Consolidated Requirements on ontological approaches for integration of multi-level biomedical information

Project Number: FP6-2005-IST-026996

Deliverable id: D 7.1

Deliverable name: Consolidated Requirements on ontological approaches for integration of multi-level biomedical information

Date: 19 January, 2007

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	ACGT
Project Full Name:	Advancing Clinico-Genomic Clinical Trials on Cancer: Open Grid Services for improving Medical Knowledge Discovery
Document id:	D 7.1
Document name:	Consolidated Requirements on ontological approaches for integration of multi-level biomedical information
Document type (PU, INT, RE)	RE
Version:	Final
Date:	19.01.2006
Editor: Organisation: Address:	Luis Martín UPM School of Computer Science, UPM, Madrid, Spain

Document type PU = public, INT = internal, RE = restricted

ABSTRACT:

This deliverable presents an analysis of the requirements needed in Work Package (WP) 7: *Ontologies and Semantic Mediation Tools.* It has been extended to provide a vision of the state of the art in some of the focused issues within WP7. A description of the approaches adopted, at the time of writing this document, has been included, as well as a complete requirements specification of the methods and tools to be investigated and developed during the next phases of the ACGT project.

The partners involved in WP7 have adopted a standard-based methodology for gathering requirements. It is based on specific scenarios provided by both technicians and end users. Such requirements can evolve during the entire project. They can be modified to follow up the evolution of the state of the art in the involved areas.

This document includes the ACGT Master Ontology on Cancer. This important objective for ACGT is described here to facilitate its understanding and use as a core resource for semantic mediation and interoperability.

KEYWORD LIST: Ontologies, Semantic Mediation, Database Integration, Semantic Heterogeneity, Query tools, Mapping, Web Services, Grid Services

MODIFICATION CONTROL			
Version	Date	Status	Editor
0.1	22.11.2006	Draft	Luis Martín
1.0	15.12.2006	Draft	Luis Martín
Final	19.01.2006	Final	Luis Martín

List of Contributors

- Alberto Anguita, UPM
- Guillermo Calle, UPM
- Andrés Silva, UPM
- Victor Maojo, UPM
- Gabriele Weiler, IBMT-FhG
- Mathias Brochhausen, IFOMIS
- Anand Kumar, IFOMIS
- Patrick Durand, INRIA
- Haris Kondylakis, FORTH
- Martin Doerr, FORTH

Contents

Table of Figures

Executive Summary

This deliverable presents an analysis of the requirements needed in Work Package (WP) 7: *Ontologies and Semantic Mediation Tools.* It has been extended to provide a vision of the state of the art in some of the focused issues within WP7. A description of the approaches adopted, at the time of writing this document, has been included, as well as a complete requirements specification of the methods and tools to be investigated and developed during the next phases of the ACGT project.

The partners involved in WP7 have adopted a standard-based methodology for gathering requirements. It is based on specific scenarios provided by both technicians and end users. Such requirements can evolve during the entire project. They can be modified to follow up the evolution of the state of the art in the involved areas.

This document includes the ACGT Master Ontology on Cancer. This important objective for ACGT is described here to facilitate its understanding and use as a core resource for semantic mediation and interoperability.

This deliverable is divided in three parts and one technical annex:

Part 1 contains an introduction of the role of WP7 within the ACGT project, as well as a detailed description of the tools and systems that will be part of the WP7 layer. The first section introduces WP7 and details its motivation within the project. A brief description of its main components is also given.

Next section presents the **Mediator**. This tool is in charge of performing the semantic integration of distributed and heterogeneous data sources. It will offer query services to end users and will provide analytical tools inside ACGT itself. The Mediator relies on the Mapping Process, which facilitates the inclusion of new data sources within the integration platform.

We also present and discuss the **ACGT Master Ontology on Cancer**. It is used in the integration process. This ontology will contribute to collect all the information related to cancer that needs to be handled within ACGT, offering the conceptual basis for a structured knowledge repository.

The Mediator and the end users will be linked by the **Query Interfaces**. These will offer to non-technical users an easy interaction with the Mediator, facilitating to construct complex queries while maintaining simplicity at a reasonable level.

WP7 also includes tools for creating and managing components of the ACGT clinico genomic trials. This feature will facilitate researchers to share a common and intuitive framework, including an easier monitoring of the involved processes, the creation and edition of Case Report Forms (CRFs) and the automation of different tasks.

Part 2 of the document contains the requirement specification of The Mediator and the Query Tools. The IEEE 830-1998 standard has been followed for this purpose.

These requirements do not aim to be definitive, as they will evolve throughout the project life. Many aspects will continue to grow during the development phases, so further revisions of this document are expected.

Part 3 of the document contains a review of the State of the Art regarding the methods and tools described in Part 1. Four sections are included:

a) *Visualization Tools*: several kinds of interfaces for query construction are reviewed here. Their pros and cons are analyzed, taking into account the non-technical background of many prospective users and the expected complexity of the queries they pretend to perform.

b) *Semantic Integration*: Methods are basically divided in two types: data warehousing and query translation. It is suggested that the latter fits better with the different challenges raised within ACGT. A list of different systems is also analysed.

c) *Biomedical Ontologies*: this section contains an extensive review of the ontologies currently available in the biomedical domain.

d) *Top Challenges in relation with Data Management and Ontologies*: the relation between ontologies and KDD, GRID computing, and images is analyzed here, emphasizing how ontologies might contribute to advance R&D in all those areas.

Technical Annex: Use Cases, contains the use cases that have arisen during the analysis of the requirements.

1 Introduction

1.1 WP7 IN THE ACGT PROJECT

ACGT project aims at building a platform that helps biomedical researchers in their investigations against cancer, namely Wilm's tumor and breast cancer. It will offer services that support development of clinical trials, providing analytical tools, tumor growth simulations tools and querying services to heterogeneous databases.

Due to the tremendous growth in number and size of biomedical databases during the last years, its proper semantic integration turns into a critical feature in this project. Semantic mediation will allow end users (clinicians) to increase the productivity of conducted clinical trials, relieving them of the problems that dispersion and disparateness of required data presents. More information will be available in an easier manner, thus resulting in more fruitful experiments. Furthermore, integrating heterogeneous and disparate sources can lead to the discovery of new semantic relationships between different resources. Besides this, analytical tools within the ACGT environment itself will take advantage of the services offered by the mediator too. KDD tools, as well as workflow execution environments will make use of these services within specific workflow steps.

Work Package 7 in ACGT is devoted to the development of a semantic mediation layer within the ACGT environment. Its main goal is achieving the integration of heterogeneous data sources, and offering querying services on them. This is a key feature in the ACGT project, since it will allow clinicians to better perform clinical trials, and will be utilized by other ACGT tools.

The objectives of WP7 are:

- Develop a shared semantic mediation middleware, offering different kinds of semantic services for data access and integration.
- Create a model describing the domain related to SIOP and Breast Cancer clinical trials, building the ACGT Master Ontology on Cancer.
- Exploit the ACGT Master Ontology on Cancer, building ontology-compliant tools and annotating resources to guarantee semantic interoperability.

The core of this layer will be composed by the mediator tool and the Master Ontology. The former will offer the query services mentioned above, while the latter will provide the necessary semantic background during the integration process. Other ontological tools will also be developed. Next sections describe in more detail all layer components, as well as its interface with lower layers.

The mediator is the main tool in the semantic layer, for which WP7 is responsible. Its task will be integrating several data sources so they can be easily queried by users and tools. The mediator will be surrounded by other tools which will add more functionality to this layer. Namely, these are The Mapping Tool, the Unification Tool, The CRF Creator and the GUI Interface. They are all described in subsequent sections. The mediator is responsible for offering querying services against distributed and heterogeneous data sources in a uniform manner. These will include ACGT databases, external databases, web sources and web data services. These services will be accessed by both users and tools, which will see the system as a query interface to a single data source comprising all needed data. The stress has been put on the need of transparency in the integration process, enhancing user-friendly characteristics in the system.

1.1.1 WP7 ARCHITECTURE

In Figure 1 the general architecture of tools to be developed in WP7 can be seen.

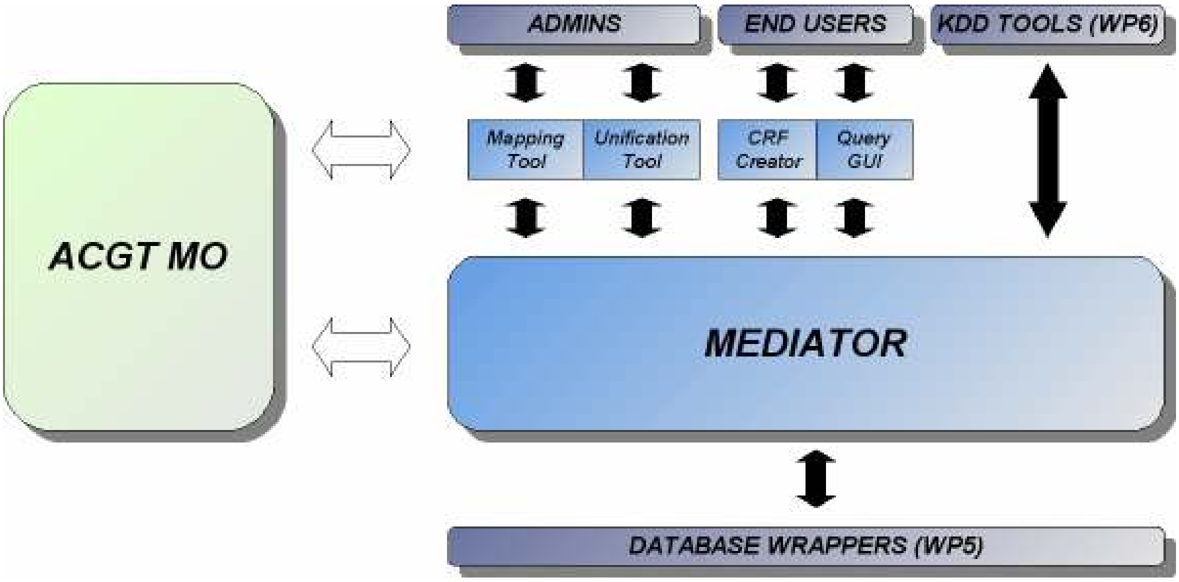


Figure 1: WP7 Architecture

It can be seen that this mediation layer has direct communication with the KDD (Knowledge discovery in Databases) tools (WP6) and the Database Wrappers (WP5). The technical details of these interfaces will be described at the design stage.

Next the reader can find a brief description of each one of these tools.

1.1.1.1 The Master Ontology on Cancer

The mediator will be supported by the Master Ontology on Cancer (MO). It will comprise all ontological needs of the ACGT clinical scenarios, and offer the necessary framework for the existing terminologies and ontologies in the biomedical domain. The queries asked by users in the mediator will be created in terms of the MO, guaranteeing semantic interoperability among different data sources employed in the integration process and hiding specific structural and semantic details of such sources.

The MO will allow the mediator performing all necessary tasks in the integration process. It will act as resource in the construction of the required structures and models, providing a common framework of information and notation.

1.1.1.2 Ontology-Based Tools

Ontology driven software development refers to the use of ontologies as “building blocks” during software development. Its main motivation is achieving semantic interoperability between disparate applications. Users will manage such systems in a standardized manner, leading to easier reuse and sharing of data.

WP7 will include the development of several ontological tools, such as the CRF Creator, the Mapping Tool and the Unification Tool. The CRF Creator will allow clinicians managing and editing Clinical Report Forms (CRFs) during clinical trials. As mentioned above, this will ease their work and facilitate interoperability between different institutions. Both Mapping and Unification tools will be intended for system administration. They will facilitate administrators labour in the inclusion of new data sources to the mediator for their subsequent integration.

1.1.1.3 Query GUI

The Query GUI will act as a layer that allows non-technical users to successfully interact with the mediator. Given the area of expertise of expected users, the interface with the mediator cannot be too complicated (for example, an SQL-like interface is not viable). On the other hand, users expect to perform quite complicated queries. This implies that the interface must be powerful enough to allow such cases. This is the reason why a dedicated query GUI is included in the design. It will be in charge of catching users input, and transforming it into queries understandable by the mediator. As a result, this tool will be placed between the users and the mediator.

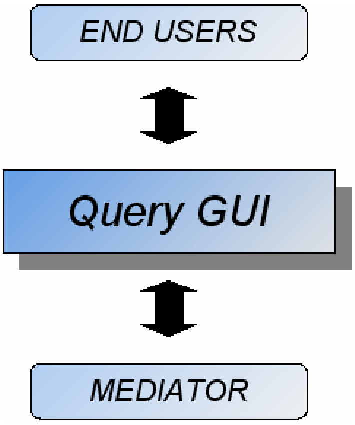


Figure 2: Query GUI

There exist several kinds of query interfaces: command line interfaces, web interfaces and GUI interfaces (the one chosen for the mediation layer). Next section describes them in detail.

1.1.1.4 Mapping tool

The Mapping tool will support system administrators in the mapping process. The results of such process are virtual schemas that represent the data source schemas in terms of the Master Ontology. This is a necessary step prior semantic integration of the data. Therefore every new data source to be included in the integration environment must go through this process.

The mapping process consists on defining associations between terms and relations from the data source schema with terms and relations from the Master Ontology. In many cases, understanding the context will be the key to find such relations. This implies that this tool is not fully automatic, and just serves as a support to administrators who define the relations. This support consists on offering administrators a visual environment in which explore both the ontology and the source schema, and build the virtual schema.

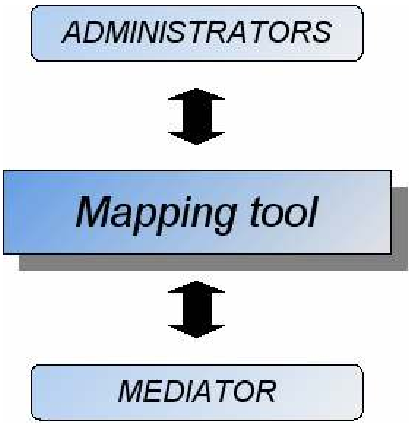


Figure 3: Mapping Tool

1.1.1.5 Unification tool (View Integrator)

The unification tool, also called “view integrator”, is in charge of allowing users to define different integration profiles based on the needs of a specific trial.

Although virtual schemas are based on elements taken from the Master Ontology, this model is too big and complex to be used as a global schema for all data sources. In order to constraint the domain, restricted views of integration will be built.

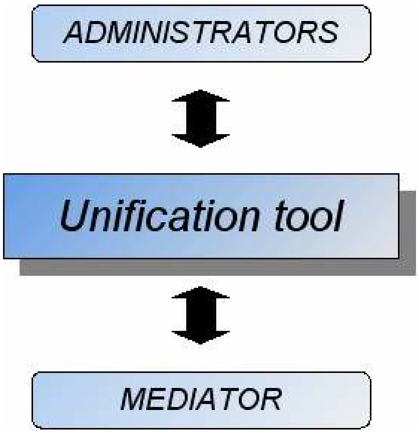


Figure 4: Unification Tool (View Integrator)

1.1.1.6 The Mediator

In recent years, the number of biomedical databases has shown a large increase in both number and size. Similarly, the number of different, remote locations where these databases are located is also increasing. Furthermore, this kind of world-wide scientific environment, where all the data needed by a specific research can be distributed among different settings introduces an informatics issue to be resolved: heterogeneity. Different databases tend to store data using different platforms and software, often incompatible formats, or requiring specific access systems or languages, making data gathering a highly time-consuming task.

Semantic mediation addresses this problem. Its final goal is to offer a seamless integration of distributed heterogeneous databases, allowing end users to take full advantage of the data of several databases, while offering a simple and homogeneous access system to all those resources. This process can be completely or partially assisted by informatics methods and tools.

ACGT proposes to investigate and build a “Mediator” to hide the complexity of query translation and data integration. While following the state of the art in the area, it proposes an innovative approach to semantic mediation. In this approach, the user performs queries against a single, “virtual” repository. This virtual repository represents the integration of several heterogeneous sources of information. This integration process relies on a common interoperability infrastructure, based on a conceptual level on a domain ontology.

In the ACGT scenario, the Mediator has to deal with two major challenges:

- Schema level inconsistencies: referred to differences in the schemas of the external databases to be integrated. The same concept can be represented by different names or the same name can represent different concepts in different databases. Thus, in order to integrate the different database contents, this kind of heterogeneity must be eliminated.

- Instance level inconsistencies: refer to inconsistencies in the actual data stored in the databases. For example, different identifiers can be used to express the same instance, or some measurement can differ in the units that are used. This kind of heterogeneity requires a unification process, so the result of a query shows a unique result format.

This classification is made from a technical perspective. Types of heterogeneity can be organized from a pure theoretical perspective, although we will follow a pragmatic approach to the field.

1.1.1.6.1 Selected Approach

We have adopted a data-driven approach. The mediator code will be completely independent from the databases to be integrated. This approach will allow that, if a new database is included in the system, the mediator does not need to be updated (only the associated data). All data regarding data management will comply with the ACGT Master Ontology.

The basic role of the Mediator within the ACGT environment is to provide ACGT users with a powerful tool for retrieving data from integrated database systems (originally distributed and heterogeneous). An adequate query interface will be provided, along with a useful visualization model for results of queries.

The mediator will be accessible as a service (based on GRID, Web or APIs), and will be able to accept queries and retrieve results from different integrated databases. This type of access allows an easy integration with other systems (e.g. KDD systems), as well as the development of tools requiring the Mediator as a resource (e.g. query tools for integrated databases).

The design and development of the Mediator must be focused on its integration within the ACGT platform. Given that focus, some issues must be taken into account. In the ACGT architecture, the mediation layer is located between Knowledge Discovery Tools (WP6) and Database Wrappers (WP5). The Mediator acts as a service for Knowledge Discovery Tools, and as a client for Database Wrappers.

In ACGT, we apply a LAV (Local as View) approach to schema mediation. In this approach there preexists a global schema. Local schemata to be integrated are mapped to the global schema so that local schema elements are completely expressed in terms of the global schema.

This requires a global schema powerful enough to cover the semantics in the local schemata. The approach works well if the local semantics are predictable, and as long as amendments to the global schema can be made in an upwards-compatible way.

The advantage of this approach is the tight integration and guaranteed powerful capabilities of reasoning on the integrated data, in particular joins across data from all different sources.

In ACGT, the global schema will be a subset of the ACGT Master Ontology.
Following the experience of the partners in charge, the known and expected semantics of ACGT applications can be covered to the degree described above to support an LAV approach. The domain is known enough and well treated by ontology engineering, so that core concepts can be standardized for all intended uses of ACGT, and possible evolution in terminology does not affect the level of conceptualization present in schema entities to be mediated. This relieves from the adventure to integrate schemata stepwise, detecting underlying semantics during the integration process or not. It makes the mapping process accessible to domain experts, given suitable user-friendly tools. It allows distributing the mapping process without affecting global properties of the system.

In more details, each source will be wrapped exposing to the mediator the local schema in terms of concepts of the master ontology, i.e. as a view of it. In other terms, the local source appears under a virtual schema by virtue of the wrapper, which is a view of the Master Ontology. The wrapper will be based on mapping specifications described by a Mapping Tool as detailed in section 1.3. It basically resolves heterogeneity, in particular cases requiring joint processing of data elements.

The View Integrator ("Unification Tool") will combine the virtual schemata employed and complement them to form the Virtual Global Schema in use. The latter forms again a subset of the Master Ontology. It is a union of all concepts in the local views, enriched by default generalizations and relations between them, such as all subsumptions declared between its classes in the Master Ontology, but not necessarily present in any of the views. Following experience, it is assumed that the cases of heterogeneity described in 1.3 are restricted to within an individual source. Therefore view integration does not require any more complex mechanisms.

The ontology based mediation mechanism is restricted to discrete notions. It does not apply to numerical data, which need processing by respective mathematical problem solvers. The ontology based mediation will transport numerical data retrieved by queries against the global schema to another level of integration for numerical processing. It makes sense to do ontology-based mediation first, because it decides if numerical data are related at all.

Therefore, in addition to concepts in the Master Ontology, the actual virtual schemata may need enrichment by some fixed, generic information system elements to package numerical data and handle them through the ontology-based mediation layer. In the ontology, universal concepts represented as schema entities or as terms in data records are not distinguished. The mediator may make this distinction internally for practical reasons and add respective generic information system elements to the global schema. The user interface may or may not make such distinctions, depending on user requirements.

Besides concepts of the Master Ontology, the Global Schema should contain only generic information system elements, supposedly only those just mentioned.

1.1.1.6.2 Mediation services

The Mediator offers a query API for KDD tools and Workflow management. Mediation services will be accessible for the development of integrated database querying tools as well.

