPEG, GIF)	

Table 61: Services – Electronic Publishing

5.9.11 Syntax of Interfaces

To describe the interfaces in WP2 the WSDL will be used. Each WSDL file will describe the separate interface, which is dedicated to access each subsystem. This amount of the descriptions depends on the big amount of the related components, and also the big variety of the exchanging information. Also each subsystem will have a different access rights to the stored information of each subsystem

An example of a WSDL interface is given below (complete interface is given into annexe in the part dedicated to WSDL):

The generation of the WSDL interface from the java code was done using glue java2wsdl processor (www.themindelectric.com in standard version).

WSDL interface must maps complex types used to make business information transit from the web service (soap server) and its client, and declare the web services name to make its services available to clients

Example:

The web service called “getProduct” returns a “Product” business object description extracted from the database using the productId (id: string type) given as an input parameter.

```
<message name='getProduct9SoapIn'>
    <part name='id' type='xsd:string'/>
  </message>
  <message name='getProduct9SoapOut'>
    <part name='Result' xmlns:ns1='http://www.themindelectric.com/package/com.REGNET.marketplace.business/' type='ns1:Product'/>
  </message>
```

Clients are able to get a business object of type “Product”. It means that he is able to map xml flow from the soap server using the next description:

```
      <complexType name='Product'>
        <sequence>
          <element name='description' nillable='true' type='string'/>
          <element name='id' nillable='true' type='string'/>
          <element name='name' nillable='true' type='string'/>
          <element name='price' nillable='true' type='string'/>
          <element name='refsupplier' nillable='true' type='string'/>
```

```
    <element name='supplier' nillable='true' type='string'/>
    <element name='thumbnail' nillable='true' type='string'/>
    <element name='REGNETId' nillable='true' type='string'/>
  </sequence>
</complexType>
```

(this description allows client to generate proxies able to marshall data from xml flow extracted from soap server)

All the messages will contain the different elements of the data, that’s why it’s necessary to attract more attention to the following tags of the WSDL document:

<type>

…………

</type>,

<message>

……….

</message>,

<binding>

………..

</binding> ,

because they describe the structure, order, transmitted messages which was mentioned earlier.

This is a sample data interchange schema between Portal and:

- e-Business
- Knowledge Base
- Search System

```
<EBusiness>
       <objectID/>
       <supplierID/>
       <quantity/>
</EBusiness>



<UserProfile>
       <userID/>
       <userName/>
       <password/>
       <language/>
       <address/>
       <phoneNumber/>
       <email/>
       <userType>
              <guest/>
              <member/>
              <admin/>
       </userType>
</UserProfile>


<TerminalProfile>
       <connectionType>
              <wireless/>
              <wireline/>
       <\connectionType>
       <browserType>
              <netscape/>
              <iExplorer/>
              <motorola/>
              <openwave/>
       <\browserType>
       <bandwidth>
              <lan/>
              <adsl/>
              <56Kbps/>
              <28Kbps/>
              <9Kbps/>
       <\bandwidth>
<\TerminalProfile>


<Search>
       <userID/>
       <source/>
       <sessionID/>
<\Search>
```

5.9.12 Protocols & Metadata

Protocols used comprise **TCP/IP**, **ISO 23950** and e-Business related as far as available. **WAP** will be used to access the REGNET 'business access points'. Data exchange will be based on the different XML developments already available in the field of Cultural Heritage (XML/**Dublin** Core, **EAD**-XML, etc) and e-Business (e.g. XML/**EDI**, **ebXML**, etc).

5.9.13 Programming Languages and Interfaces

As stated in this document programming languages used are PHP and JAVA. Components build according to this heterogeneous technologies will communicate through SOAP.

Interfaces will be based on WSDL and stored in UDDI registry if necessary.

Communication between homogeneous technologies (JAVA to JAVA for example) will be based on adapted technologies (RMI for JAVA).

Interfaces will relational database will be based on standard SQL.

5.10 Conclusion

This document produced both, the functional and technical architecture of the REGNET System according to the two track process described. These architectures allows to build the REGNET system on a sound basis and is a good starting point for the development within WP2.

In addition, this document allows to identify main risks in the very beginning of the project. As mentioned in the software development plan, these risks will be evaluate in the first increment in order to be eliminate as soon as possible. Moreover it is very important to notice that this document is not frozen: it is going to be updated during the whole project.

6 Glossary and acronym list

Acronym	Description	Link
A2A	Application to Application Integration	
AAT	Arts and Architecture Thesaurus	http://www.getty.edu/research/tools/vocabulary/aat/index.html
ADO	ActiveX Data Object	
AF	Architectural Forms (HyTime)	http://xml.coverpages.org/hytime.html
AHDS	Arts and Humanities Data Service	http://ahds.ac.uk/
AITF	Art Information Task Force	http://www.getty.edu/research/institute/standards.html
AMICO	Art Museums Image Consortium	http://www.amico.org/
ANSI	American National Standards Institute	http://www.ansi.org/
Apache JServ	Servlet Engine fully Servlet API 2.0 compliant	http://java.apache.org/jserv/
Apache Tomcat	Reference Implementation of JAVA Servlet and JAVA Serverpages Technologie	http://jakarta.apache.org/tomcat/
API	Application Programming Interface	
ASN.1	Abstract Syntax Notation One	http://www-sop.inria.fr/rodeo/personnel/hoschka/asn1.html
B2B	Business to Business	
B2C	Business to Consumer	
B4B	Business for Business	
BATIK	Toolkit for developing SVG-Applications	http://xml.apache.org/batik/
BEA Weblogic	The #1 Web and wireless application server	http://www.bea.com/products/weblogic/server/index.shtml
BizTalk	Resources for Learning XML	http://www.biztalk.org
BOV	Business Operational View - One View of UMM	
BUC	Business Use Case	
C2C	Consumer to Consumer	
CatXML	Open interoperable standard for catalogue information exchanges	http://www.catxml.org/
CDWA	Categories for the Description of Works of Art	
CDWF	Categories for the Description of Works of Art	http://www.getty.edu/research/institute/standards/cdwa/

CEN/ISSS	European Committee for Standardization/ Information Society Standardization System	http://www.cenorm.be/isss/
CGI	Common Gateway Interface	
CH	Cultural Heritage	
CIDOC	Le Comité international pour la documentation du Conseil international des musées	http://www.cidoc.icom.org/
CIMI	Consortium Museum Intelligence	http://www.cimi.org/
CIO	Cultural Institutions and Organisations	
COCOON	Publishing Framework	http://xml.apache.org/cocoon/
CommerceNet		http://www.commercenet.com
Component-X	**XML- and JAVA-based Web services platform**	http://www.enterprise-component.com/
CORBA	Common Object Request Broker Architecture	
COVAX	Contemporary Culture Virtual Archives in XML	http://www.covax.org/
CPP	Collaboration Protocol Profile	http://www.ebxml.org
CRM	Conceptual Reference Model	http://cidoc.ics.forth.gr
CS	Coding Schemes (MPEG- 7)	
DBD	Data Base Driver	
DBI	Data Base Interface	
dbXML	native XML Database	http://www.dbxml.org/
DC	Dublin Core	http://dublincore.org/
DCES	Dublin Core Element Set	http://dublincore.org/documents/dces/
DCMES	Dublin Core Metadata Element Set	http://dublincore.org/documents/dces/
DCMI	Dublin Core Metadata Initiative	http://dublincore.org/
DCOM	Distributed Component Object Model	
DDL	Description definition language (MPEG- 7)	
DESIRE	Project focussed on enhancing existing European information networks for research users across Europe	http://www.desire.org/
DG	Data Generation: Part of the REGNET System	
DHTML	Dynamic HTML	http://builder.cnet.com/Resources/Info/Glossary/Terms/dhtml.html
DMZ	De- militarized Zone	
DOM	Document Object Model	http://www.w3.org/DOM/
DS	Description Scheme (MPEG- 7)	