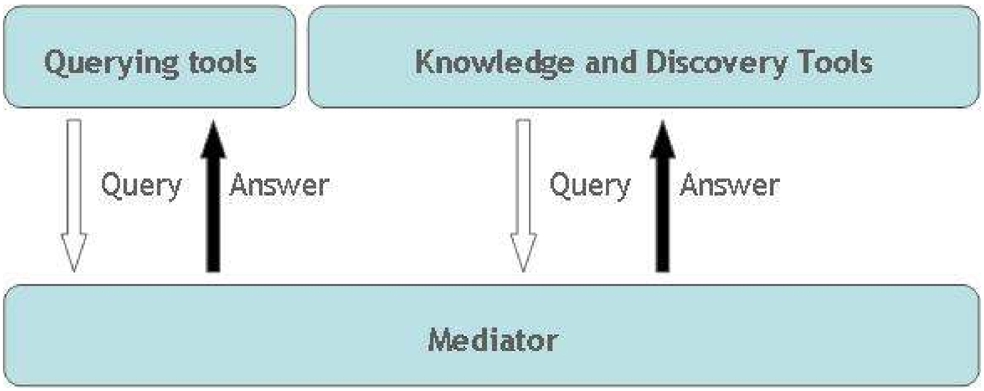


Figure 5: The Mediator as a service provider

The KDD tools to be developed in WP6 will require services from the Mediator. An agreement is necessary in two issues: which services will be needed and the API specifications for such services (how these services will be requested, and how results will be returned). However, the definition of these interface details will be tackled at the design phase. Preliminary discussions within the involved partners led to the idea of wrapping the Mediator through Web-Services, and returning the results via XML format. Offering the Mediator services through Web-Services will allow an easier integration with other tools, since its interface would rely on well defined standards and would not tie client tools to any specific language or platform. A Web Service can be wrapped with a GRID service if it is necessary.

Query tools that make use of the mediator will be exploited by end users (clinicians). They will expect a simple yet powerful query system (since they are not experts in query languages). There are already ongoing discussions on the specifications of the query language to be used, a decision that will be adopted during the next months, corresponding to a forthcoming deliverable.

1.1.1.6.3 Services required from lower layers

As shown in figure 2, The Mediator layer is located on top of the WP5 layer. This layer will provide a seamless and interoperable data access services to any kind of database to be included in the ACGT platform, by implementing wrappers for each of such databases. These wrappers must hide the complexity and details of every database, offering a common access interface.

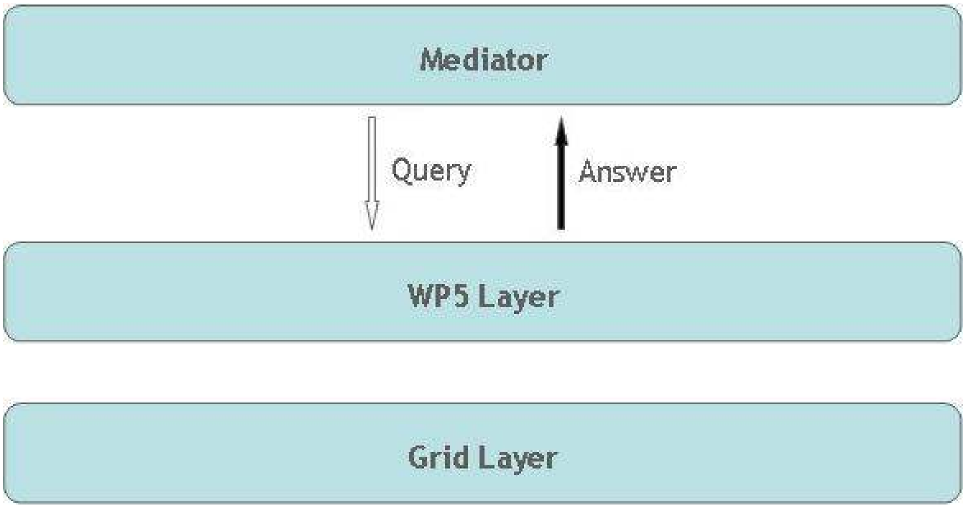


Figure 6: The Mediator as a client

WP5 wrappers will allow access to heterogeneous sources of information. This access will include private and public (accessible via Web) databases. The specific interface details to allow communication with WP5 layer will be discussed within the forthcoming design phase of WP7 —and related WPs.

1.1.1.6.4 Interface to lower layers

Database Wrappers will act as a boundary between WP7 and WP5, allowing the mediator to easily query the physical data sources to be integrated. They will offer a seamless and uniform access interface, thus simplifying the mediator design. Services will be provided to support data acquisition from heterogeneous sources, regardless their nature or structure.

1.2 THE MAPPING PROCESS

The integration of heterogeneous sources of information must be approached using informatics methods and tools that can guarantee semantic interoperability. The ACGT Master Ontology is a key feature for such purpose. The mapping process aims to build interoperable schemas for all the databases that are going to be integrated in the system.

The mapping process is essential in heterogeneous database integration. This mapping process and the development of wrappers —also required to provide homogeneous access to databases—, are usually, together, the bottleneck of the whole integration procedures. Mapping different data sources needs to be assisted by experts in both IT and the specific application domain. This process requires of the analysis of semantic heterogeneities in data, a time consuming task.

1.2.1 WHAT IS THE MEANING OF "MAPPING" TWO SCHEMATA?

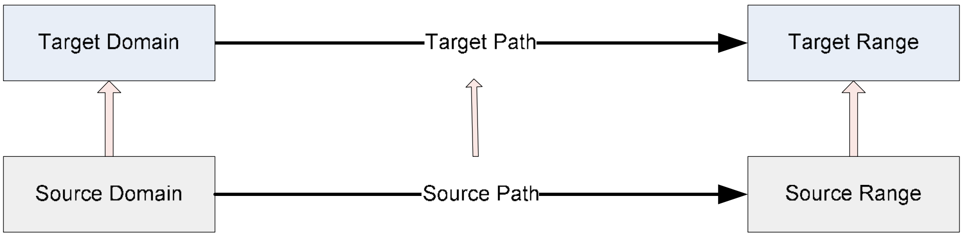


Figure 7: The basic mapping schema

The definition of mapping two schemata is "a transformation of each instance of schema 1 into an instance of schema 2 with the same meaning". The definition should be independent of particular instances. Mapping should implement an automatic transformation algorithm for all instances of schema 1 into instances of schema 2, only following the definition of the transformation.

We suppose that the mapping definitions are produced manually —or semi-automatically, using some specific software mapping tool— by a domain expert, possibly assisted by an IT expert. A whole ontology —or a subset— can be used or interpreted as a "target schema" in the process.

To carry out an efficient mapping procedure we need to define:

a) The mapping between the Source Domain and the Target Domain.
b) The mapping between the Source Range and the Target Range.
c) The proper Source Path.
d) The proper Target Path.
e) The mapping between Source Path and Target Path

To the best of our knowledge, no other mapping language or mechanism provides all those definitions, and most of them assume that some of those are implicitly defined. Moreover they may combine Paths with Ranges when trying to define mapping rules. We argue that all the previous definitions should be explicitly defined in order to have efficient mapping rules.

1.2.2 CASES OF HETEROGENEITY

The process of creating the mapping rules is not a straightforward process since multiple conditions and cases may exist. The problem of defining mappings between arbitrary schemata can not be solved in all cases. Moreover, instances of schemata may not follow the intended meaning of the source schema, leading to exceptions. However, in a given domain, the cases of heterogeneity are normally quite limited. In order to overcome those problems we suggest defining a mapping mechanism that will cover the most common cases. In the rest of cases, the mechanism should be extended.

The mapping mechanism should be intuitive enough to facilitate understanding and use by the domain expert. To achieve this goal we examine carefully the most common cases of heterogeneity between the Source and Target Schema in our application context.

Our model can be used for XML and Relational Databases. Moreover this model can be used in primitive object oriented databases, too. For relational databases, we consider their schema compared to a semantic model. In order to do that we consider:

- Tables, columns as entities
- Records as entity instances
- Fieldnames as relationships and entities
- Field contents as entity instances.

Each field is interpreted as a relationship class-role-class (c-r-c),and the whole schema is decomposed in c-r-c's. Then, each c-r-c is mapped individually to the target schema.

The reader can find later a catalog of cases of heterogeneity, with examples explaining those cases.

1.2.2.1 Case 1. Introducing an intermediate node.

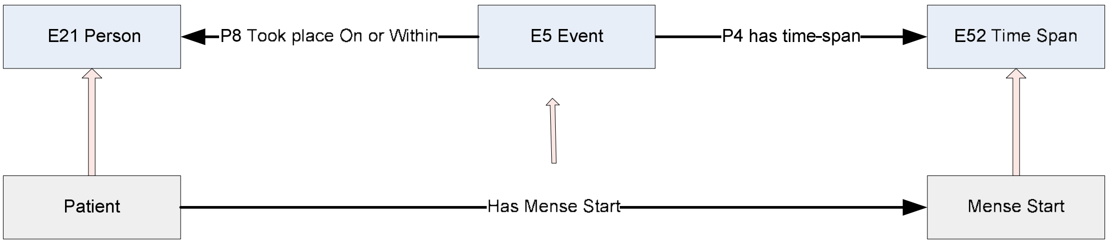


Figure 8: Introducing an intermediate node

In this case, an intermediate node should be introduced in order to define precisely the whole path that needs to be mapped to the source path. This is because in the source model, it is common to compact a large path into a single relation (usually events) when no information needs to be stored in the intermediate nodes. Those intermediate nodes are necessary when other schemata have information that relates the intermediate node. This is implemented in the proposed mapping format by the "internal_link" and "internal_path" tags than can exist within a “target_path” tag that provides the capability to have many intermediate link and paths.

1.2.2.2 Case 2. Compound Contraction

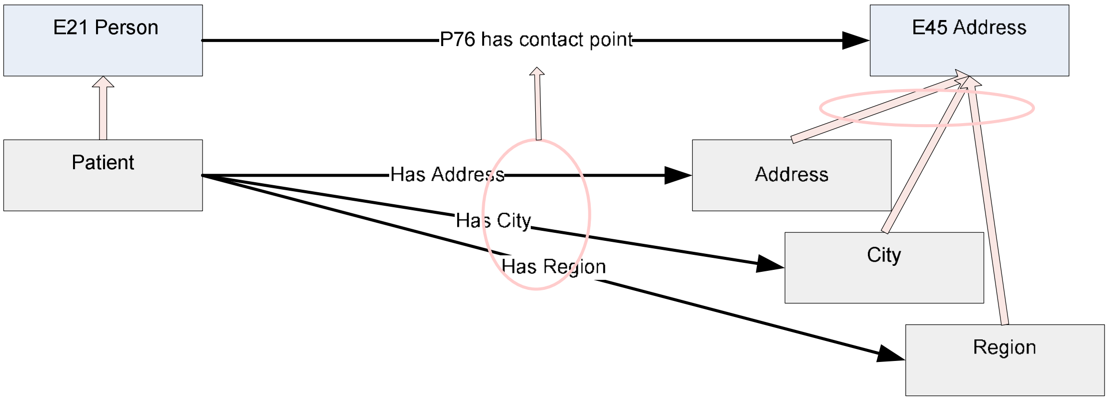


Figure 9: Compound Contraction

The Compound Contraction is frequently observed in information such as addresses, special names, coordinates, and others. In this case, several classes in the source schema are parts of one identifier for one actual thing, one class in our target schema. There must be a way to declare that all those classes are parts of the same class in out target schema (Gruber 1993). This is implemented in the proposed mapping format by the attribute “compound_on” of the “combined_links” tag.

1.2.2.3 Case 3. Parallel to Nested

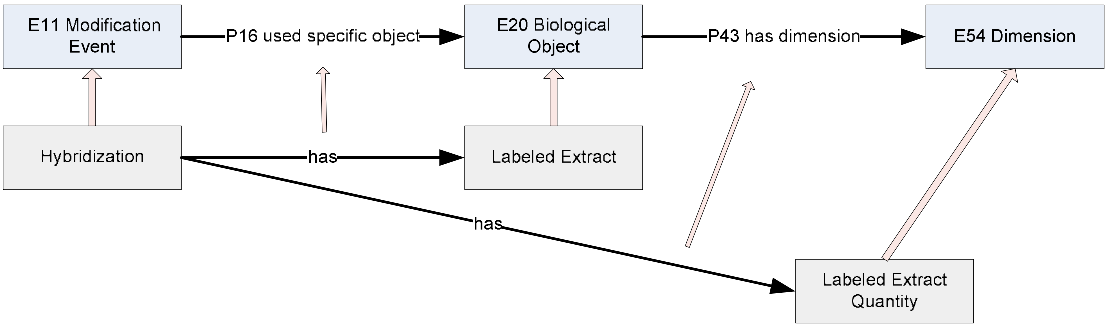


Figure 10: Parallel to nested

There are cases, in the source schema, where a class A is related to other classes Bi, and those relations imply causal connection between some of the related classes Bi —rather than between A and Bi, as stated in the schema. This denormalized form is typically idiosyncratic to one source schema. Therefore those connections must be made explicit in order to insure interoperability with information from other sources. For example as we can see in figure 6, within the source model. “Hybridization” is related to “Labeled Extract Quantity”. However “Labeled Extract Quantity” is a class that should be related to a “Labeled Extract” since in reality it is its attribute.

1.2.2.4 Case 4. Parallel to Intermediate Parallel

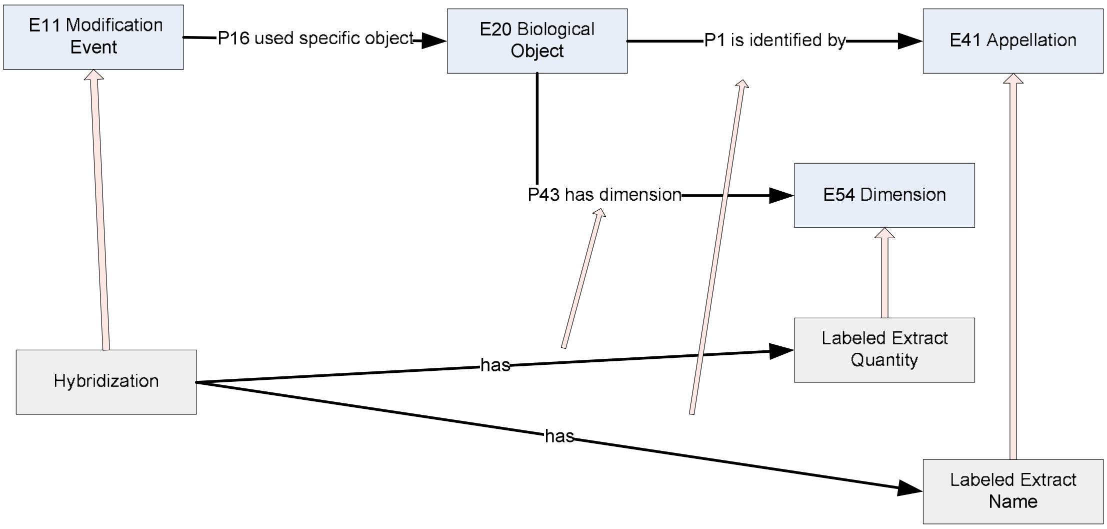


Figure 11: Parallel to Intermediate Parallel

This case only extends the previous one: Two relations in the source schema mapped to a path in the target schema with the same intermediate node. Note that not only the class of the intermediate node is the same but also the instance by which the intermediate node has to be instantiated is the same for the mapping of both paths.

1.2.2.5 Case 5. Same instance participates in multiple mappings

Therefore, we need a general mechanism to define that the same instance of a class in a mapping rule appears in multiple mappings. This is implemented in the proposed mapping format by the attribute “joined_on” of the “combined_links” tag and by using several “link_maps” within the “combined_links” tag.

1.2.2.6 Case 6. Conditions

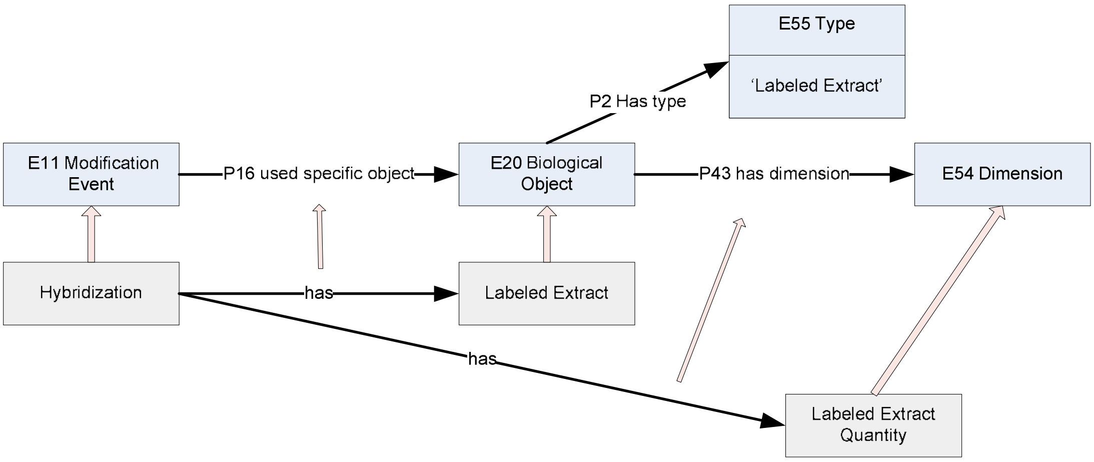


Figure 12: Biological object should have type 'Labeled Extract'

Finally we need a mechanism to define the following simple conditions for the mapping:

a) An attribute of an instance is equal to a term or constant

e.g if instance.Attribute=" … "

b) An attribute of an instance is a term "a" subsumed by another term "b"

e.g if instance.Attribute $\subset$ term b

c) An instance of the attribute does exist or not.

Those most common conditions cases described here, are confirmed from the MIDAS (a manual and data standard in monument inventories developed by FISH) mapping effort using CIDOC CRM. This is implemented in the proposed mapping format by the tags "src_path_condition", "target_path_condition", "src_domain_condition", "target_range_condition" and the "value_binding" attribute of the "internal_link" tag.

1.3 THE ACGT MASTER ONTOLOGY ON CANCER

1.3.1 AIM AND SCOPE OF THE ACGT MASTER ONTOLOGY

The ACGT consortium seeks to provide complex data querying and mediation functionality for the ACGT Grid infrastructure. Building an ontology appears to be the best conduct in this respect, in order to supply the foundations for semantic data integration.

One of the definitions of ontology most often cited states that an ontology is a formal, explicit specification of a shared conceptualization [STU1998]. As regards, an ontology-based approach will not prevent the existence of a multitude of non-interoperable ontologies. Yet, it is important to ensure that the problems arising from the existence of multiple ontologies do not outnumber the considerable advantages of using an ontology in the first place. This issue is addressed by the ACGT project and solved by the creation of the ACGT Master Ontology. Any project of such magnitude deals with enormous amounts of data coming from different sources; these data are being queried by different users in different countries, with different linguistic and scientific backgrounds. The advantage of using ontologies should, hence, be obvious: adopting a common ontological framework avoids the use of proprietary and idiosyncratic terms and terminologies, and thus fosters interoperability and uniform resolution. Terminological differentiation and the growing number of terminologies for handling data is one of the most pressing problems. The so called “tower of Babel” problem is an obstacle for progress in many scientific disciplines, and in the biomedical science in particular. ACGT aims at resolving this problem for the domain of cancer research and management.

The ACGT Master Ontology is meant to constitute a reference ontology for the field targeted by the ACGT project, and has as an immediate objective the enabling of semantic data integration across its various sections. One of the underlying principles being used in building this ontology is the assumption of a robust realistic perspective; one notable consequence is that its nodes purport to represent universals or classes, as opposed to mere “concepts” as it is customary in Computer Science. Building the ontology will also involve heavy recourse to logico-philosophical principles. While aiming to create a common reference that is both human and machine understandable, we will have to integrate state of the art knowledge from the medical domain as provided by the clinical and biomedical institution in the ACGT consortium.

A reference ontology for cancer research and management will inevitably contain entities and universals from a wide range of topics, from the genetic and medical field to the administrative field (e.g. participation in a study) or the legal domain (e.g. consent). This leads to certain challenges for the development of the ACGT Master Ontology which will be discussed in more detail below.

It is crucial that the aim of this ontology is data integration. It follows that the ACGT Master Ontology is dealing with content, not with services.