DSMCC	Digital Storage Media Command and Control	
DTD	Document Type Definition	
DVD	Digital Versatile Disc	
EAD	Encoded Archival Description	http://www.loc.gov/ead/
EAI	Enterprise Application Integration	
EB	E- Business	
EBNF	Extended Backus-Naur Form	http://www.cl.cam.ac.uk/~mgk25/iso-ebnf.html
EBU	European Broadcasting Union	http://www.ebu.ch/
EBU P/META	Metadata Project of EBU	http://www.ebu.ch/pmc_meta.html
ebXML	Goal: To provide an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties.	http://www.ebxml.org
EC	Electronic Components	
ECMA	European Computer Manufacturers Association Standardizing Information and Communication Systems	http://www.ecma.ch/
EDI	Electronic Data Interchange	
EDIFACT	EDI standard	http://www.edifact-wg.org
EJB	Enterprise JavaBeans	http://java.sun.com/products/ejb/
ELISE	Electronic Library Image Service for Europe	http://nile.dmu.ac.uk/elise/e2_intro.html
EP	Electronic Publishing: Part of the REGNET System	
ERP	Enterprise Resource Planning	
EULER	The aim of the project is to provide strictly user-oriented, integrated network based access to mathematical publications.	http://www.emis.de/projects/EULER/
FR	Functional Requirements	
FSV	Functional Service View	http://www.ebxml.org
GPRS	General Packet Radio Service	
GRINS	A Smil Editor	http://www.oratrix.com/GRiNS/
GUI	Graphical User Interface	
HDML	Handheld Device Markup Language	
HL7	Health Level 7 - a medical Standard	http://www.hl7.org/
HTML	Hyper Text Markup Language	
HTML+TIME	Timed Interactive Multimedia	http://www.w3.org/TR/NOTE-

	Extensions for HTML, a w3c submission by Microsoft	HTMLplusTIME
HTTP	Hyper Text Transfer Protocol	
HTTPS	Hyper Text Transfer Protocol over SSL	
HyTime	The HyTime standard actually defines a number of related but largely independent architectures and facilities	http://www.hytime.org/
IAI	**Internet Application Integration**	
IASA	International Association of Sound and Audiovisual Archives	http://www.iasa.org/
ICA	International Council of Archives	http://www.ica.org/
ICOM	International Council of Museums	http://www.icom.org/
IDL	Interface Description Language	
IETF	The Internet Engineering Task Force	http://www.ietf.org/
IFLA	International Federation of Library Associations and Institutions	http://www.ifla.org/
IIOP	Internet Inter-ORB Protocol	
INTERMARC	a MARC format	
IONA/NETFISH	Company/product	http://www.iona.com/
ISAD(G)	General International Standard Archival Description	http://lettere.unipv.it/obc/add/infap/archdes/isad(g)e.html
ISBD	International Standard Bibliographic Description	http://www.alice.it/library/law.lib/isbd.htm
ISO 12207	This standard describes the major component processes of a complete software life cycle, their interfaces with one another, and the high-level relations that govern their interactions	http://www.software.org/quagmire/descriptions/iso-iec-12207.asp
ISO 23950	Information retrieval protocol standard (=ANSI/NISO Z39.50)	http://www.loc.gov/z3950/agency/ http://www.niso.org/z3950.html
ISO 2907	Defines a record structure (= ANSI/NISO Z39.2)	http://www.loc.gov/marc/concise/concise.html
ISO/IEC JTC 1/SC 29	Sub Committee of the Joint Technical Committee 1 of the ISO/IEC dealing with coding of Audio, Picture, and Multimedia and Hypermedia Information	http://www.jtc1.org/
J2EE	JAVA 2 Platform, Enterprise Edition	http://java.sun.com/j2ee/
J2ME	JAVA 2 Platform, Micro Edition	http://java.sun.com/j2me/
J2SE	JAVA 2 Standard Edition	http://java.sun.com/j2se/
JAVA 2D	JAVA 2D API	http://java.sun.com/products/java-media/2D/

JAVA 3D	JAVA 3D API	http://java.sun.com/products/java-media/3D/
JAVA Applet	JAVA Applets	http://java.sun.com/applets/?frontpage-spotlight
JAVA Plugin	JAVA Plugin	http://java.sun.com/products/plugin/
JAVA Servlet	JAVA Servlet Technology	http://java.sun.com/products/servlet/
JAVA WebStart	JAVA Application Deployment Technology	http://java.sun.com/products/javawebstart/
JBOSS	Open Source Enterprise JavaBeans Application Server	http://www.jboss.org/
JDBC	JAVA DataBase Access API	http://java.sun.com/products/jdbc/
JetSpeed	Open source Implementation of an Enterprise Information Portal using JAVA and XML	http://jakarta.apache.org/jetspeed/site/index.html
Jext	A JAVA Text Editor	http://www.jext.org/
JMS	JAVA Messaging Service	
JNDI	JAVA Naming and Directory Interface	
JRMP	JAVA Remote Method Procedure call	
JSP	JAVA Server Pages	
JTA	JAVA Transaction API	
JTS	JAVA Transaction Service	
JVM	JAVA Virtual Machine	
JZKit	JAVA toolkit for building distributed information retrieval systems	http://www.k-int.com/jzkit/
k42	Knowledge Server	http://k42.empolis.co.uk/home.html
KB	Knowledge Base Access: Part of the REGNET System	
LATCH	Location, Alphabet, Time, Category, Hierarchy	
LDAP	Lightweight Directory Access Protocol	
MALVINE	Manuscripts and Letters via Integrated Networks in Europe	http://www.malvine.org/
MARC	The MARC formats are standards for the representation and communication of bibliographic and related information in machine-readable form.	http://www.loc.gov/marc/
MARC 21	A MARC subformat	http://lcweb.loc.gov/marc/umb/um07to10.html
MDF	Metadata Processing Framework	http://www.techquila.com/mdf.html

Mercator	E- Business Integration Broker	http://www.mercator.com/products/index.html
MHEG-1	MHEG object representation-base	http://www.km.giti.waseda.ac.jp/WG12/
MHEG-5	Support for base-level interactive applications	http://www.km.giti.waseda.ac.jp/WG12/
MHEG-6	Support for enhanced interactive applications	http://www.km.giti.waseda.ac.jp/WG12/
MHEG-7	Interoperability and conformance testing	http://www.km.giti.waseda.ac.jp/WG12/
MHEG-8	Standard to provide XML encodings for ISO/IEC 13522-5 (MHEG-5)	http://www.km.giti.waseda.ac.jp/WG12/
Microsoft SQL Server	Fully Web-enabled database product, providing core support for Extensible Markup Language (XML) and the ability to query across the Internet and beyond the firewall.	http://www.microsoft.com/sql/default.asp
MIDlet	A MIDlet is a JAVA application that conforms to the specifications set out by the Connected, Limited Device Configuration (CLDC) and the Mobile Information Device Profile (MIDP).	http://developer.java.sun.com/developer/products/wireless/midp/articles/intro/
MOM	Message Oriented Middleware	
MOSAIC	Museums Over States And vIrtual Culture	http://mosaic.infobyte.it/
MPEG- 7	Will be a standardized description of various types of multimedia information	http://www.darmstadt.gmd.de/mobile/MPEG7/Documents/ w4006.htm
MPEG-4	The MPEG-4 format is meant to become the universal language between broadcasting, movie and multimedia applications.	http://www.iis.fhg.de/amm/techinf/mpeg4/index.html
MySql	Open Source Database	http://www.mysql.com/
NCSU	North Carolina State University	http://www.csc.ncsu.edu/
NEDLIB	NEDLIB is a collaborative project of European national libraries It aims to construct the basic infrastructure upon which a networked European deposit library can be built.	http://www.kb.nl/coop/nedlib/
Netfish	see IONA/NETFISH	
NWI	Nordic Web Index	http://nwi.dtv.dk/nwi_info.html
OASIS	Organization for the Advancement of Structured Information Standards	http://www.oasis-open.org/
OBI	Open Buying on the Internet	http://www.openbuy.org/
OCR	Optical Character Recognition	
ODBC	Open Database Connectivity	
OMG	Object Management Group	http://www.omg.org/