1.3.2 THE TOP LEVEL ONTOLOGY

Even before the ACGT project started, IFOMIS —one of the ACGT partners participating in WP7— was active in developing ontological solutions for the cancer domain. These efforts resulted in an ontology of colon carcinoma [KUM2005]. During this process, a reference ontology was developed which integrated domain ontologies from anatomy, physiology and pathology. This reference ontology is called the Ontology of Biomedical Reality (OBR) [ROS2005]. It proves that integration is not only needed among medicine and different sciences, but it has to be achieved intradisciplinary —in medicine itself. However useful this system might be, it cannot unfortunately constitute the basis for the ACGT Master Ontology, since the top-level entity of OBR is "biological entity". Extensive parts of data to be integrated within the ACGT environment do not deal with biological but, as mentioned above, with administrative or legal entities. Thus, the range of the ACGT project presents one of the challenges to the Master Ontology, since a purely biological ontology cannot suffice to solve the problems at hand.

Deciding the structure of the topmost level of the ontology to be built is the most important step in ontological research. Basic Formal Ontology (BFO) [BFO] has been adopted as the top level for the ACGT Master Ontology, due to its superior manner in which it categorizes reality; BFO is based on the following four theoretical principles:

- **Realism** - reality exists independently of our representations;
- **Fallibilism** - scientific theories and science-based ontologies can be subject to revision;
- **Perpectivalism** - there is a plurality of equally legitimate perspectives on reality;
- **Adequatism** - no reduction of the different perspectives.

A central feature of BFO is a basic dichotomy between continuants and occurrents (the SNAP-SPAN dichotomy [GRE2004]), which emphasizes two distinct modes of existence in time. Furthermore, BFO exists now in an OWL-DL implementation which increases the possibility of syntactical integration and reasoning. Any systematization of the world (or of any given domain) has to start with basic ideas on what entities exist, or what are the criteria to use in order to categorize the elements of reality at a basic level. In this process, questions about the essence of things have to be tackled. This Top-Down move as part of ontology development is vitally important in order to reach common terms and principles. As we have seen, the major distinction in BFO is based on how entities are related to time.

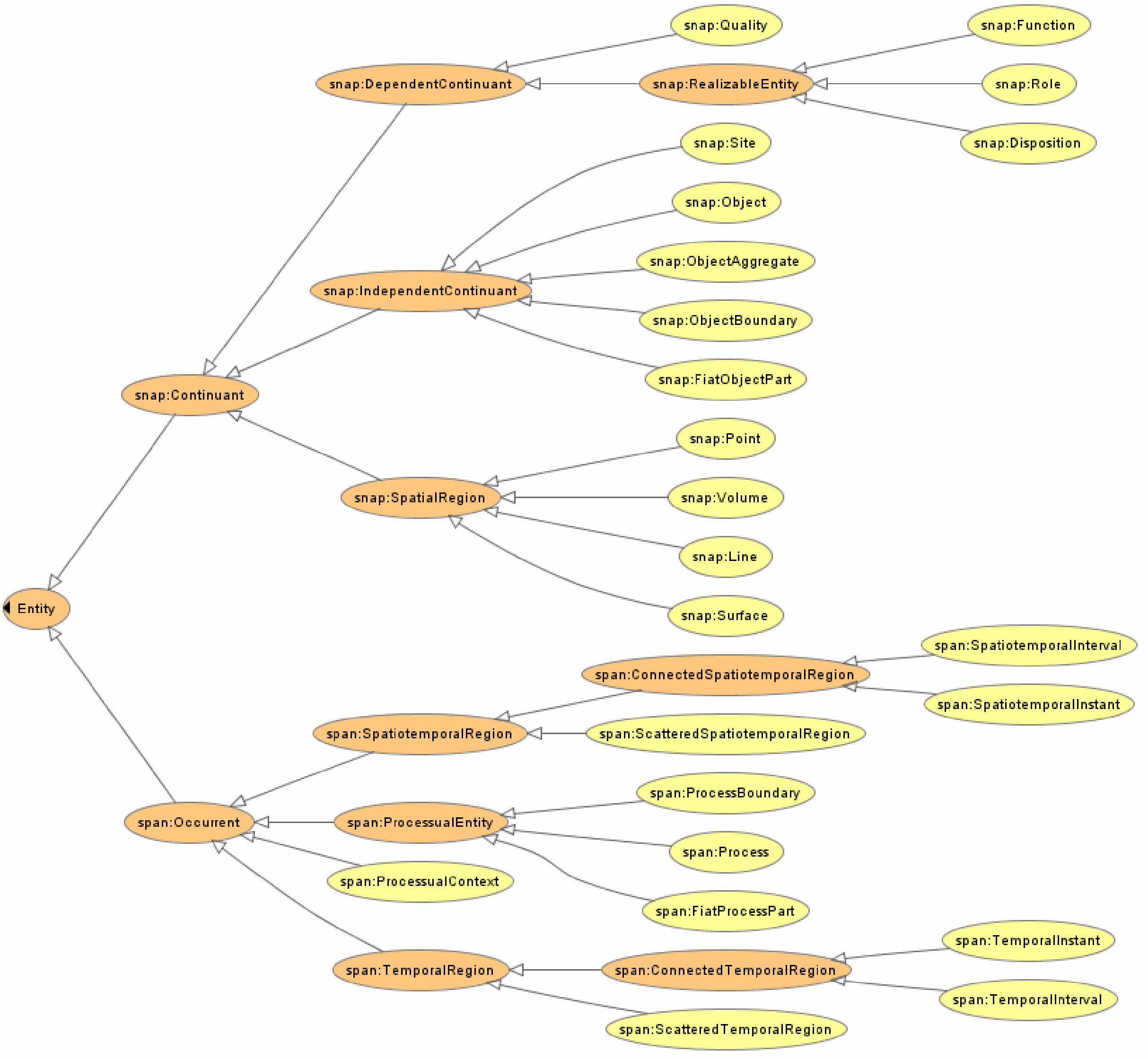


Figure 13: The Basic Formal Ontology (BFO)

The existence of a coherent top level ensures the reusability of the ontology, since it prevents the development of ontologies based on a top level which is restricted to one specific domain.

1.3.3 CLINICAL PRACTICE AND THE ACGT MASTER ONTOLOGY

The first step in adding entities from clinical practice in the ontology was to integrate clinical report forms (CRF) into the system. CRFs contain data from different data types in the ACGT domain, with the exception of molecular data. In typical Bottom Up fashion, we edited universals to which patient data refers. Within the ACGT framework it makes perfect sense to start with CRFs, since we might achieve a situation where data integration sets in at the very moment data is produced.

The principles used to design the topmost level of the ontology are obviously to be observed further in developing lower levels.

1.3.4 QUALITY STANDARDS

An important issue that has to be addressed by any project that relies on a reference ontology is the problem of quality management regarding the ontology. It is our goal to make the ACGT Master Ontology a member of the Open Biomedical Ontologies (OBO) Foundry [OBO]. This is a library of interoperable reference ontologies for the biomedical sector, which subscribe to the same quality standards. All ontologies in the foundry are open source. The OBO Foundry is one way to prevent the “tower of Babel” challenge mentioned above. Since FMA and GO also count themselves among the members of the OBO Foundry, we will be aiming at integrating these systems into our ontology—to the extent, of course, to which the target domains intersect. If we are successful in becoming a member of the OBO Foundry, the ACGT Master Ontology will be among the most extended ontologies in the biomedical domain. By developing the ontology in cooperation with other members, we will keep it updated and growing consistently for a long time to come. Thus, ACGT contributes to the global efforts to build ontology-based health care systems and data integration for biomedicine.

1.3.5 FUTURE CHALLENGES

However promising, this work process leads to another problem: If we start to integrate data by making use of the Master Ontology, we will come to a point at which the clinician has to deal with the ontology or, at least, ontology-based applications. Ontology is by definition not based on practical or clinical perceptions of reality. Entities which seem to be closely related from the clinical point of view, might be essentially different from the ontological point of view. E.g. “Neoplasm” is a continuant, whereas “TumorStage” is an occurrent. The clinician’s view is necessarily focused on the health of the patient and the different approaches directed at restoring it. Sound ontological reasoning cannot focus on workflows in clinical practice. The clinician’s manner of dealing with patient’s situation is roughly governed by epistemological and practical considerations. This situation leads to another challenge to ACGT: How can the ontology be visualized in applications which are easily manageable by the clinician in reasonable time? This is extremely difficult since medicine itself deals with different points of view, e.g. the clinical view on disease classification versus the pathological systematization of diseases.

The task described in the last paragraph cannot be solved solely via ontology development, since a realistic ontology should never take epistemological considerations into account. As soon as questions such as “How does a clinician perceive this?” are part of ontological reasoning, the positive effects of realism will be lost. This will lead to yet another conceptualist vocabulary. Other R&D activities in the ACGT project will address this problems in close communication with clinicians and ontologists.

1.3.6 REFERENCES

[BFO] http://www.ifomis.uni-saarland.de/bfo/home.php

[GRE2004] P. Grenon, B. Smith, and L. Goldberg, “Biodynamic Ontology: Applying BFO in the Biomedical Domain,” in: Ontologies in Medicine, D. M. Pisanelli, Ed., Amsterdam: IOS Press, 2004, pp. 20-38.

[KUM2005] A. Kumar, Y. L. Yip, B. Smith, D. Marwede, and D. Novotny, “An Ontology for Carcinoma Classification for Clinical Bioinformatics”, in proceedings of MIE2005, Geneva, 2005, pp. 635-640.

[OBO] The OBO Foundry: http://www.obofoundry.org/

[ROS2005] C. Rosse, A. Kumar, J. L. V. Mejino Jr., D. L. Cook, L. Detwiler, and B. Smith, “A Strategy for Improving and Integrating Biomedical Ontologies,” in proceedings of the AMIA Symposium 2005, Washington DC, 2005, pp. 639-643.

[STU1998] R. Studer, V.R. Benjamins, and D. Fensel, “Knowledge Engineering: Principles and Methods”, IEEE Transactions on Data & Knowledge Engineering, vol. 25 no. 1-2, pp. 161-197, 1998.

1.4 QUERY INTERFACES

1.4.1 QUERY INTERFACES FOR THE MEDIATION LAYER

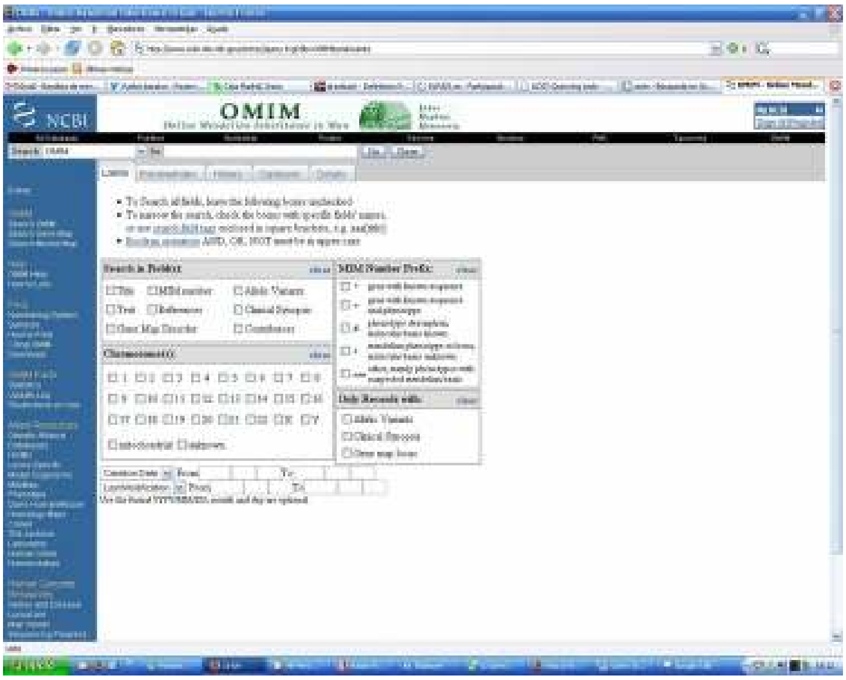
GUI Tools will allow users to interact with the Mediator. They will be responsible for accepting user queries in a simple enough form for non-technical users to handle.

The most important decision regarding Query Interfacing was choosing the most appropriate type of interface. There are several options available, listed next:

Command line interfaces: queries are constructed through text statements. It offers a high flexibility, but lacks in simplicity. An example is SQL language.

Web-based access: these include web pages containing forms the user must fill in order to construct the desired queries. This kind of interface is much simpler than the previous one; however it does not allow performing complex queries, since their structure is highly predetermined. Next we can see an example, taken from the public database OMIM[OMIM]:

-



- To Search all fields, leave the following boxes unchecked.
- To narrow the search, check the boxes with specific fields' names, or use search field tags enclosed in square brackets, e.g. aaa[MIM]
- Boolean operators AND, OR, NOT must be in upper case

Figure 14: OMIM interface example

Interactive GUI access: This third kind of interface relies on visual query languages (VQLs) to offer user an intuitive way of submitting queries. Simplicity is combined with the ability to perform complex queries. The following image shows an example of this, taken from the GenoLink[DUR2006] tool:

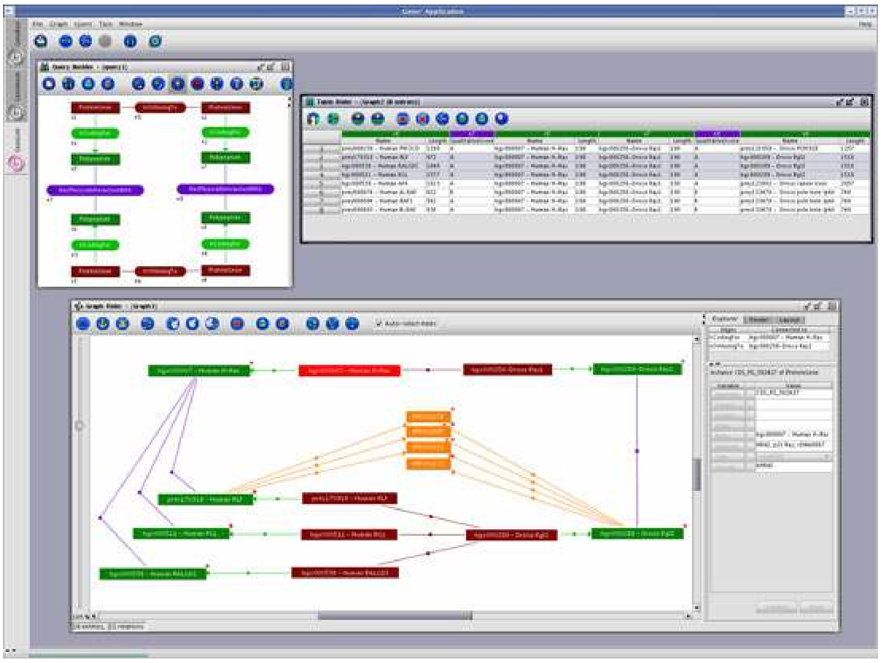


Figure 15: GenoLink interface example

Interactive GUI access will be the kind of interface that will implement the GUI Tools.

1.4.2 REFERENCES

[DUR2006] Durand P, Labarre L, Meil A, Divol JL, Vandenbrouck Y, Viari A, Wojcik J (2006) GenoLink: a graph-based querying and browsing system for investigating the function of genes and proteins. BMC Bioinformatics. 7(1):21

[OMIM] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM

1.5 ONTOLOGY DRIVEN APPLICATIONS. ONTOLOGY DRIVEN SERVICES FOR THE CREATION AND MANAGEMENT OF CLINICO GENOMIC TRIALS

1.5.1 INTRODUCTION

The need for services for the creation and management of clinico genomic trials has been motivated in Deliverable 5.1 [D5.1]. There has been described that these services have to allow domain experts (clinicians) to define requirements for a clinical trial and the definition of all kind of research and administrative data that has to be collected during the trial in a standardized, user-friendly way. Furthermore it has to be possible to deploy data management services automatically from the defined requirements for the conduction of the trial in the ACGT environment.

Please refer to Deliverable 5.1 for further requirements and a general description of such services [D5.1]. In this chapter we want to focus on the advantages by basing these services on the ACGT master ontology to overcome the problem of semantic interoperability.

The biggest problem that has to be faced during developing these services is to assure that the data collected during a clinical trial is annotated with comprehensive metadata. The problem in current clinical trials is that even clinical biostatisticians are only rarely able to make good use of data collected on studies they were not directly involved with, mostly, due to incomplete or non-existent annotation and standardization of the collected data.

This fact will also make the semantic integration of legacy trial databases into ACGT error prone and time intensive despite the best mapping tools provided.

One requirement for the data management services is that the data collected during a trial can be automatically integrated into the mediator architecture of ACGT. That implies that the trial databases have to be master ontology – compliant.

That means that during development of the data management services the ACGT master ontology has to be integrated. The trial data definition as well as the metadata for the trial databases has to be based on this ontology.

Information Integration with all other kind of data sources provided by ACGT over the mediator architecture can then be done semi or automatically.

The approach of basing the development of software applications on reusable reference ontologies has already been described in the literature as a ‘development paradigm for its own’ called ‘ontology driven software development’ [KNU2005]. A lot of different names can be found for this idea in literature, like ontology driven architecture or ontology driven information systems. Nevertheless we will refer below to this approach as “ontology-driven software development”. The main aim of this approach is achieving semantic interoperability between the developed systems. While research on this field is still in its infancy one of the goals of ACGT is to contribute to the field with some practical advances.

To understand the problems related to building software driven by ontologies, we want to clarify what we understand by ontology driven software development and give a review of what can be found in the literature associated to this topic.

1.5.2 ONTOLOGY DRIVEN SOFTWARE DEVELOPMENT

Current domain models for software applications are mostly crafted from scratch during application development. This process is time intensive and leads to semantically interoperable domain models —and therefore into interoperable applications.

The vision of ontology driven software development, in general, is that reusable reference ontologies can be used as 'building blocks' during the software development process. Software applications are then built based on these blocks or from smaller pieces. All applications would then have overlapping domain models based on the reference ontology and therefore would have a certain degree of built-in features to facilitate semantic interoperability.

Furthermore, ontologies can be used as 'mechanisms' for rigorously describing, identifying, discovering and sharing software artifacts. Taking into account the old computing adage that "all the software functionality needed in the world has already been written somewhere" we can take a new approach to build software. If all this functionality were made openly available and semantically described by ontologies, software development should be based in discovering and gluing together existing functionality with pieces of interoperable reference ontologies. This would result in semantically interoperable software applications that are easier to build, understand and maintain over time. [KNU2006][TET2006][GUA1998]. Such approach can even lead to a future where an end user without informatics skills can build by herself new applications in a standardized way. This perspective will require user-friendly development frameworks.

However this is a vision for the future, parts of this vision have already been successfully demonstrated and tools and techniques been developed to contribute to this idea.

Already in 1997 Guarino described a scenario for the future that the semantic content expressed by an ontology (or ontologies) selected from an ontology library gets transformed and translated into an information system component, reducing the costs of conceptual analysis and assuring – on the assumption of a correct ontology – the ontological adequacy of the Information system. This view should enable developers a higher level of reuse than it is actually the case in software engineering (i.e. knowledge reuse instead of software reuse). However he stated, that the availability of "off the shelf" ontologies to be used in this way was —and it is, actually— extremely limited. The reason is that the available ontologies are not generic enough to be effectively specialized for various applications. [GUA]

This situation raises the question whether it is theoretically possible to provide such ontologies. This problem is related to the 'interaction problem' that was discussed at the end of the 90s in the AI community. The interaction problem states that when an ontology is developed to solve a particular problem it is always dependent on the problem and therefore it is very difficult to reuse the ontology for other applications. [BYL1998]

This problem seems to be solved today for most researchers. The AI-community has demonstrated successfully that knowledge-based systems can be build effectively from reusable domain ontologies. The latter provide a characterization of the concepts and relationships in an application area, and problem solving methods that offer abstract algorithms for achieving solutions to stereotypical tasks. It has been proposed that a similar method is useful not only for knowledge-based systems but for software applications in general. [MUS2002]

The latest research initiative contributing to ontology driven software development and the most interesting for our work is the Semantic Web. The vision behind the Semantic Web is to make web-content machine-understandable so that it can be analysed by software agents and shared among Web Services. To reach this goal ontologies are used to formalize web-content for describing machine understandable metadata.

In the context of the Semantic Web Best Practices it is stated that in an ideal world, developers would discover sharable ontologies from a variety of interrelated repositories and then wire them together with the remaining object oriented components —a concept slowly becoming recognised as ontology driven architecture.