ONTOLOGY	An ontology is a specification of a conceptualisation.	http://www-ksl.stanford.edu/kst/what-is-an-ontology.html
Ontopia Navigator	Tool for navigating Topicmaps	http://www.ontopia.net/
OODBMS	Object Oriented Database Management Systems	
OPAC	Online Public Access Catalogue	
OpenEDI	B2B Transaction Engine	http://www.pentagon2000.com/p_edi.html
ORB	Object Request Broker	
OTM	Object Transactional Monitor	
PCM	Product Catalogue Management: Part of the REGNET System	
PCM	Pulse Code Modulation	
PD	Procurement and Delivery: Part of the REGNET System	
PDA	Personal Digital Assistant	
PDF	Portable Document Format	
PHP/YAZ	extension to PHP that implements Z39.50 origin (client) functionality.	http://www.indexdata.dk/phpyaz/
PIP	Partner Interface Process. Defined by RosettaNet	
PSTN	Public Switched Telephone Network	
PT	Portal: Part of the REGNET System	
QT	Query Translator (see SOAP)	
Rational Unified Process	Web-enabled software engineering process that enhances team productivity and delivers software best practices to all team members.	http://www.rational.com/products/rup/index.jsp
RDBMS	Relational Database Management System	
RDF	Resource Description Framework	http://www.w3.org/RDF/
REGNET	Cultural Heritage in REGional NETworks	http://www.REGNET.org/
RM	Repository Management: Part of the REGNET System	
RMI	Remote Method Invocation	http://java.sun.com/products/jdk/rmi/
RNIF	RosettaNet's Implementation Framework	http://www.rosettanet.org
RPC	Remote Procedure Call: a call to a routine that results in code being executed on a different system from the one where the request originated. An RPC system allows calling procedures and called procedures to execute on different systems without the programmer	

	needing to explicitly code for this.	
RS	Reference System: A subsystem of the REGNET System.	
S&R	Search and Retrieval: A subsystem of the REGNET System.	
SAX	SAX is a standard interface for event-based XML parsing, developed collaboratively by the members of the XML-DEV mailing list and OASIS. Currently in version 2.0.	
SCHEMAS	EU project that provides a forum for metadata schema designers involved in projects under the IST Programme and national initiatives in Europe.	http://www.schemas-forum.org
Schmunzel	Interactive SMIL 1.0 compliant player developed by Salzburg Research Forschungsgesellschaft m.b.H., member of the REGNET consortium	http://www.salzburgresearch.at/
SET	Secure Electronic Transactions is a system for ensuring the security of financial transactions on the Internet developed by VISA and Master Card.	
SGML	Standard Generalised Markup Language (ISO 8879). A generic markup language for representing documents. SGML is a system for defining structured document types, and markup languages to represent instances of those document types.	
Shiloh	Product extension for the MS SQL-Server for the mapping of XML structure to RDBMS	
SHOE	Simple HTML Ontology Extensions which allows web page authors to annotate their web documents with machine-readable knowledge.	http://www.cs.umd.edu/projects/plus/SHOE/
Simple-EDI	Object oriented extension for EDI	
SM	Semiconductor Manufacturing	
SME	Small and Medium sized Enterprises.	
SMIL	Synchronised Multimedia Integration Language is a specification from the w3c allowing the integration of a set of independent multimedia objects into a synchronised multimedia presentation, currently in version 1.0	http://www.w3.org/TR/REC-smil/
SMPTE	Society of Motion Picture and Television Engineers	http://www.smpte.org/
SMTP	Simple Mail Transfer Protocol according to RFC 821	

SOAP	Simple Object Access Protocol providing a simple and lightweight mechanism for exchanging structured and typed information between peers in a decentralised, distributed environment using XML proposed under the w3c.	http://www.w3.org/TR/SOAP/
SOAP	Simple Object Access Protocol	
SOAP4J	A JAVA implementation supporting SOAP, developed under the APACHE project.	http://xml.apache.org/soap/index.html
SOJA	SMIL compliant player from HELIO.	http://www.helio.org/products
SQL	Structured Query Language: ISO, ANSI standard user front end to a relational database management system.	
SR/Z39.50	see Z39.50	
SSL	Secure Sockets Layer is a transaction security standard developed by Netscape Communications to enable commercial transactions to take place over the otherwise notoriously non-secure Internet.	
SSL	Secure Socket Layer	
SUTRS	Simple Unstructured Text Record Syntax is part of the Z39.50 standard.	
SWIFT	Society for Worldwide Interbank Financial Telecommunication founded in 1973 to support world-wide financial transactions	http://www.swift.com/
Swing	Software library for building user interfaces, based on AWT.	http://java.sun.com/products/jfc/
Tamino	A XML database product of the Software AG	http://www.softwareag.com/tamino/
Tcl/Tk	Tool Command Language (Tcl) whitch is associated user interface toolkit (Tk) for quickly creating cross-platform applications with graphical user interfaces, developed by John Ousterhout.	http://www.pconline.com/~erc/tcl.htm
Textml	XML database product from ixiasoft.	http://www.ixiasoft.com/
TGN	Getty Thesaurus of Geographic Names.	http://www.getty.edu/research/tools/vocabulary/tgn/
TM4J	The topic map engine for JAVA is a product of Tequila	http://www.techquila.com/tm4j.html
Tmproc	Is an open source Python topic map engine.	http://www.ontopia.net/software/tmproc/

Tomcat	Servlet Engine developed by the Apache-Jakarta project.	http://jakarta.apache.org/tomcat/
TR&P	Transport Routing and Packaging	
UC	Use Case	
UDDI	Universal Description, Discovery and Integration is an industry initiative for creating a platform-independent, open framework for describing services, discovering businesses, and integrating business services using the Internet.	http://www.uddi.org/
UID	Unique Identifier	
ULAN	Union List of Artist Names	
UML	Unified Modelling Language from Rational encompassing the object-oriented analysis and design methodologies of Booch, Rumbaugh, and Jacobson.	
UMM	UN/EDIFACT Modelling Methodology	
UMTS	Universal Mobile Telecommunications System	http://www.umts-forum.org/
UN/EDIFACT	United Nations Rules For Electronic Data Interchange for Administration, Commerce and Transport	http://www.disa.org/international/edif.htm
UNICODE	Character encoding standard which is part of the ISO-10646.	
URI	Uniform Resource Identifier is a generic set of all names/addresses that are short strings that refer to resources. Defined in RFC 2396.	http://www.ietf.org/rfc/rfc2396.txt
URL	Uniform Resource Locator	
VoiceXML	Voice eXtensible Markup Language is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed-initiative conversations. Submitted to the w3c.	http://www.w3.org/TR/voicexml/
WAE	WAP Application Environment	
WAIS profile	Wide Area Information Servers profile based on ANSI/NISO Z39.50 Version 2.	http://ftp.std.com/obi/Standards/TEXT/WAIS-Profile-of-Z3950-V2.html
WAP	Wireless Application Protocol	http://www.wapforum.org/
WML	Wireless Markup Language	http://www.wapforum.org/what/technical.htm
WSDL	Web Services Description Language is an XML format for describing	http://www.w3.org/TR/wsdl