There is little or no guidance on how to build ontology driven applications. Ontology driven software development is still more a vision than a development paradigm. It lacks methodologies that can guide software developers about how exactly the ontology can be integrated into the software development process most effectively and how to have ontologies interact with the rest of the application architecture. Some promising approaches are beginning to appear [KNU2004][TET2006].

Initiatives linked to the idea of the Semantic Web initiative have contributed with a lot of tools that can help in integrating ontologies into software applications.

First of all, the World Wide Web Consortium (W3C) recommends OWL (Web Ontology Language) and RDF(S) (Resource Description Framework (Schema)) as standard ontology languages. OWL builds on top of RDF(S) and is a combination of three increasingly expressive sublanguages. One of them called OWL DL is based on Description Logics. OWL DL provides maximum expressiveness, but also guarantees that all conclusions are computable and will finish in a finite time.

These ontology languages are today widely used for ontology development. Although these languages can not solve the problem of semantic interoperability between ontologies they can at least contribute to syntactically interoperable ontologies.

Around these ontology languages the World Wide Web Consortium (W3C) is also recommending a set of tools for developing, maintaining, using and sharing ontologies.

For our work we have chosen to use Jena [JEN] and the Protégé OWL API [PRO]. Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment to integrate ontologies into software applications. It contains classes and methods to load and save OWL files, to query and manipulate OWL data models, and to perform reasoning. The Protégé-OWL API is an open-source Java library for the Web Ontology Language and RDF(S). This API is optimized for the implementation of graphical user interfaces. Both APIs are open source and compliant to each other. Therefore, they are well suited to be used in the described services.

Another initiative currently contributing to ontology driven development is the ODM (Object Definition Metamodel) initiative of the OMG that wants to bring ontology development closer to software developers aiming to integrate ontologies into the context of their software development paradigm Model Driven Architecture (MDA). The aims are to augment the model languages of the MDA with features of ontologies, encourage the various groups of OMG to adopt appropriate ontology technologies and establish relationships among OMG activities and ontology related activities, such as the Semantic Web. [ODM][GAS2006]

However promising the approach of ontology based software development seems there are still a lot of challenges to face. For instance, although the 'interaction problem' has been solved, there are still missing methodologies to build reusable ontologies. Perhaps realistic ontologies that are build referring to the real world —and not to specific problems, tasks or applications— can address this problem. But standards for developing those ontologies are currently not available. Developed ontologies are semantically often interoperable due to homonyms, synonyms or different partitions used.

However the OBO Foundry seems to have the potential to solve these problems for the biomedical domain. The aim of this open organization is to develop interoperable reference ontologies for the biomedical domain based on the realistic approach. The ACGT master ontology will be part of the OBO Foundry. [OBO]

An even more challenging problem is to give guidance in how exactly an ontology has to be integrated in a software application. Tools that integrate the ontology in the development process of software have to be provided.

1.5.3 INTEGRATION OF ONTOLOGY INTO THE SERVICES FOR CREATION AND MANAGEMENT OF CLINICAL TRIALS

We want to achieve that a clinician can set up required data management services that are needed to collect the data for a particular clinical trial in a standardized way.

During this process it has to be possible to design standardized user interfaces, called Case Report Forms (CRFs), which can be used to collect the defined data during the trial.

Therefore it has to be possible that the clinician builds the domain model for the clinical trial database from the ACGT master ontology. That is not possible by representing the ontology to the clinician in an ontology editor and assuming he is able to build a database from that ontology. The clinician has to be led in this process by user friendly tools.

Although an ontology is 'human understandable' by providing natural language definitions of entities and relationships it is by definition not based on practical or clinical perceptions of reality. The ACGT master ontology will moreover be based on description logics and therefore very hard to understand by clinical users. Therefore, a tool —an 'ontology visualizer' —is required to provide an application specific view on the ontology in a way that a clinician can understand it. A clinician aiming to design a trial will naturally want to focus on the user interfaces and try to integrate and adapt them into the workflow of the specific clinical trial she wants to perform. For this task a CRF creator is needed.

The 'ontology visualizer' has to be integrated into the process of building CRFs in a way that the ontology will guide the clinician to only design sensible user interfaces. Furthermore this tool has to assure that for all the data on the CRFs that will be collected during running the trial, comprehensive metadata is chosen from the ontology.

In many trials similar or equal data are collected. So it is highly desirable to store CRFs or parts of them, once specified, in a repository within the ACGT environment for their reuse in later trials. The ACGT master ontology can here be used in the role of a 'mechanism' for rigorously describing, identifying, discovering and sharing software artefacts in this case CRFs or parts of them that have already been designed and annotated with metadata from the ontology. In that way the required CRFs can be easily found by semantic search based on the ontology and integrated in the CRF creator software.

From the definitions done by the clinicians with the help of the 'CRF creator' clinical data management services that are based on the ACGT master ontology can then be deployed automatically. The clinical trial database will have comprehensive metadata in terms of the ontology. The data collected in the trial can then be integrated automatically into the ACGT mediator architecture since it has comprehensive metadata in terms of the ACGT master ontology.

Through this approach the vision of machine understandable metadata and automatic, intelligent processing of the data may become true in the future.

1.5.4 CONCLUSION

The described approach of ontology driven software development seems to be a very promising approach to solve the problem of semantic interoperability in the future but faces a lot of challenges. The approach seems to be well suited to develop services for the creation and management of clinico-genomic trials. So it is highly relevant for the project to investigate this approach further. Since the main focus of WP7 lies on semantic integration of heterogeneous data sources, the consortium will investigate how to use the resources of the project beyond the preliminary prototypes that will made in the next phases of ACGT.

1.5.5 REFERENCES

[BYL1998] T. Bylander and B. Chandrasekaran, Generic Tasks for Knowledge-Based Reasoning: The "Right" Level of Abstraction for Knowledge Acquisition. In: B.R. Gaines and J.H. Boose (eds), Knowledge Acquisition for Knowledge-Based Systems. Academic Press, London, 1988, pp. 65-77.

[D5.1] Deliverable 5.1. of the ACGT project.

[GAS2006] D. Gasevic, D. Djuric, V. Devedzic (2006). Model Driven Architecture and Ontology Development. First Edition, Springer, Berlin, Heidelberg.

[GUA1998] N. Guarino, Formal ontology and information systems. In Nicola Guarino, editor, Formal Ontology and Information Systems, (FOIS'98). IOS Press, 1998.

[JEN] Jena – A Semantic Web Framework for Java; http://jena.sourceforge.net; (last accessed: 27.07.2006).

[KNU2004] H. Knublauch. Ontology-Driven Software Development in the Context of the Semantic Web: An Example Scenario with Protégé/OWL. International Workshop on the Model-Driven Semantic Web, Monterey, 2004.

[KNU2005] H. Knublauch. Ramblings on Agile Methodologies and Ontology-Driven Software Development. 4th International Semantic Web Conference, Galway, 2005.

[KNU2006] H. Knublauch, D. Oberle, P. Tetlow, E. Wallace. A Semantic Web Primer for Object-Oriented Software Developers, W3C Working Group Note (work in progress), 2006; http://www.w3.org/TR/sw-oosd-primer (last accessed: 06.11.2006).

[MUS2002] M.A. Musen, Ontology-oriented design and programming. In: J. Cuena, Y. Demazeau, A. Garcia, and J. Treur, eds. Knowledge Engineering and Agent Technology. Amsterdam: IOS Press, 2002.

[OBO] The OBO Foundry; http://obofoundry.org; last accessed 06.11.2006).

[ODM] Ontology Definition Metamodel; http://www.omg.org/ontology; (last accessed 06.11.2006).

[ODM] Ontology Definition Metamodel; http://www.omg.org/ontology; (last accessed 06.11.2006).

[PRO] Protégé-OWL API; http://protege.stanford.edu/plugins/owl/api; (last accessed 06.11.2006).

[TET2006] P. Tetlow, J. Z. Pan, D. Oberle, E. Wallace, M. Uschold, E. Kendall. Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering, Editors Draft (work in progress), 2006; http://www.w3.org/2001/sw/BestPractices/SE/ODA (last accessed: 06.11.2006).

2 Mediator and Query tools requirements specification

2.1 INTRODUCTION

This document contains the Software Requirement Specification (SRS) of the Mediator layer, part of the ACGT project.

2.1.1 PURPOSE

The purpose of this document is to have a consistent and unambiguous specification of the characteristics of the software to be developed. This document will also serve as a guide in the design of the tools to be implemented. It aims also to become the meeting point between the engineers in charge of the development and the end users for which the system will provide services.

This document is addressed to:

- Tool developers: will act as a guide in the design process.
- End users: will reflect their needs.

It must be also pointed out that this document can act like an agreement between developers and users since it specifies what the system will and will not do.

2.1.2 SCOPE

The software products that will be produced are:

- The Mediator,
- Query GUI Interface for The Mediator,
- The CRF Creator.
- Mapping Tool
- Unification Tool

These software products will provide a common way to gather the data from heterogeneous databases (The Mediator, Mapping and Unification tools), with a specific interface to the user (Query GUI Interface), and a tool that will allow designing CRFs for a clinical trial in a standardized way (The CRF Creator).

The final goal for the Mediator is to offer a seamless integration of distributed heterogeneous databases, allowing end users to take full advantage of data from several databases, while offering a simple access system to all those resources. This will dramatically reduce the time and effort researchers usually waste on data gathering during their work.

2.2 METHODOLOGY

A proper methodology for software development becomes critical in ACGT. The platform to be developed includes several software packages which comprise numerous features. There are a great number of partners participating, from several different countries, and many kinds of users from different areas of research and expertise. Embracing software engineering practices is therefore necessary to assure the success of a project of such magnitude.

Standard requirement specification techniques are adopted within WP7. They are described in D2.1 "User requirements and specification of the ACGT internal clinical trial". These include the use of scenarios and prototypes, as well as an iterative requirement elicitation and specification process. Discussions with clients of WP7 layer have already taken place. This practice will remain in the feature, allowing requirements to evolve according to users needs.

2.3 OVERVIEW

The rest of the document contains a detailed description of the product to be developed, specifying features, requirements and constraints related to it. Section 3.2 contains the description of the product, including the expected user characteristics and constraints that may be found. Section 3.3 contains specific requirements of the product, described by means of features and functional requirements descriptions. These requirements descriptions serve as definitions for system use case diagrams, included at the end of the document.

2.4 OVERALL DESCRIPTION

In this section general requirements of the product will be described. It is not meant to be a detailed analysis, but rather a background for more detailed descriptions.

2.4.1 PRODUCT PERSPECTIVE

The Mediator is part of the ACGT project, and thus it must communicate with other ACGT subsystems. Besides being a tool for end users, the Mediator will offer services to different analysis tools (KDD tools, Workflow executors…). Interface specifications and formats will be discussed and agreed with the developers of these tools. The Mediator also acts as a client at a lower layer, responsible for offering a seamless access to different kinds of data sources. Again, the interface and formats of the services will be properly agreed.

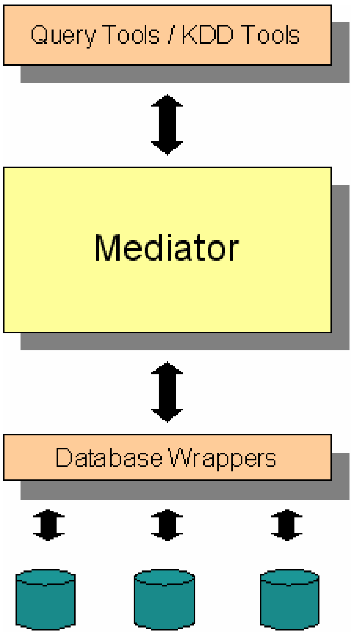


Figure 16: Mediator relation with adjacent layers

2.4.2 PRODUCT FEATURES

The system will offer the following features:

1. Query Integrated Databases
2. Query Reuse
3. Management of Trial Projects
4. CRF Management
5. CRF Editing
6. Creation of Virtual Schemas
7. Unification of Virtual Schemas

System features will be described in section 2.4.1 and in the Technical Annex.

2.4.3 USER CHARACTERISTICS

The end users of the system will be clinicians and researchers in the biomedical area. It cannot be therefore assumed they possess any technical knowledge regarding database query languages. This puts an important constraint in the design of the system. Developing a query interface tool must be easy for inexperienced users. This is the reason why it is planned to develop a natural language query interface tool. Furthermore, end users will require performing quite complex queries, which implies more difficult constraints to such query tool.

2.4.4 CONSTRAINTS

The system goal is the integration of distributed and heterogeneous databases. This implies the access to data sources across the Internet, which might result in delayed response times. It is necessary to reduce the time the users must wait for results as much as possible. Otherwise they might feel uncomfortable with the system, and refuse to use it. Therefore an emphasis should be put on optimizing this aspect.

Clinical data to bqe integrated by the system will include patient data regarding their privacy. It might even allow identify participants of clinical trials. Both legal and ethical principles meet here, and it is of vital importance to preserve the anonymity of the affected people. Therefore a pseudonymization procedure is required.

As it was described previously, the system will sit on a middle layer within the ACGT platform. Proper communication with upper layer (analysis and KDD tools) and lower layer (database wrappers) must be guaranteed.

2.5 USE OF THE MEDIATOR IN THE ACGT SCENARIOS

A series of scenarios are described in D2.1 "User requirements and specification of the ACGT internal clinical trial". Workflows are specified for each one, and several steps in such workflows require the use of the Mediator. Following sections detail which are these steps and why the Mediator is needed in each of them.

2.5.1 SCENARIO SC1: A COMPLEX QUERY SCENARIO FOR THE TOP TRIAL

Steps 1, 2 and 5 explicitly require Mediation services. Reutilization of results of queries in subsequent ones is also needed, therefore support for storing results must be provided.

2.5.2 SCENARIO SC2: IDENTIFICATION OF NEPHROBLASTOMA ANTIGENS

Retrieving of data will be performed from Web databases in step 1, thus the Mediator will be used. These results will be subsequently filtered.

2.5.3 SCENARIO SC3: CORRELATING PHENOTYPICAL AND GENOTYPICAL PROFILES

Step 2 explicitly requires query services for data integration. The data retrieved will be patients' clinico-hispathology and gene-expression data. These results must be stored.

2.5.4 SCENARIO SC5: IN-SILICO MODELLING OF TUMOR RESPONSE TO THERAPY

Clinical and imaging data must be retrieved, and used as input for the OncoSimulator. These data is therefore implicitly obtained through the Mediator tool.

2.5.5 SCENARIO SC6: MOLECULAR APOCRINE BREAST CANCER

Step 2 includes retrieving of data from the ACGT database for its posterior loading into the environment. This data is implicitly accessed through the Mediator.

2.5.6 SCENARIO SC7: VAN'T VEER STUDY

Step 1 includes the loading of sample data into the environment. These data is retrieved from data sources, therefore the Mediator must be accessed. Step 3 again involves the retrieving of data, thus the Mediator tool is again invoked.

2.5.7 SCENARIO SC8: ANTIGEN CHARACTERISATION SCENARIO

In goal 1, step 2, user must collect information from different sources. This will be done through the Mediator. Goal 2, step 2 involves exploring literature related to given identified diseases.

2.6 SPECIFIC REQUIREMENTS

2.6.1 SYSTEM FEATURES

2.6.1.1 Query Integrated Databases

2.6.1.1.1 Introduction/purpose of feature

A unified virtual schema represents the integration of several real data sources. This unified virtual schema can be queried using an appropriate query language the same way as an actual database. This feature allows the user to perform such queries.

2.6.1.1.2 Associated functional requirements

2.6.1.1.2.1 User Log-in

The action of logging into the system is a need of security. Each user will have a set of database repositories associated to his account, so he will be able to access only databases he is allowed to access.

The process of logging is simple: the system requests a user name and a password. If they are right, the user gains access to the system, and is allowed to submit queries.

2.6.1.1.2.2 Submit Query

The action of summiting a query can be performed by different type of users. Both KDD and query tools, as well as final users will be able to submit a query into the system, and they will expect to be retrieved with appropriate results.

The query to be submitted must be expressed in a proper language, and has to be completely compliant with the ACGT Master Ontology. The goal of the query is retrieving results from a virtual schema representing the integration of one or more databases loaded in the system.

From the system's point of view, only a string representing the query is needed. However, the user must be logged into the system in order to be able to access the corresponding virtual schemas.

The results retrieved by the system will be formed by the corresponding rows and metadata associated. All this information will be ontology compliant as well.

2.6.1.2 Query Reusing

2.6.1.2.1 Introduction/purpose of feature

Within a single clinical trial, or among different ones, it may be needed to reuse specific queries stored in a repository. This feature includes the functionalities associated to the reusing of queries.

2.6.1.2.2 Associated functional requirements

2.6.1.2.2.1 Store Query

Queries that have been already used can be stored for future reusing. When the user requests the system to store a query, it keeps a copy of such query in a local repository.

A different repository of queries will be associated to each user with an account on the system. A repository of queries can only be accessed from the account environment of the user associated to it.

2.6.1.2.2.2 Load Query

Queries stored in the repository can be recovered for reusing. In this action, the user request the system to recover a single query previously stored in the respository.

2.6.1.3 Management of Trial Projects

2.6.1.3.1 Introduction/purpose of feature

A clinical trial requires specific software and data resources. The management of such resources can be guided by a tool that considers clinical trials as projects. This feature includes functionalities associated to the management of such clinical trial projects.

2.6.1.3.2 Associated functional requirements

2.6.1.3.2.1 Create new Trial Project

In this action, a software project for creating a clinical trial will be created (called in the following cases, “trial project”), filename and directory for storing the project can be selected.

After creating a new trial project the user interface for describing the metadata for the trial is shown.

2.6.1.3.2.2 Save Trial Project

The trial project that is currently under development will be saved for later editing. Along with the trial project the metadata for the project and the corresponding CRFs are saved.

2.6.1.3.2.3 Open Trial Project

A trial project from a local repository can be selected and will be opened for further editing.

2.6.1.3.2.4 Describe Trial Project with Metadata

User interfaces to describe the metadata for the trial are shown where the following data can be entered:

- Trial name
- Abbreviation of trial
- Responsible persons

Inclusion - and exclusion criteria to recruit patients into the trial; specification for which patient has to be filled in which CRF.

The metadata is saved in terms of the ontology. Relations to other data (e.g. the data collected on the CRFs) are automatically generated from the software.

2.6.1.3.2.5 Show List of CRFs

List of all CRFs with their name and description which are part of the current trial project is shown.

2.6.1.3.2.6 Create Header/Footer for all CRFs of the Trial Project

Templates for the header (Text that is shown on top of the CRF) and the footer (text that is shown on the bottom of the CRF) of all CRFs of the trial can be designed. Static text for all

CRFs can be specified as well as CRF dependent parts (e.g. name of the CRF) that can then be specified for every CRF.

2.6.1.3.2.7 Select Template for Layout for all CRFs of the Trial Project

A template for the layout (specifying colour, font size…) of all CRFs for the clinical trial project can be chosen.

2.6.1.3.2.8 Validate Trial Project

It will be checked if the current trial project is valid and complete. Only when a trial project is valid and complete Data Management Services can be created to conduct the clinical trial.

2.6.1.3.2.9 Set up Clinical Data Management Services

When the trial project is valid and complete, Clinical Data Management Services that allow collecting data during conducting the trial can be deployed automatically into the ACGT environment. For that purpose additionally a general framework will be developed that provides roles and rights management, security functionalities and patient management for the clinical data management services using the basic functionalities of the ACGT Grid environment

2.6.1.4 CRF Management

2.6.1.4.1 Introduction/purpose of feature

Creating a Case Report Form can be aided by a software tool that guides the clinician in the design of this document. Templates can be used to give the user the initial framework. These templates can be created and stored in a well organized repository, and can be annotated and classified by means of an ontology. This feature includes functionalities to perform this CRF Management.

2.6.1.4.2 Associated functional requirements

2.6.1.4.2.1 Create empty CRF

An empty CRF template will be added to the current trial project.

2.6.1.4.2.2 Save CRF Locally

Current CRF can be saved in a local directory to use it as a template in other trial projects.

2.6.1.4.2.3 Delete CRF

Current CRF is deleted from the repository.

2.6.1.4.2.4 Design Header/Footer for CRF

The user designs generic header and footer for all CRFs in the trial project.

2.6.1.4.2.5 Select CRF Template from Ontology Based CRF Repository

A User interface is shown where user can conduct a semantic search for a CRF template. Search criteria can be entered in terms of the ACGT master ontology. CRFs which satisfy the query are shown to the user; user can select one of the templates or conduct a new search. The selected template will be added to the current trial project. In all alternatives a name and a description for the CRF can be entered. The currently added CRF template is shown to the user for editing.

2.6.1.4.2.6 Save CRF in Local Repository

CRFs or parts of CRFs (Items or Item groups) can be saved in an ontology driven CRF repository. Only valid CRFs or CRF parts can be saved.

2.6.1.4.2.7 Select CRF Template from Local Directory

User can select a CRF template from a local directory. The selected template will be added to the current trial project.

2.6.1.4.2.8 Select CRF Template Ontology based Repository

User can select a CRF template from an Ontology based repository using a search tool. The selected template will be added to the current trial project.

2.6.1.5 CRF EDITING

2.6.1.5.1 Introduction/purpose of feature

Case Report Forms can be designed with the help of a software tool that uses an ontology to guarantee consistency among terms. This feature includes functionalities to aid in the CRFs design process.

2.6.1.5.2 Associated functional requirements

2.6.1.5.2.1 Create new Itemgroup

Itemgroups (group of questions on a CRF) can be attached to a CRF. An ItemGroup can be selected from the CRF repository or an empty Itemgroup can be attached.

For every Itemgroup the following properties can be entered:

- Designation
- Labelling

2.6.1.5.2.2 Modify Itemgroup

The user modifies the properties of the itemgroup.

2.6.1.5.2.3 Delete Itemgroup

To delete this itemgroup.

2.6.1.5.2.4 Create new Item without ontology support

For every item the following properties can be specified in a dialog box:

- Question that will be shown on the CRF.
- Data type of the answer: e.g. String, Integer,..
- Constraints (optional): see section Constraints
- Possible values of answer (optional)
- Measurement unit of answer (optional): e.g. kg
- If item is optional

2.6.1.5.2.5 Create new Item with ontology support

For creating a new item, a path from the ACGT master ontology can be selected that describes the item semantically. For this purpose the ‘Ontology visualizer’ will be started. That means that a Window is opened that provides a view on the ontology, in a way that paths for describing one or more items can be easily selected from the ontology. A view on the ontology that is understandable by the clinician suggesting him possible items for the CRFs has to be provided.

First realization possibility: Patient is depicted in the center. Range of the relations and attributes of patients are shown around the patient. Ranges can be further specified (subclasses are shown) or ranges of the relations and attributes of this classes can be shown. The path from patient to an attribute can be directly selected to describe an item on the CRF.

As many properties as possible needed to create an item have to be filled in automatically according to the selected path.

2.6.1.5.2.6 Modify Item

User modifies item's properties.

2.6.1.5.2.7 Delete Item

The user deletes an item from an itemgroup.

2.6.1.5.2.8 Define Constraints for single Item

It can be specified, that the value of an item has to be less or greater than or equal to a constant value or that it has to satisfy a regular expression. An error message can be specified that is shown if another value is filled in during conducting the trial.

2.6.1.5.2.9 Define Constraints across Items

It can be specified that when an item/some items has/have a particular value/particular combination of values, another item/itemgroup/error message will be shown.

2.6.1.6 Creation of virtual schemas (Mapping)

2.6.1.6.1 Introduction/purpose of feature

A Virtual Schema represents the structure of the information contained in a database in an ontology-compliant form. The creation of a virtual schema for every single database is a prerequisite for the integration process. This feature includes functionalities to manage the creation of such virtual schemas.

2.6.1.6.2 Associated functional requirements

2.6.1.6.2.1 Create new Virtual Schema

The user builds a virtual schema for one database, based on the mapping of elements of it into elements from the domain ontology.

2.6.1.6.2.2 Open Virtual Schema

The user opens an existing virtual schema, that is maybe not complete, to modify it.

2.6.1.6.2.3 Modify Virtual Schema

The user modifies an existing virtual schema, editing or deleting one or more mapping relations, or creating new ones.

2.6.1.6.2.4 Save Virtual Schema

The user saves the virtual schema he is currently working with. All unsaved changes are stored in the disk.

2.6.1.6.2.5 Load Database Schema

The user loads an existing database schema into the working environment. It will be used to create the virtual schema.

2.6.1.6.2.6 Load Domain Ontology

The user loads an existing database schema into the working environment. It will be used to create the virtual schema.

2.6.1.6.2.7 Map Element

The user creates a new mapping relation, which relates one element of the current database schema with one element of the current virtual schema.

2.6.1.6.2.8 Map Attribute

The user creates a new mapping relation, which relates one attribute of the current database schema with one attribute of the current virtual schema.

2.6.1.6.2.9 Map Relation

The user creates a new mapping relation, which relates one relation of the current database schema with one relation of the current virtual schema.

2.6.1.6.2.10 Add Class to Virtual Schema

The user asks the system to add a new class into the current virtual schema.

2.6.1.6.2.11 Add New Relation to Virtual Schema

The user asks the system to add a new class into the current virtual schema.

2.6.1.7 Unification of virtual schemas

2.6.1.7.1 Introduction/purpose of feature

A Unified Virtual Schema represents the unification of a set of possibly heterogeneous data sources. A Unified Virtual Schema can be queried the same way as an actual database schema. The creation of a Unified Virtual Schema is based on virtual schemas representing data sources to be integrated. This feature includes functionalities associated with the unification of virtual schemas.

2.6.1.7.2 Associated functional requirements

2.6.1.7.2.1 Create New Unification

The user asks the system to create a new unification, which will be empty at the beginning.

2.6.1.7.2.2 Include Virtual Schema in Unification

A virtual schema is included in the unification, from a list.

2.6.1.7.2.3 Request Unification

The user tells the system to perform the unification process with the current working unification.

2.6.1.7.2.4 Unify

The user asks the system to include a new virtual schema into de current unification, and perform the unification process when it is done.

2.6.1.7.2.5 Delete Unification

The user asks the system to erase an existing unification.

2.6.1.7.2.6 Save Unification

The user asks the system to save the current unification.

2.6.1.7.2.7 Load Unification

The user asks the system to load an existing unification.

3 State of the Art

3.1 VISUALIZATION TOOLS

3.1.1 VISUAL INTERFACES TO QUERY DATA MODEL AND DATA

3.1.1.1 Context

As specified in document ACGT_D2.1, the ACGT project will handle various kinds of data types and data. Data types are intended to be described by the ACGT master ontology. This ontology and real data pieces (i.e., data that conform to the ACGT ontology) will be stored somehow on distributed data servers accessible through the ACGT grid infrastructure.

For the point of view of the end users, which are supposed not being specialized in database technologies, ontology and data have to be *accessible* in a *friendly* way. Accessible means here that users will *query* ACGT data bases to retrieve data of interest. Friendly means that the ACGT project should provide easy-to-use interfaces that will hide the complexity of both ontology and real data, as well as will provide a transparent access to the ACGT technical platform.

The following paragraphs will summarize the actual state of the art of systems providing end-users with an access to databases. Since this field of computer science has a long time story, our purpose is not to give an exhaustive list of existing systems. Hence, this paper solely presents the major approaches propose to the end-users to query databases and visualize the query results. These approaches will be illustrated below by examples from existing software tools. The idea is to expose the features that could be relevant to design and implement a VQL-based system for ACGT.

3.1.1.2 End-user access to databases: a state of the art

3.1.1.2.1 Command-line access

Current database systems, whether they are relational, object oriented or XML based, always provide a text-based query language. Such a language provides the user with a *command-line* access to both the database structure (i.e., the description of the data types stored in the database) and the real data. In that way, some standard languages have been proposed: SQL to query relational databases, OQL to query object oriented database and XQuery for XML based databases. Particular versions of these languages, and especially SQL, have also been adapted to target the complex process of simultaneously querying several (possibly distributed) databases (e.g., IBM's DiscoveryLink [HAS2001]).

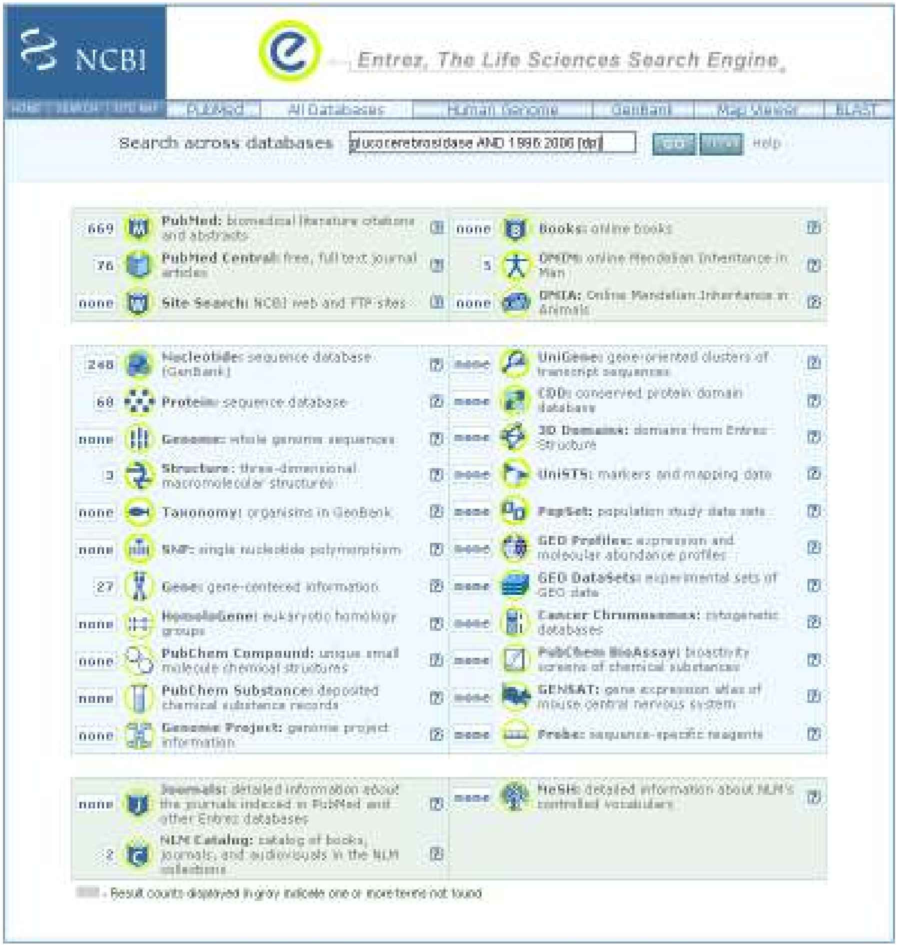
For the purpose of end users that need to access data repositories, but are not familiar with computer systems, a query language is far from being easy to use. First, there is the difficult step of learning these languages. Moreover, in the field of biomedical information, different types of data (sequences, micro-arrays, images, etc) may be stored in different DBMS, requiring to lean different languages. Second, the data structure (i.e. data types, attributes) has to be known to formulate a query, and that kind of information is not directly accessible at query writing time. Finally, since we are talking about command-line access to databases, query results are frequently presented in poorly-formatted text forms that are rarely straightforward to interpret.

3.1.1.2.2 Web-based access

To circumvent the problem of using a command-line, databases access has greatly benefited from the advance of the World Wide Web. Now, the end-users can fill in pre-formatted query forms and the results can be nicely formatted to help users interpreting the query results.

Such web-based database accesses are widely spread in the biomedical community; see for example the extensively used web portal of the US National Centre for Biomedical Information (NCBI, www.ncbi.nlm.nih.gov).

If this kind of database access is quite easy for the end-users, there is a rapidly emerging lack: a user can only execute particular queries since they are pre-formatted with web forms that do not give access to the full expressivity power of previously mentioned query languages. Taking the example of the NCBI, a query is usually a boolean expression of text-free keywords possibly decorated with a data type (figure 20 and figure 21). However, that system does not give a direct access to the data types at query writing time: one has to read some additional documentation situated elsewhere on their web portal.

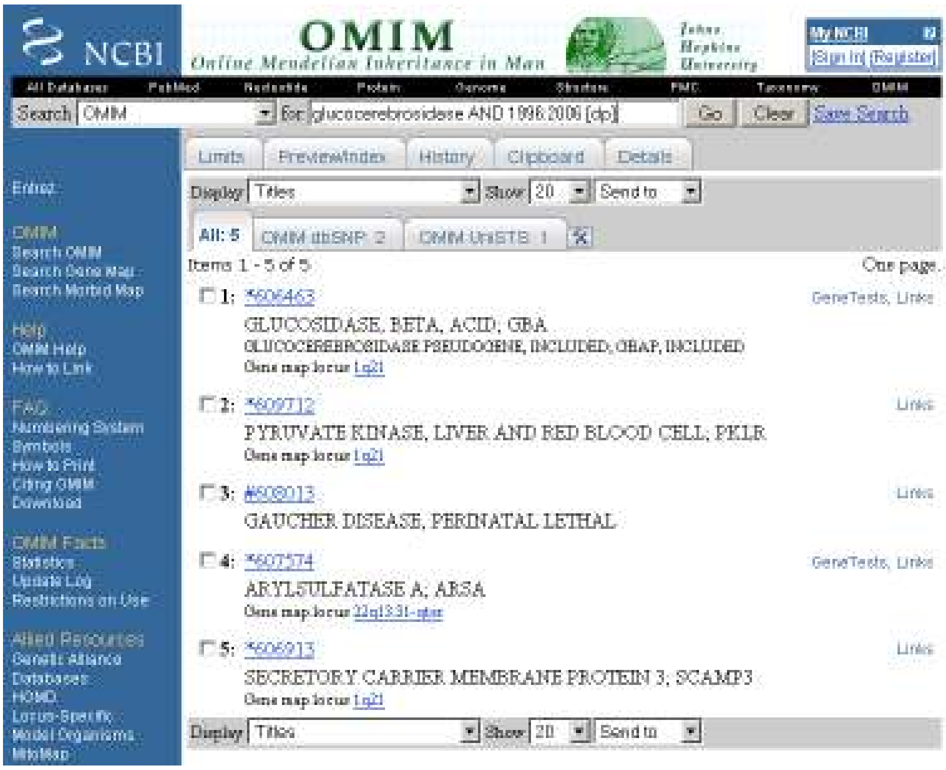


Search across databases glucocerebrosidase AND 1996 2006 [dp]

669	**PubMed:** biomedical literature citations and abstracts	none	**Books:** online books
76	**PubMed Central:** free, full text journal articles	5	**OMIM:** online Mendelian Inheritance in Man
none	**Site Search:** NCBI web and FTP sites	none	**OMIA:** Online Mendelian Inheritance in Animals
248	**Nucleotide:** sequence database (GenBank)	none	**UniGene:** gene-oriented clusters of transcript sequences
68	**Protein:** sequence database	none	**CDD:** conserved protein domain database
none	**Genome:** whole genome sequences	none	**3D Domains:** domains from Entrez Structure
3	**Structure:** three-dimensional macromolecular structures	none	**UniSTS:** markers and mapping data
none	**Taxonomy:** organisms in GenBank	none	**PopSet:** population study data sets
none	**SNP:** single nucleotide polymorphism	none	**GEO Profiles:** expression and molecular abundance profiles
27	**Gene:** gene-centered information	none	**GEO DataSets:** experimental sets of GEO data
none	**HomoloGene:** eukaryotic homology groups	none	**Cancer Chromosomes:** cytogenetic databases
none	**PubChem Compound:** unique small molecule chemical structures	none	**PubChem BioAssay:** bioactivity screens of chemical substances
none	**PubChem Substance:** deposited chemical substance records	none	**GENSAT:** gene expression atlas of mouse central nervous system
none	**Genome Project:** genome project information	none	**Probe:** sequence-specific reagents
none	**Journals:** detailed information about the journals indexed in PubMed and other Entrez databases	none	**MeSH:** detailed information about NLM's controlled vocabulary
2	**NLM Catalog:** catalog of books, journals, and audiovisuals in the NLM collections		

- Result counts displayed in gray indicate one or more terms not found

Figure 17: Querying biomedical data on the NCBI web portal (http://www.ncbi.nlm.nih.gov/). This web page snapshot displays the results of searching for all entries published between 1996 and 2006 related to 'glucocerebrosidase'. This page gives a very interesting overview of the results found in the various databases maintained at the NCBI.



Search OMIM for glucocerebrosidase AND 1996:2006 [dp]

Display Titles Show 20 Send to

All: 5 | OMIM dbSNP: 2 | OMIM UniSTS: 1

Items 1 - 5 of 5 One page.

□ 1: *606463 GeneTests, Links
GLUCOSIDASE, BETA, ACID; GBA
GLUCOCEREBROSIDASE PSEUDOGENE, INCLUDED; GBAP, INCLUDED
Gene map locus 1q21

□ 2: *609712 Links
PYRUVATE KINASE, LIVER AND RED BLOOD CELL; PKLR
Gene map locus 1q21

□ 3: #608013 Links
GAUCHER DISEASE, PERINATAL LETHAL

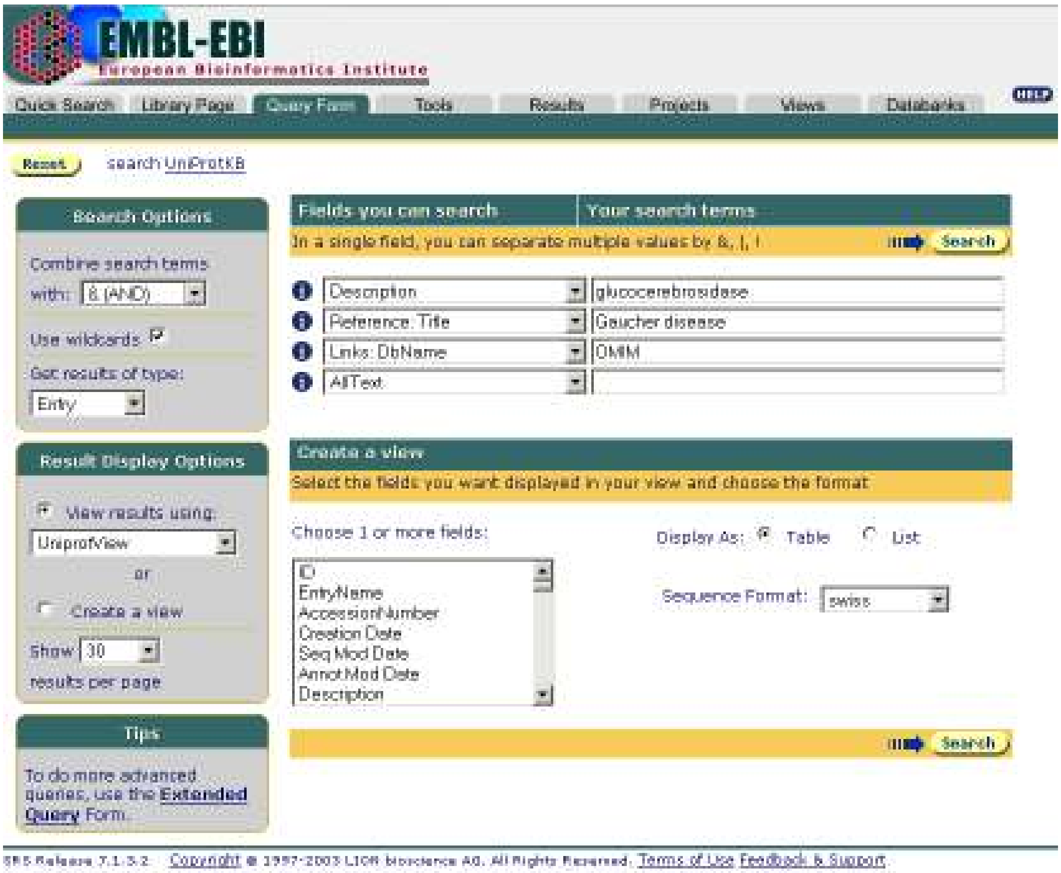
□ 4: *607574 GeneTests, Links
ARYLSULFATASE A; ARSA
Gene map locus 22q13.31-qter

□ 5: *606913 Links
SECRETORY CARRIER MEMBRANE PROTEIN 3; SCAMP3
Gene map locus 1q21

Display Titles Show 20 Send to

Figure 18: Results of the query from figure 20 that relate to the OMIM database. On such a page a very relevant information, apart a summary of the 5 OMIM entries that relates to our query, is the 'Links' hypertext link located on the right of each summary. Each link allows the user to see other types of data located in other NCBI's databases.

More sophisticated web-based systems exist, such as SRS [ETZ1996] and TAMBIS [STE2000], which provide the possibility to create more complex queries either with HTML forms (figure 22) or using Web browser embedding Java Applets (figure 23). Usually these forms allow the specification of queries with visually created boolean expressions, the data types being presented to the user (like on figure 22). Query creation is then facilitated, but these more advanced systems still limit the expressivity of queries in comparison with database query languages.



search UniProtKB

Search Options

Combine search terms with: & (AND)

Use wildcards ☑

Get results of type: Entry

Result Display Options

● View results using: UniprotView

or

○ Create a view

Show 30 results per page

To do more advanced queries, use the Extended Query Form.