	network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.	
WSP	Wireless Session Protocol	
WTLS	Wireless Transport Layer Security Protocol	
WTSL	Wireless Transport Security Layer	
WYSIWYG	What You See Is What You Get	
X12	EDI standard	
XALAN	XSLT processor of the apache project	http://xml.apache.org/xalan-j/index.html
XER	XML Encoding Rules	http://asf.gils.net/xer/standard.html
XERCES	XML parser of the apache project	http://xml.apache.org/xerces-j/index.html
XM	eXperimentation Model (MPEG- 7)	
XMI	XML Metadata Interchange defined by the OMG for file export/import of models, and a transfer format specification for unique identification of the version of the MOF meta-metamodel and any metamodels referenced but not included in an SMIF-compliant transfer.	
XML	EXtensible Markup Language	
XML/EDI	EDI encoding using XML	
XPath	XML Path Language is a language for addressing parts of an XML document, designed to be used by both XSLT and XPointer.	http://www.w3.org/TR/xpath
XSL	Extensible Stylesheet Language. An XSL stylesheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses the formatting vocabulary.	http://www.w3.org/TR/xsl/
XSLT	XSL Transformations is a language for transforming XML documents into other XML documents.	http://www.w3.org/TR/xslt
XSU	XML Schema Upgrade service	http://www.w3.org/2001/03/webdata/xsu
XSV	XML Schema Version checking service	http://www.w3.org/2000/09/webdata/xsv
XTM	XML Topic Map	http://www.topicmaps.org/xtm/1.0/
Z39.2	American National Standard for Information Interchange	http://www-it.hr.doe.gov/standards/stdsdesc.cfm?Id=

	(= ISO 2907)	11
Z39.50	Information retrieval protocol standard (= ISO 23950)	http://www.niso.org/z3950.html http://www.loc.gov/z3950/agency/
ZAP!	A Z39.50 apache module	http://www.indexdata.dk/zap/

7 Bibliography and References

- Bluetooth: The Bluetooth Specification v. 1.1 core, Bluetooth, 2000.
- ebXML: ebXML Technical Architecture Specification v1.0.4, ebXML, 2001.

http://www.ebxml.org/specs/ebTA.pdf

- ebXML: Business Process Specification Schema v1.01, ebXML, 2001.
- Gamma, Erich et.al.: Design Patterns - Elements of Reusable Object Oriented Software, Addison Wesley, 1995.
- Grady Booch, Jim Rumbaugh, and Ivar Jacobson: Unified Modeling Language—Reference Manual, Addison-Wesley, 1999.
- International Standardisation Organisation: ISO 23950 Information and documentation -- Information retrieval (Z39.50) -- Application service definition and protocol specification, 1998.
- International Standardisation Organisation: ISO/IEC 13250 Topic Maps Information Technology
- Document Description and Processing Languages, International Standardisation Organisation, 1999.
- Ivar Jacobson, Grady Booch, and Jim Rumbaugh, Unified Software Development Process, Addison-Wesley, 1999.
- MPEG Comittee: ISO/IEC WD 15938-1 MPEG-7 Systems Specification, MPEG Comittee, 2001.
- Sun Microsystems: Enterprise JavaBeans(TM) 2.0 Specification Final Release, Sun Microsystems, 2000.
- Sun Microsystems: JAVA(TM) 2 Platform, Enterprise Edition Final Release 1.3, Sun Microsystems, 1999.
- Sun Microsystems: Mobile Information Device Profile (MIDP) Specification 1.0, Sun Microsystems, 2000.
- Sun Microsystems: J2ME(TM) Connected Limited Device Configuration Specification 1.0a, Sun Microsystems, 2000.
- Susanne Boll, Wolfgang Klas, Utz Westermann: Multimedia Document Models – Sealed Fate or Setting Out for New Shores?, Databases and Information Systems (DBIS), 1999.
- Thomas, Anne: Enterprise JavaBeans Technology - Server Component Model for JAVA Platform, Patricia Seybold Group, 1998. http://java.sun.com/products/ejb/pdf/white_paper.pdf
- Topicmaps Consortium: XML Topic Maps (XTM) 1.0, Topicmaps Consortium, 2001.
- UDDI: UDDI Version 2.0 Data Structure Reference, UDDI, 2001.

http://www.uddi.org/pubs/DataStructure-V2.00-Open-20010608.pdf

List of Figures

List of Tables

Table of Appendices