Fields you can search	Your search terms
Description	glucocerebrosidase
Reference: Title	Gaucher disease
Links: DbName	OMIM
AllText	

In a single field, you can separate multiple values by &, \|, !

Create a view

Select the fields you want displayed in your view and choose the format

Choose 1 or more fields:

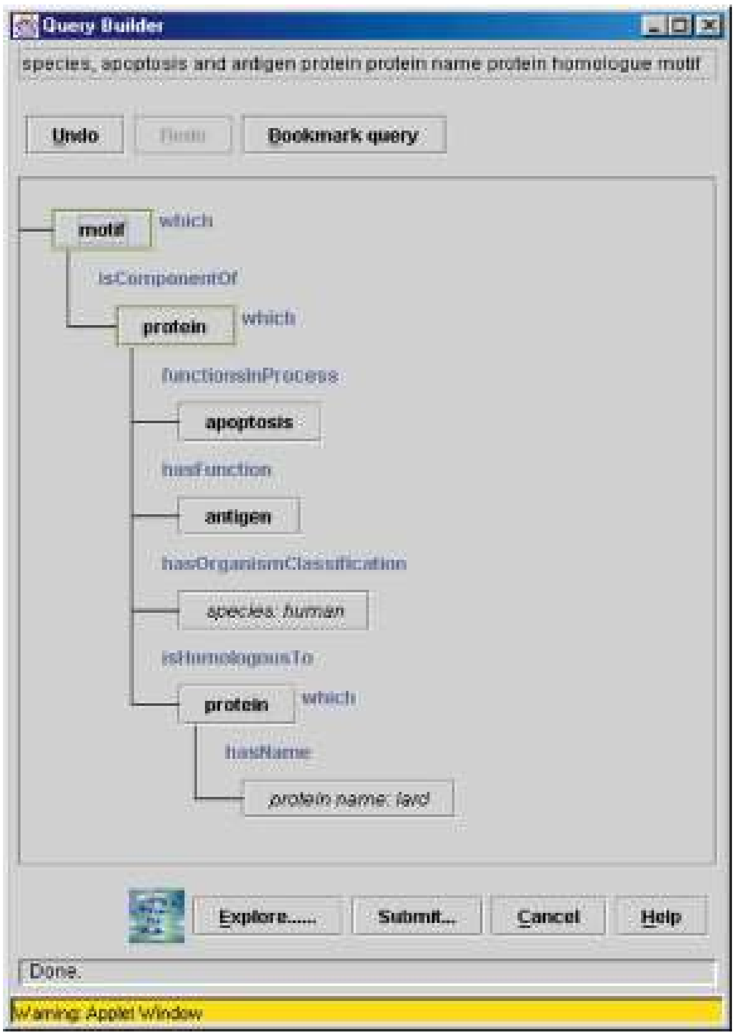
ID
EntryName
AccessionNumber
Creation Date
Seq Mod Date
Annot Mod Date
Description

Display As: ● Table ○ List

Sequence Format: swiss

SRS Release 7.1.3.2 Copyright © 1997-2003 LION bioscience AG. All Rights Reserved. Terms of Use Feedback & Support

Figure 19: SRS query at the EBI web portal (http://srs.ebi.ac.uk).



species, apoptosis and antigen protein protein name protein homologue motif

Figure 20: TAMBIS graphical query builder running from Mozilla web browser.

3.1.1.2.3 Interactive Graphical User Interface access

Interactive graphical user interface access to database systems relies upon visual query languages (VQLs). Query by Example (QBE, [ZLO1977]) has been the first VQL proposed to query relational databases. Queries are created by 'assembling' visual representation of tables, constraints being added in the columns of the tables. This system has then been extended to provide a more convenient graphical display (figure 24) still available today in database software such as Microsoft Access.

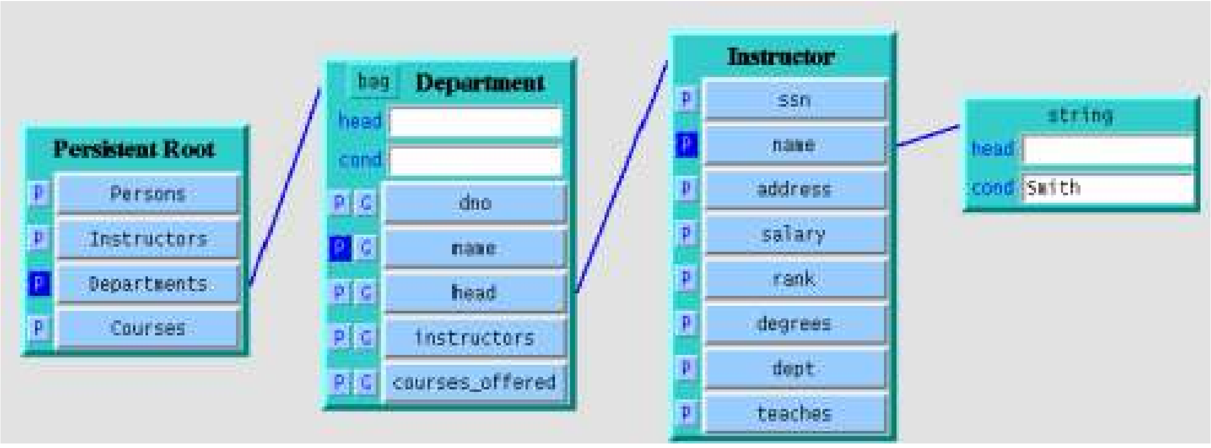
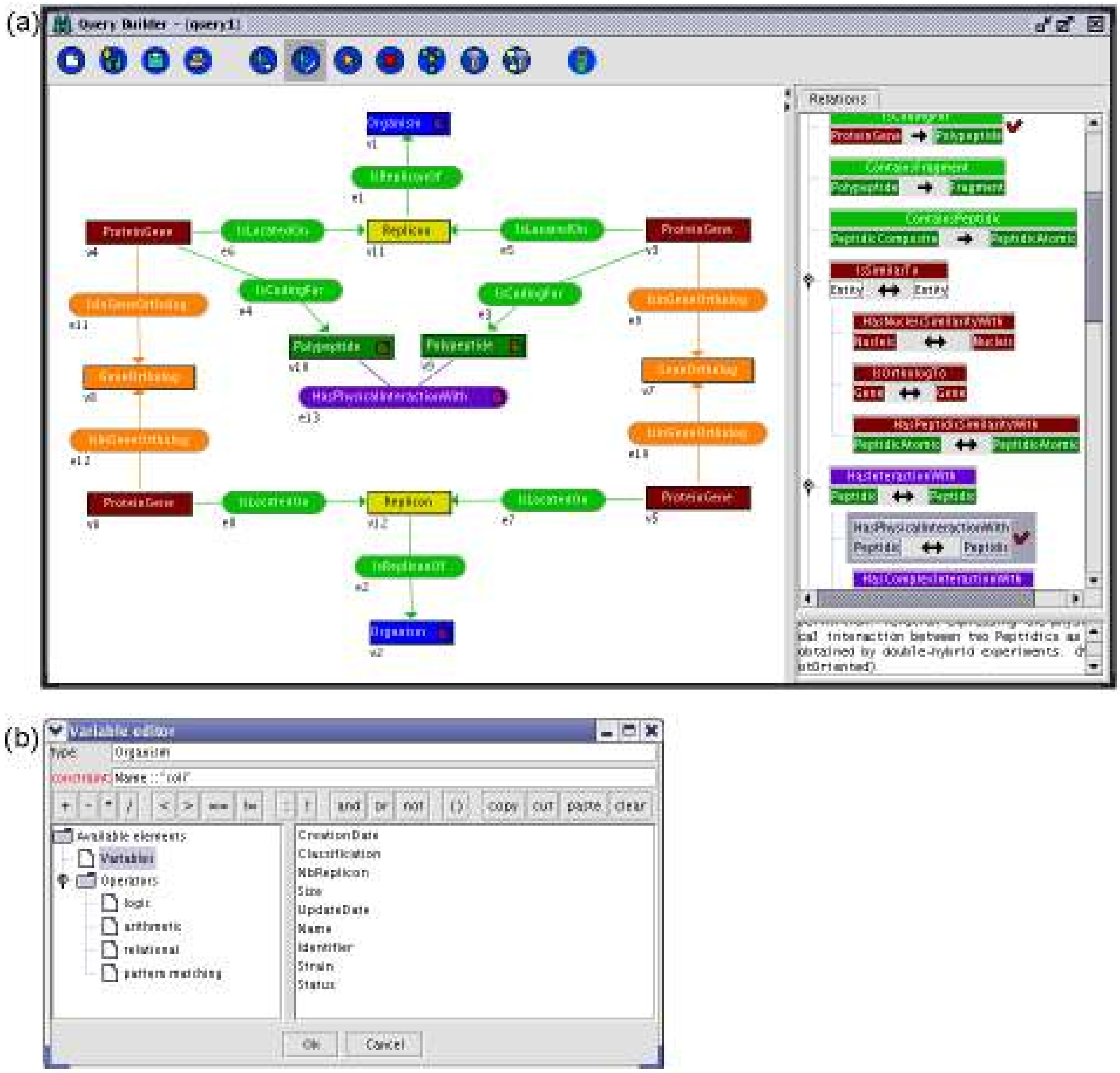


Figure 21: An example of modern Query by Example (from [DOT2006]).

Visual queries described using the graph paradigm is probably the most prominent VQL today available. The graph is used to represent the elements of the schema describing the database structure. Significant systems like GOOD [GEM1993], Hy+ [CON1997], Gql [PAP1994], Hyperlog [POU2001], the system from Butler et al. [BUT2005], Snow [SNOW], HyperFlow [DOT2005] and GenoLink[DUR2006] provide visual graph query 'languages'.

Such systems are of particular interest for the end users. First, they usually explicitly display the schema model as a unique graph (figure 25). In that way, the user does not need to know which kind of database he/she targets: whether underlying DBMS is relational or object oriented, whether users target a single database or a set of distributed ones, the visual interface displays the schema in a unique way. The schema graph itself may rely upon a particular data modelling system (see below) providing more flexibility and independence with regard to the DBMS implementation(s) used.



cal interaction between two Peptidics as obtained by double-hybrid experiments. (
otOriented)

Type: Organism

constraint: Name :: "coli"

Figure 22: The GenoLink Query Builder. (a) Main window. The left panel displays the graph query being constructed. The right panel displays either the hierarchy of classes or the hierarchy of associations of the data model. Here the user is adding an association therefore the hierarchy of associations is shown. The associations with non empty set of instances are marked with a red "V", allowing the user to quickly know data types having real instances in the database. (b) Clicking on a vertex or edge will popup this constraint editor to add an algebraic constraint on the corresponding object. Here the name of the organism (represented by vertex v2) should match "coli".

Second, both the query and the results are displayed as graph facilitating the interpretation of results since they are written like the query. GenoLink goes one step further by allowing the users to visually explore the neighbours of vertices reported in the results (figure 26).

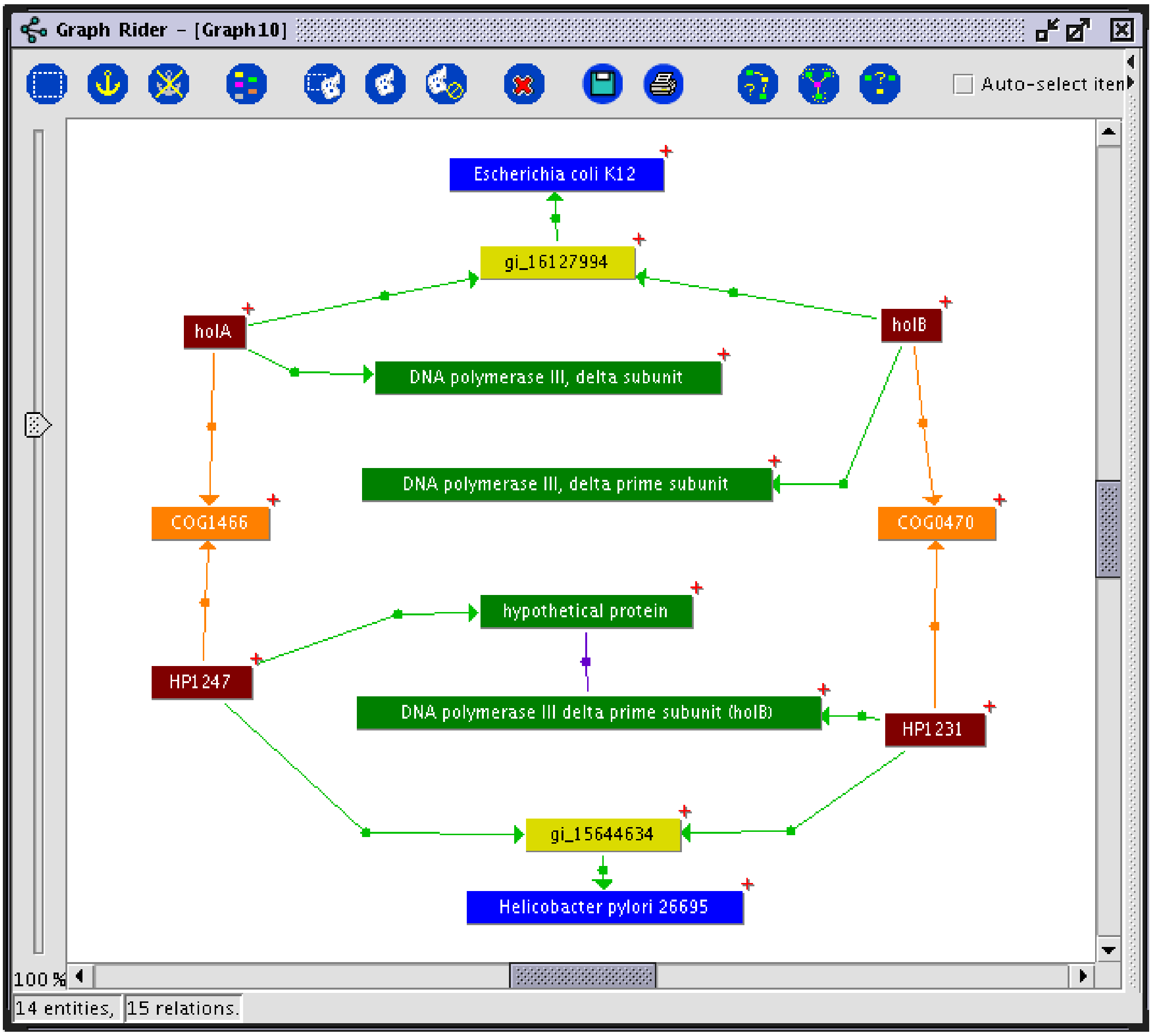
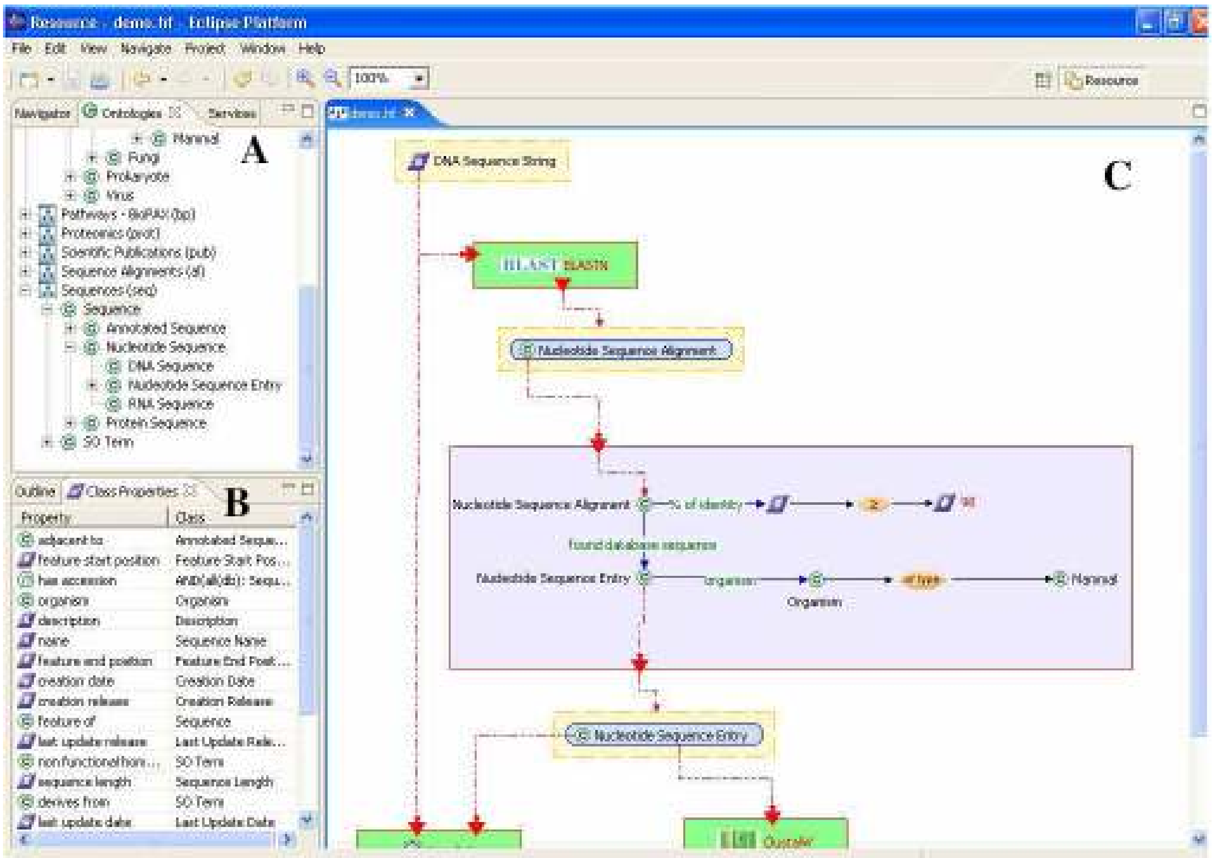


Figure 23: GenoLink Result Graph Explorer. This snapshot shows an example of a result graph corresponding to the Query from figure 25. The edge linking the two H. pylori Polypeptides corresponds to a physical interaction. The red crosshair on the top-right of some vertices denotes that they are linked to some others that are not currently shown. These vertices may therefore be further expanded to gain more information about the full data graph. In this example, this operation has been performed on vertices holA and holB (from E. coli) in order to display the corresponding Polypeptides (DNA polymerase III) that were not part of the query (see figure 25).

Third, systems like HyperFlow and GenoLink, relies upon an intermediate data modelling system capable of representing complex data schema. This additional level of abstraction implies that HyperFlow and GenoLink do not rely on a particular DBMS implementation. HyperFlow relies on OWL, whereas GenoLink relies on an entity-relationship knowledge representation system (AROM, [GEN2000]). Vertices and edges of their query graphs are then tidily linked to OWL or AROM entities. Now, to execute a query against a real database (where the data is actually stored), the graph query has to be translated and passed to the DBMS for execution. This step is not yet implemented in HyperFlow. GenoLink uses a different approach since it implements its own graph query engine, the DBMS being only used to feed that graph query engine with real data.

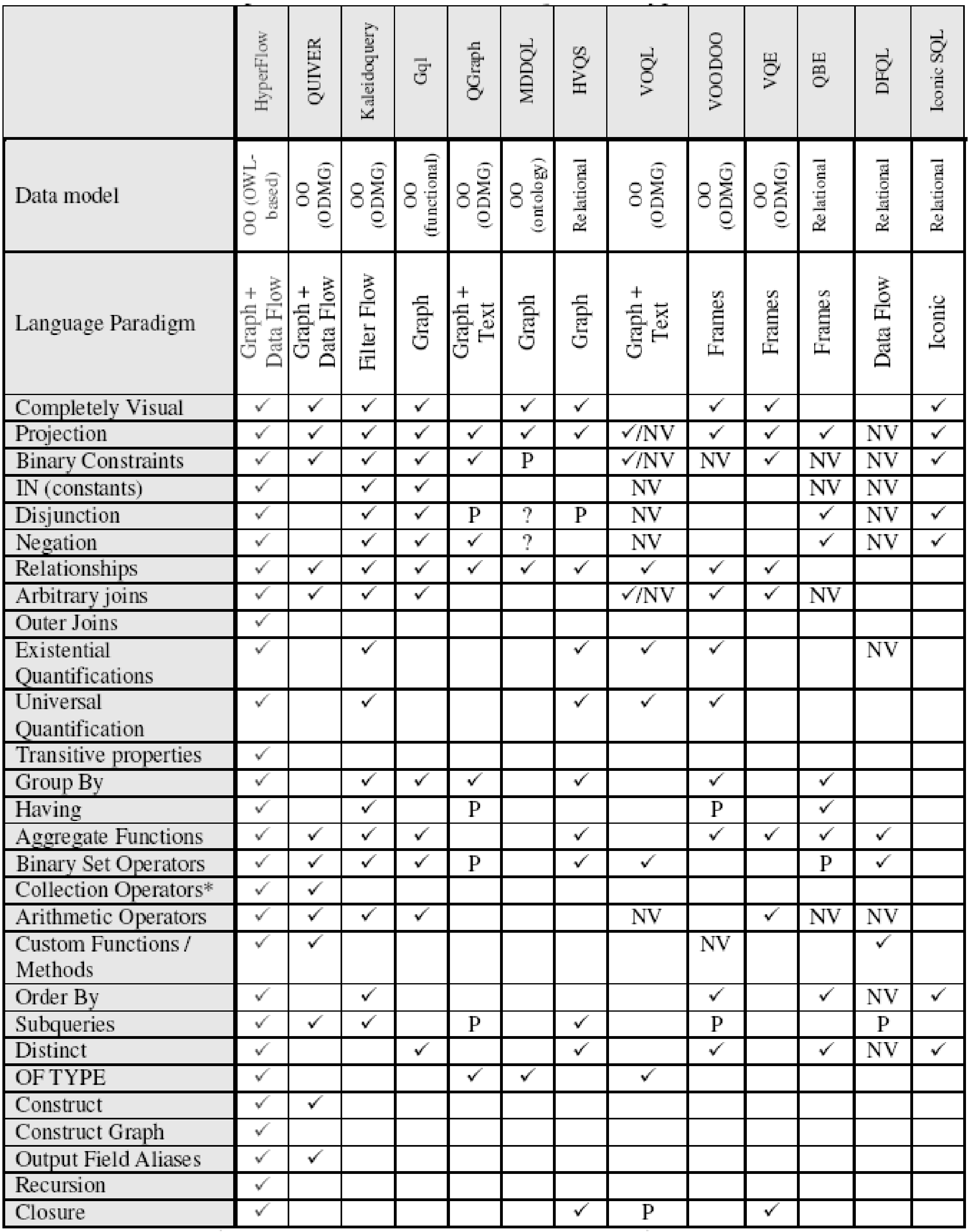
Finally, a system like HyperFlow combines a visual query language with a visual scientific workflows builder, thus providing a single graphical user interface capable of creating complex 'queries' in an easy way (figure 27).



Property	Class
adjacent to	Annotated Seque...
feature start position	Feature Start Pos...
has accession	AND(all(db): Sequ...
organism	Organism
description	Description
name	Sequence Name
feature end position	Feature End Posit...
creation date	Creation Date
creation release	Creation Release
feature of	Sequence
last update release	Last Update Rele...
nonfunctionalhom...	SO Term
sequence length	Sequence Length
derives from	SO Term
last update date	Last Update Date

Figure 24: HyperFlow Framework. (a) Ontology from which the user can create the query. (b) Properties of the Sequence type selected in (a). (c) Workflow/query graph builder. The part that is really a query graph is highlighted by the blue rectangle. (Example from [DOT2006]).

Figure 28 presents a table comparing the expressivity power of various VQL-based systems. This could be of interest to determine the functionalities to implement for ACGT's VQL system.



	HyperFlow	QUIVER	Kaleidoquery	Gql	QGraph	MDDQL	HVQS	VOQL	VOODOO	VQE	QBE	DFQL	Iconic SQL
Data model	OO (OWL-based)	OO (ODMG)	OO (ODMG)	OO (functional)	OO (ODMG)	OO (ontology)	Relational	OO (ODMG)	OO (ODMG)	OO (ODMG)	Relational	Relational	Relational
Language Paradigm	Graph + Data Flow	Graph + Data Flow	Filter Flow	Graph	Graph + Text	Graph	Graph	Graph + Text	Frames	Frames	Frames	Data Flow	Iconic
Completely Visual	✓	✓	✓	✓		✓	✓		✓	✓			✓
Projection	✓	✓	✓	✓	✓	✓	✓	✓/NV	✓	✓	✓	NV	✓
Binary Constraints	✓	✓	✓	✓	✓	P		✓/NV	NV	✓	NV	NV	✓
IN (constants)	✓		✓	✓				NV			NV	NV	
Disjunction	✓		✓	✓	P	?	P	NV			✓	NV	✓
Negation	✓		✓	✓	✓	?		NV			✓	NV	✓
Relationships	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Arbitrary joins	✓	✓	✓	✓				✓/NV	✓	✓	NV		
Outer Joins	✓												
Existential Quantifications	✓		✓				✓	✓	✓			NV	
Universal Quantification	✓		✓				✓	✓	✓				
Transitive properties	✓												
Group By	✓		✓	✓	✓		✓		✓		✓		
Having	✓		✓		P				P		✓		
Aggregate Functions	✓	✓	✓	✓			✓		✓	✓	✓	✓	
Binary Set Operators	✓	✓	✓	✓	P		✓	✓			P	✓	
Collection Operators*	✓	✓											
Arithmetic Operators	✓	✓	✓	✓				NV		✓	NV	NV	
Custom Functions / Methods	✓	✓							NV			✓	
Order By	✓		✓						✓		✓	NV	✓
Subqueries	✓	✓	✓		P		✓		P			P	
Distinct	✓			✓			✓		✓		✓	NV	✓
OF TYPE	✓				✓	✓		✓					
Construct	✓	✓											
Construct Graph	✓												
Output Field Aliases	✓	✓											
Recursion	✓												
Closure	✓						✓	P		✓			

Figure 25: Comparison of VQL systems expressivity (from [DOT2006]). NV – supported in a non visual manner. P – Partially supported. * - Collection operators – listtoset, flatten, element, etc.

References

[BUT2005] Butler G., Wang G., Wang Y. and Zou L. A graph database with visual queries for genomics. Proceedings of the 3rd Asia-Pacific Bioinformatics Conference (APBC2005): 17-21 January 2005, Singapore.

[CON1997] Consens M. and Mendelzon A. Hy+: a Hygraph-based query and visualization system. SIGMOD Rec 1997, 22(2), 511-516.

[DOT2006] Dotan, D. (2006). HyperFlow: a Visual, Ontology-Based Query and Data-Flow Language for End-User Information Analysis. Master of Science in Computer Science thesis. Technion - Israel Institute of Technology, Haifa, Israel.

[DOT2005] Dotan, D. and Pinter, RY. HyperFlow: an Integrated Visual Query and Data-Flow Language for End-User Information Analysis. Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05), September 2005, pp. 27-34.

[DUR2006] Durand P, Labarre L, Meil A, Divol JL, Vandenbrouck Y, Viari A, Wojcik J (2006) GenoLink: a graph-based querying and browsing system for investigating the function of genes and proteins. BMC Bioinformatics. 7(1):21

[ETZ1996] Etzold T, Ulyanov 4 Argos P (1996). SRS: Information retrieval system for molecular biology data banks. Methods Enzymol, 266:114-128.

[GEM1993] Gemis M., Paredaens J., Thyssens I. and Van den Bussche J. GOOD: a graph-oriented object database system. SIGMOD Rec 1993, 22(2), 505-510.

[GEN2000] Genoud, P., Dupierris, V., Page, M., Bruley, C., Ziebelin, D., Gensel, J. and Bardou, D. From AROM, a new object based knowledge representation system, to WebAROM, a knowledge bases server. 9th Int. Conf. on Artificial Intelligence: Methodology, Systems, and Applications: 20-23 September 2000, Varna, Bulgaria. (http://www.inrialpes.fr/sherpa/arom/index.html)

[HAS2001] Haas LM, Schwarz PM, Kodali P, Kotlar E, Rice JE, Swope WC (2001). DiscoveryLink: A system for integrated access to life sciences data sources. IBM Syst J, 40:489-511.

[STE2000] Stevens R, Baker P, Bechhofer S, Ng G, Jacoby A, Paton NW, Goble CA, Bras A (2000). TAMBIS: Transparent access to multiple bioinformatics information sources. Bioinformatics, 16:184-185.

[PAP1994] Papantonakis A. and King P.J.H Gql, a declarative graphical query language based on the functional data model. Proceedings of the workshop on Advanced visual interfaces (AVI '94): June 1-4, 1994, Bari, Italy, 113-122.

[POU2001] Poulovassilis A. and Hild S. Hyperlog: A Graph-Based System for Database Browsing, Querying, and Update. IEEE Trans. Knowl. Data Eng 2001, 13(2), 316-333.

[SNOW] Snow. http://www.northbears.org/.

[ZLO1977] Zloof MM (1977). Query by example - a data base language. IBM Syst J, 16(4):324-343.

3.2 SEMANTIC MEDIATION AND DATABASE INTEGRATION

3.2.1 INTRODUCTION

A mediator is a software module that exploits encoded knowledge about certain sets or subsets of data to create information for a higher layer of applications [WIE1992]. Mediators provide:

- Transformation of databases.
- Methods to access and merge data from multiple databases.
- Abstraction and generalization of underlying data.

Mediation is close to the middleware concept. A mediator can act as a source of information for another mediator, since it provides services to access data to higher layers. Usually, a wrapper is needed for every data source, and the integration of such data sources is done under demand.

Semantic mediation aims to solve the problem of discovering data in sources of information that cannot be accessed easily. A semantic mediation system should include services for formulating semantic queries, and should give transparent access to heterogeneous sources of data.

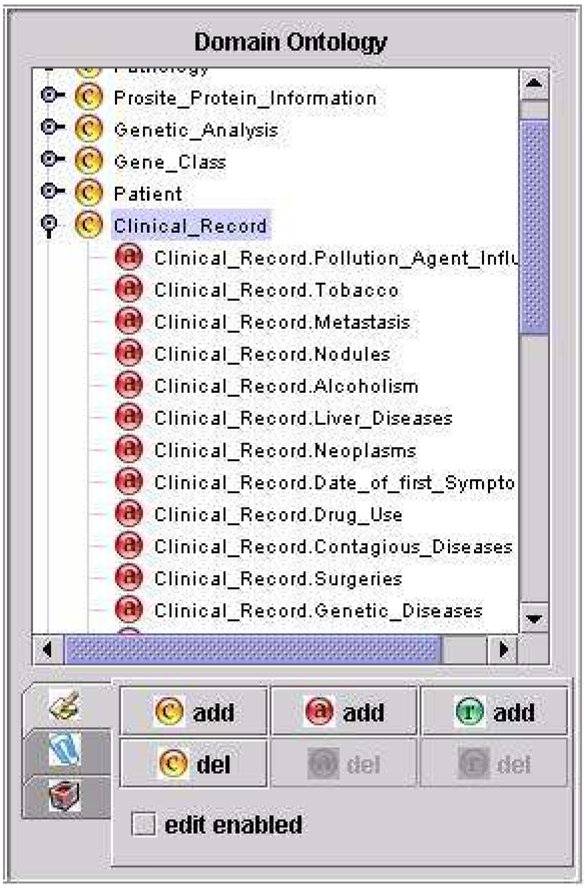
Heterogeneous database integration is a key issue in semantic mediation. The integration and access to heterogeneous sources of information can be approached in several ways, being the ontology-based approach one of the most effective, understandable and reliable methods.

Ontologies provide the semantics needed to bridge the gap between heterogeneous data sources and a formal language for information retrieval. In a semantic mediation system, the user (either if this user is a human or not) should not take care about the format of the information source, but about the terms contained in the ontology for building the query in a proper way. The system would provide then a virtual view of the data, based on how ontologies describe the domain or domains implied.

3.2.2 ONTOLOGIES IN INFORMATION SYSTEMS

There are numerous definitions for the term “Ontology”,. One of the most cited was proposed by Gruber: “An ontology is an explicit specification of a conceptualization” [GRU1993], as stated in a previous section of this document. We can also describe an ontology as what it provides: a conceptual framework for a structured representation of the meaning, through a common vocabulary, of a given domain (e.g.. medical ontologies describe certain medical domain), specifying concepts, relationships between such concepts and axioms in a formal manner.

An ontology can be seen as a set of classes and the relationships among them. It can be complemented with restrictions and instances of the elements belonging to the different classes described. Figure 29 shows a view of an ontology subset in the tool ONTOFUSION.



Domain Ontology

Pathology
Prosite_Protein_Information
Genetic_Analysis
Gene_Class
Patient
Clinical_Record
Clinical_Record.Pollution_Agent_Influ
Clinical_Record.Tobacco
Clinical_Record.Metastasis
Clinical_Record.Nodules
Clinical_Record.Alcoholism
Clinical_Record.Liver_Diseases
Clinical_Record.Neoplasms
Clinical_Record.Date_of_first_Sympto
Clinical_Record.Drug_Use
Clinical_Record.Contagious_Diseases
Clinical_Record.Surgeries
Clinical_Record.Genetic_Diseases

Figure 26: Example of a domain ontology in Ontofusion

There have been some approaches for defining formal ontologies, both in biological and medical domains, such as the Gene Ontology (GO). It aims to provide a controlled vocabulary to describe gene and gene product attributes in any organism, and the Unified Medical Language System (UMLS), which is a compendium of medical terms.

The real benefit from using ontologies in software development comes from the capacity of such ontologies to make data “understandable” for software entities. By having and explicit and formal definition of a given domain, our applications are able to categorize and manage data given its semantic meaning, something that was unavailable previously. On the other hand, ontologies are simple enough for humans to work with them, allowing experts to easily translate their knowledge about a given area into computer understandable knowledge.

However, we must be aware that the use of ontologies in information systems may have its drawbacks. For example, the increase in complexity of research projects and the lack of well-established standards for ontology construction and edition.

3.2.3 DATABASE INTEGRATION IN THE BIOMEDICAL DOMAIN

Last years the amount of information produced and stored in databases has increased greatly. For example the Human Genome project has led to large amounts of data that have been collected in different databases. To manage and handle all this information, a new area in computer science has emerged: database integration.

Database integration is the area of computer science related with information exchange and gathering, usually from heterogeneous and disparate sources. It faces problems such as bringing together data with different patterns, or allowing users to access to information located in different places in a uniform manner.

There are two basic approaches to database integration: centralized vs. federated. The centralized approach relies on a central repository where all data are to be stored, called "Data Warehouse" [KIM1996]. Users will finally access data stored in such integrated database. The Data Warehouse has its own data model, which is independent from the original databases. This allows fast response to user queries, since all data are collected locally. However, it also has several drawbacks, such as the possibility of inconsistencies in the data (changes in the original sources may take time to reach the central repository), the elevated cost of maintaining the repository and the need of additional space, since the Data Warehouse is a new database. Figure 30 shows a representation of integration in a Data Warehouse.

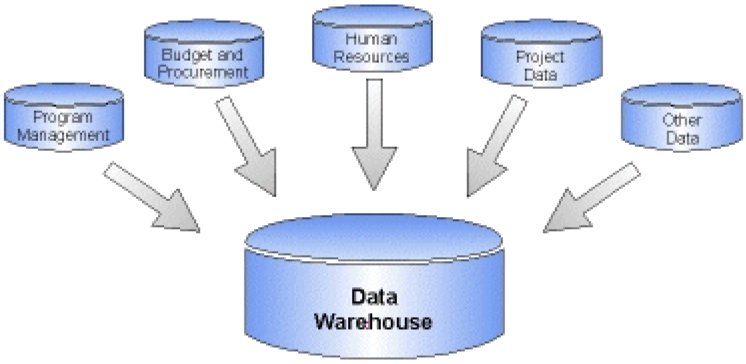


Figure 27: A graphical representation of a Data Warehouse

In the other hand, the federated approach does not rely on a central repository, leaving the data in the original sources. This is actually being more used nowadays, since it solves the problems of the centralized approach. The federated approach was first introduced by [SHE1990], and was called Federated Database System (FDBS). In a FDBS, databases in the system are autonomous, and their local operations do not depend on the FDBS. Nevertheless, this first approach had some drawbacks itself, like the difficulty for updating data sources (include or remove data).

Another distributed approach, known as "mediation", was later introduced [WIE1992]. The mediator is a middleware layer between the user and the data sources. Mediation is not based in a Data Management System.

In this case we have a virtual model composed of all the databases we want to include. User queries are translated to queries on those sources, and results are merged together before presenting them to the user. Therefore, the user *sees* a central repository containing all the data (all translations and integrations are performed transparently). This eliminates the drawbacks exposed before. However performance may be lower in this case since data must be retrieved from the sources for each query.

Systems that rely on accessing federated data sources are called “query translation systems” as well. Figure 31 shows the general architecture of a query translation system.

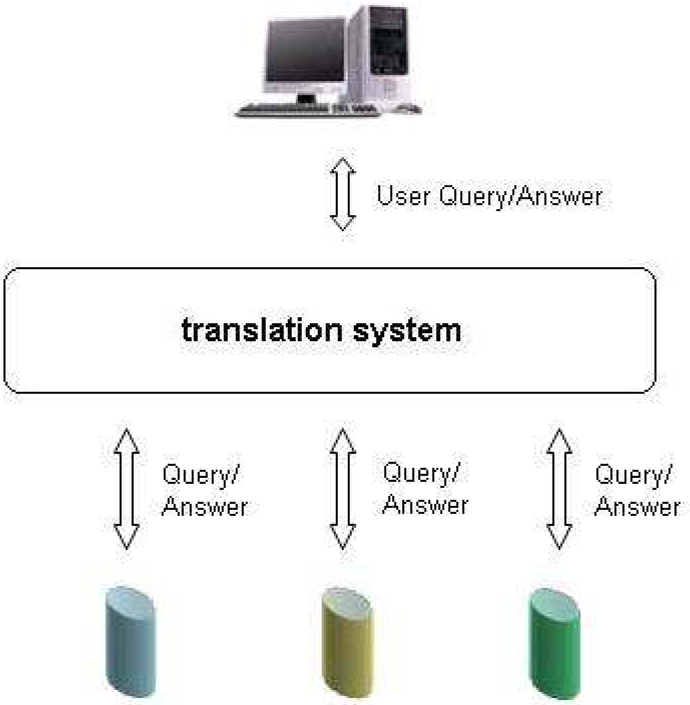


Figure 28: An example of federated database integration approach

Query translation systems can be classified in four categories: 1) pure mediation, 2) single conceptual schema approaches, 3) multiple conceptual schema approaches and 4) hybrid approaches [PER2004].

1) Pure Mediation Systems:

   In pure mediation systems there is no alternative data model presented to the user. Instead of that, a mediator is used to resolve user queries. A mediator is a software layer, close to the concept of middleware.

Main drawbacks: Pure mediation systems are usually not very much intuitive.

2) Global Conceptual Schema Systems:

   Global Conceptual Schema Systems are based on a single ontology that models the domain of interest. Database objects are linked to objects belonging to the global ontology.

Main Drawbacks: Addition or removal of databases may require the modification of the global ontology.

3) Multiple Conceptual Schema Systems:

   In the Multiple Conceptual Schema Systems each source is described using a different ontology.

Main Drawback: It is not possible to ensure that semantically equivalent entities share names. It is required to create mappings among semantically similar objects belonging to different ontologies.

4) Hybrid approach

   In such Hybrid Approaches each source is described using a different ontology. Each one of these ontologies is built using objects from an ontology approved by domain experts. Using this approach semantically equivalent objects, belonging to different ontologies, share their names.

Main Drawback: Validated domain ontology is required.

3.2.4 ONTOLOGIES APPLIED TO DATABASE INTEGRATION

Database integration requires bridging the syntactic and semantic gaps existing across data sources. Given the suitability of ontologies to provide a semantic layer to applications, database integration is moving towards an ontology-based approach. This approach seems to be promising, although there are still several issues that must be addressed.

In the biomedical domain, it has been demonstrated that ontologies can aid BI-MI integration, since they are mainly used to facilitate knowledge distribution, sharing and reuse. [PER2004]

In order to apply ontologies to database integration, several current systems use ontology-based views to facilitate the mapping from objects of specific databases to shared vocabularies. There are other approaches, with minor acceptance, such as the use of ontologies for automatic mediator generation. Some of these systems are reviewed later in this document.

Database integration is evolving towards ontology-based approaches, where ontologies are used to support mapping between equivalent concepts for integration and query formulation. It assumes that ontologies provide a common and shared vocabulary, which can be used to facilitate the communication and information transportation between users, systems and databases.

3.2.5 PROJECTS AND INTERNATIONAL INITIATIVES

3.2.5.1 DataFoundry

DataFoundry [CRI1998][CRI2001][CRI1998] is a project aimed to improve scientists' access to distributed and heterogeneous data. The approach in this project was to use ontologies for automatic mediator generation. It reduces the efforts needed to include new data sources, or existing sources that have experienced changes in their structure.

The ontology designed for this project stored metadata about the generation of mediators, that can be therefore automatically created. It included knowledge to identify and resolve both syntactic and semantic conflicts between data contained in the sources, allowing the unification of concepts contained in such data.

The ontology was composed of four basic concepts, necessary for the mediator generation:

- Abstractions: abstractions of domain specific concepts.
- Databases: database descriptions.
- Mappings: mappings between a database and an abstraction.
- Transformations: functions to resolve representation conflicts.

Figure 32 shows the architecture of the DataFoundry System.

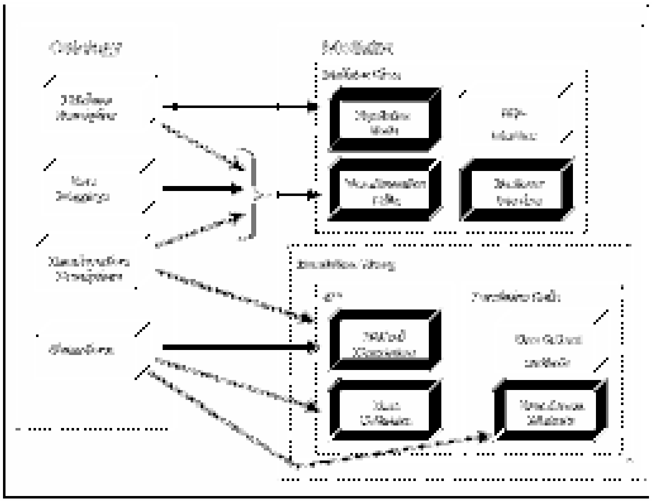


Figure 29: DataFoundry architecture

This project has already been completed.

3.2.5.2 LinkFactory

LinkFactory [VER2003] is an Ontology Management System (OMS) that offers users a GUI for creating and managing ontologies. It was for example used in the creation of LinkBase, an ontology covering the biomedical domain. The tool offers a multiple windows environment and a series of functions to allow users easily editing and managing ontologies.

LinkFactory includes and extension tool called MaDBoKS, which allows mapping external databases to ontologies. This way, any relational schemata can be mapped with an ontology, and use this information for integrating distributed heterogeneous data sources. Figure 33 shows LinkFactory GUI.

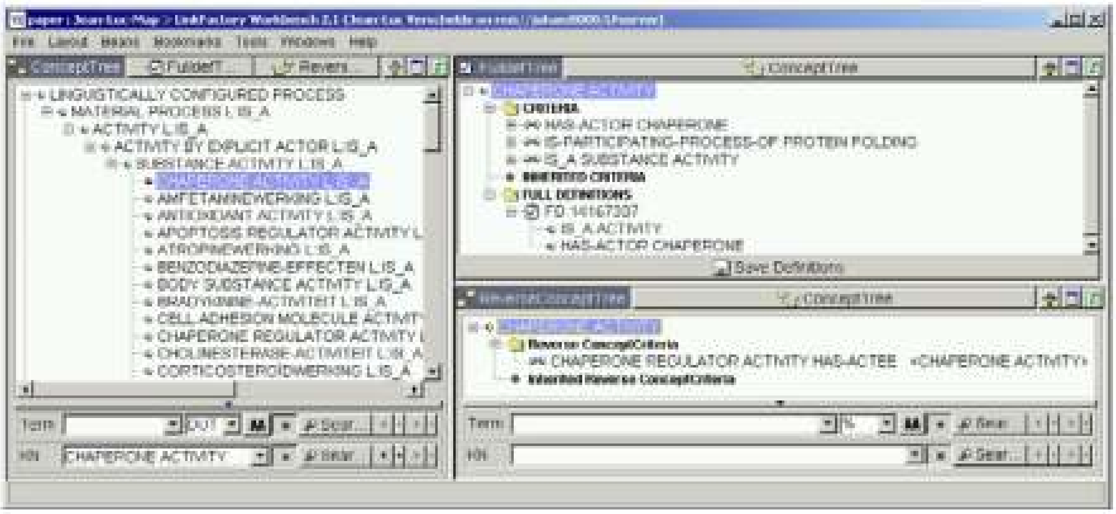

File Layout Beans Bookmarks Tools Windows Help
ConceptTree
FulldefT...
Revers...
LINGUISTICALLY CONFIGURED PROCESS
MATERIAL PROCESS L IS_A
ACTIVITY L IS_A
ACTIVITY BY EXPLICIT ACTOR L IS_A
SUBSTANCE ACTIVITY L IS_A
CHAPERONE ACTIVITY L IS_A
AMFETAMINEWERKING L IS_A
ANTIOXIDANT ACTIVITY L IS_A
APOPTOSIS REGULATOR ACTIVITY L
ATROPINEWERKING L IS_A
BENZODIAZEPINE-EFFECTEN L IS_A
BODY SUBSTANCE ACTIVITY L IS_A
BRADYKININE-ACTIVITEIT L IS_A
CELL ADHESION MOLECULE ACTIVIT
CHAPERONE REGULATOR ACTIVITY L
CHOLINESTERASE-ACTIVITEIT L IS_A
CORTICOSTEROIDWERKING L IS_A
Term
Sear...
Hit
CHAPERONE ACTIVITY
FulldefTree
ConceptTree
CHAPERONE ACTIVITY
CRITERIA
HAS-ACTOR CHAPERONE
IS-PARTICIPATING-PROCESS-OF PROTEIN FOLDING
IS_A SUBSTANCE ACTIVITY
INHERITED CRITERIA
FULL DEFINITIONS
FD 14167307
IS_A ACTIVITY
HAS-ACTOR CHAPERONE
Save Definitions
ReverseConceptTree
ConceptTree
CHAPERONE ACTIVITY
Reverse ConceptCriteria
CHAPERONE REGULATOR ACTIVITY HAS-ACTEE «CHAPERONE ACTIVITY»
Inherited Reverse ConceptCriteria
Term
Sear...
Hit
Sear...

Figure 30: LinkFactory GUI

3.2.5.3 SEMEDA

The SEMEDA [KOH2003] system offers semantic integration of biological databases. It is structured in three main components:

- MARGBench: offers query translation, thus enabling accessing data from distributed heterogeneous sources uniformly.
- SEMEDA: an ontology-based semantic metadatabase.
- SEMEDA-query: an ontology-based query interface.

SEMEDA facilitates collaborative work among different groups to construct and edit ontologies. This might be useful not only in database integration, but also in the creation of general purpose ontologies, such as Gene Ontology. The collaborating groups are classified in (i) Admins, (ii) DB Provider and (iii) Everybody, having each group different permissions. SEMEDA allows them defining concepts (well defined entities with unique meaning), relations between concepts and relational algebraic properties, such as symmetry, reflexivity and transitivity. These two last properties will allow deriving the semantic entailment of concepts, and will be especially important for SEMEDA's database query interface.

SEMEDA offers a web-based interface, which was developed using JSP with Oracle 8i. With SEMEDA helps users to query databases, examine the database tables/attributes and also build correct queries. It is also possible, through the SEMEDA Meta DB, to browse and edit semantic database metainformation. Administrative tools are also included for performing administrative tasks, through the Admin Tools option.

3.2.5.4 Hakimpour approach

Their creators suggested an approach for schema integration from different communities [HAK2001]. They propose that each group creates their own ontologies for representing concepts in their domain. These ontologies will be merged based on conceptual similarities. The final ontology, product of the fusion of all smaller ontologies, will be used to derive an integrated schema that can be used as a global schema in a federated database system.

This work pretends to solve semantic heterogeneity among different representations of data, usually due to equivalent concepts having different names. This happens very often when individual groups work on the same area of knowledge, and can be very harmful when seeking for adequate and meaningful data integration. By obtaining a global schema that integrates all local schemas, user will have a uniform and correct view of all the data. Resolving semantic heterogeneity is vital for this global schema to come out correctly. Otherwise the usage of integrated data may lead to invalid results. Figure 34 shows the global schema generation approach used in this system.

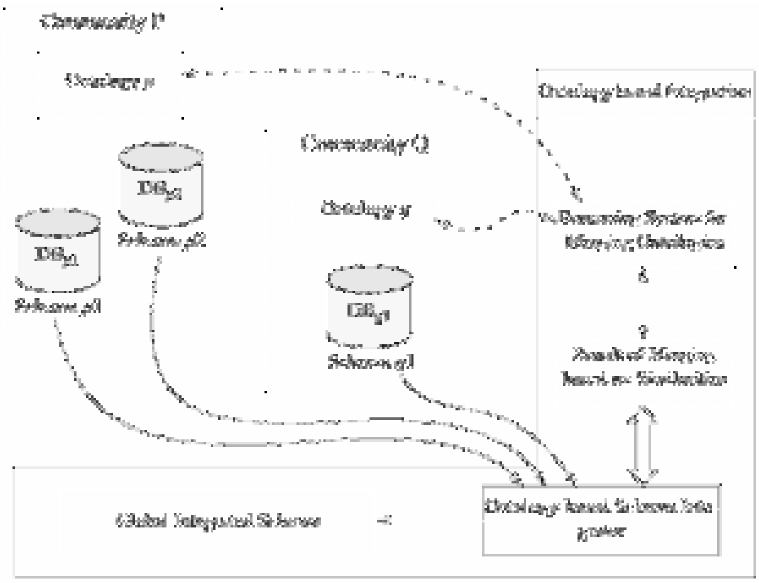

Community P
Ontology p
Ontology based Integration
Community Q
DB_{p2}
Schema p2
DB_{p1}
Schema p1
Ontology q
Reasoning System for Merging Ontologies
DB_{q1}
Schema q1
Levels of Merging based on Similarities
Global Integrated Schema
Ontology based Schema Integrator

Figure 31: Global schema generation from a common ontology, resulting from merging local ontologies

In order to obtain a global ontology from local merging ontologies, similarities and differences among concepts must be found. Similarity relations are defined among terms found in two ontologies, based in the intensional definitions (definitions of terms by logical axioms). There are four levels of similarities between two coherent intensional definitions:

- Disjoint definitions: when the two concept or relation intentional definitions imply a false outcome. This is the level with lowest degree of similarity.
- Overlapping definitions: when the intentional definitions conjunction cannot be proven to be false.
- Specialized definitions: one of the intentional definitions is an implication of the other one.
- Equal definitions: both intentional definitions are equivalent.

Ontologies can be merged by means of the similarities that are found within them. Given the level of the similarity, the merging process will be the following:

- For equal definitions, the result is a unique intentional definition, referred to by both original terms.
- If one definition is a specialization of another one, the similarity will be explicitly established between them.
- If one definition overlaps another one, an additional new concept or relation is declared as conjunction of both definitions.

The resulting global ontology will then be the key to build the global schema used to give users a uniform view of the data.

3.2.5.5 INFOGENMED/ONTOFUSION

INFOGENMED [PER2005a] is an information access workstation designed to facilitate access to private and public databases. Ontologies built on OWL language are used to map data sources (databases, text files, or even html pages) to virtual repositories, which are then mapped to a central virtual repository. Biomedical and other professionals use a graphical interface to navigate and query such repositories.

The system was initially designed as a multiagent-based system. The agents were in charge of solving the access problem to information sources. However, it has been recently redesigned towards a Web Services-based system, allowing an easier access to users.

Figure 35 shows the general architecture of the INFOGENMED approach. This approach has been used in the ONTOFUSION tool.

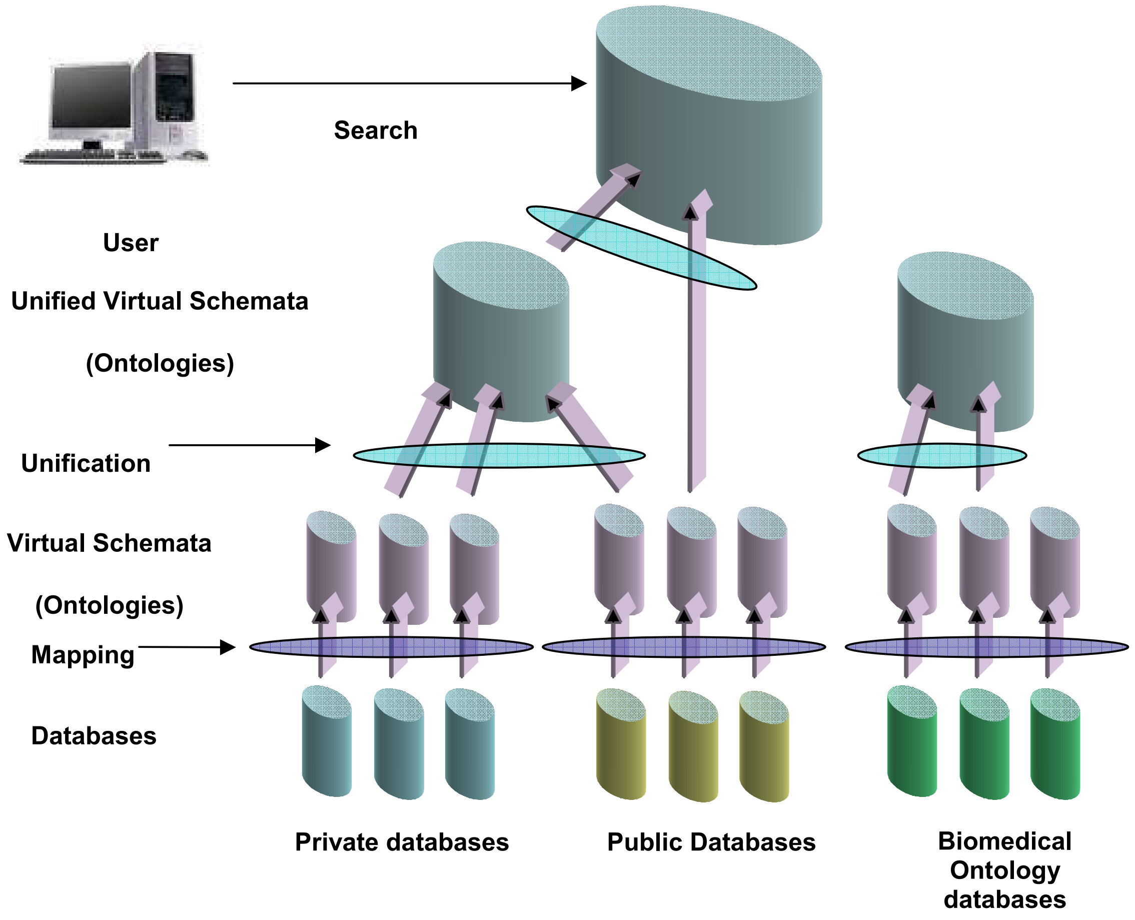


Figure 32: ONTOFUSION approach

Further improvements on the system will include the integration of a tool named OntoDataClean [PER2005b][PER2006], designed and developed by the same group responsible for ONTOFUSION. The OntoDataClean tool uses ontologies to define the required transformations on data for various cleaning and integration purposes. This approach allows a more intuitive interaction with the own transformation process, allowing the specification of complex transformations on data more easily. The user can query a database through this tool, specifying the cleaning ontology to be used. The system will query the database and transform the data according to the ontology before presenting the results back to the user. The possible transformations that the ontology admits are these, listed below:

- Cleaning missing values: this transformation allows modifying or erasing records with missing data. Missing data is defined by value ranges or specific values. The data found to be missing can be either transformed or erased.
- Format cleaning: this transformation allows modifying the data type of specific columns, which can be a requirement for subsequent integration with other data.
- Scale cleaning: this transformation allows specifying algebraic transformations on numeric data. Arithmetic operators and basic functions are allowed, as well as using previous data values as variables for calculating the resulting values.

- Pattern cleaning: the pattern cleaning transformation allows modifying the pattern of string data. A powerful yet intuitive rule system is employed in order to accomplish this task (a rule is a composition of variables and constants). The user can define rules to identify the strings to be modified, and rules to define the resulting string, allowing to easily specify the transformation to a different date pattern. E.g., from mm-dd-yyyy to dd/mm/yy.
- Cleaning terminological inconsistencies: this transformation allows the replacement of words by preferred synonyms, given either in a specified dictionary, or explicitly.
- Duplicate cleaning: this transformation allows erasing records of data containing duplicate values in fields which are supposed to be unique (for example, employee_id).

Figure 36 shows the ontology used in ONTODATACLEAN transformations.

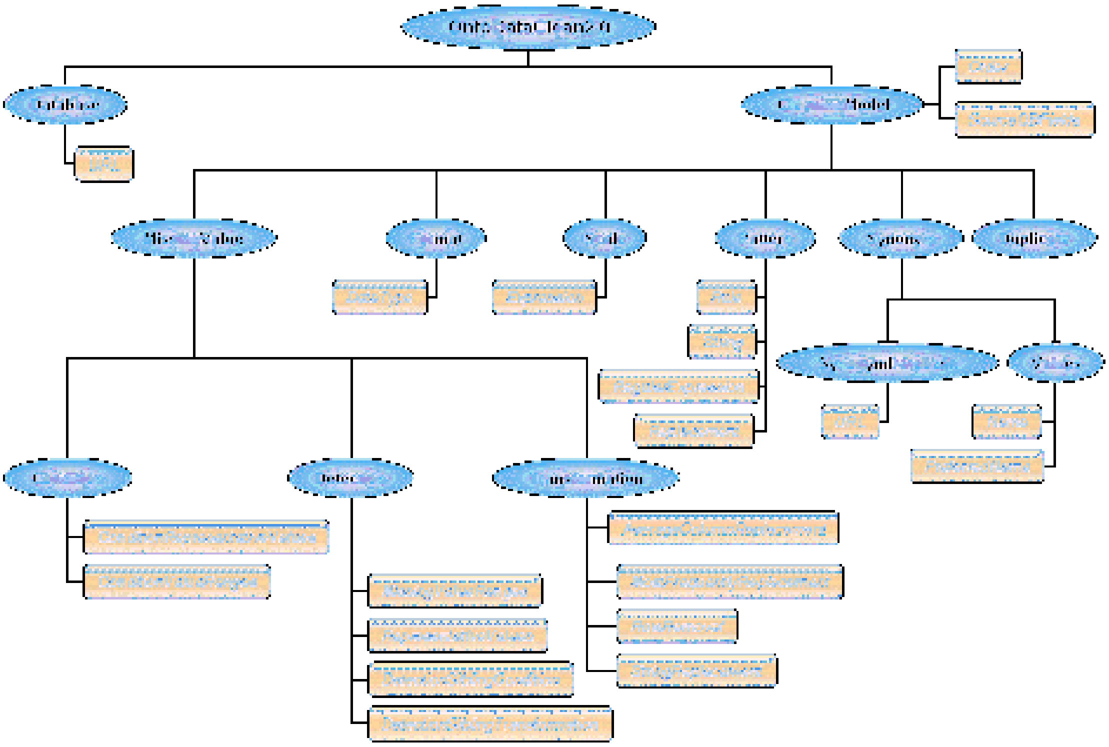


Figure 33: The cleaning ontology used by OntoDataClean

OntoDataClean also includes an extension tool that analyses databases in order to find possible inconsistencies that require cleaning. This analysis is based on statistical heuristics and can provide, to users, a helpful information before facing the specification of transformation tasks.

Further enhancements are projected for this tool, such as a wider range of transformations available, or a deeper analysis of existing inconsistencies.

3.2.5.6 Open Source Tools

3.2.5.6.1 KAON

KAON [BOZ2002] is an open source Tool suite that provides a multitude of software modules specially designed for the semantic web. It includes a persistent RDF store, an ontology store, ontology editors, etc. It has been developed as a result of a joint effort by the institute AIFB (University of Karlsruhe) and the Research Center of Information Technologies (FZI).

KAON offers an ontology management infrastructure, mainly targeted at business applications. It allows creating and managing ontologies easily and provides a framework aimed at building ontology-based applications.

KAON Reverse tool offers the possibility of mapping relational databases to ontologies, enabling two tasks: updating databases contents and performing queries through the conceptualization of a database. One drawback of this tool is that changes cannot be applied to the structure of the database with respect to the ontology, since the whole process should be repeated. This work is not reusable.

The kernel of this suite is the KAON SERVER, which brings all the software modules together. KAON SERVER is implemented with the Java programming language. The Java Management Extensions (JMX) are used to manage and monitor all the resources KAON handles. Figure 9 shows KAON architecture.

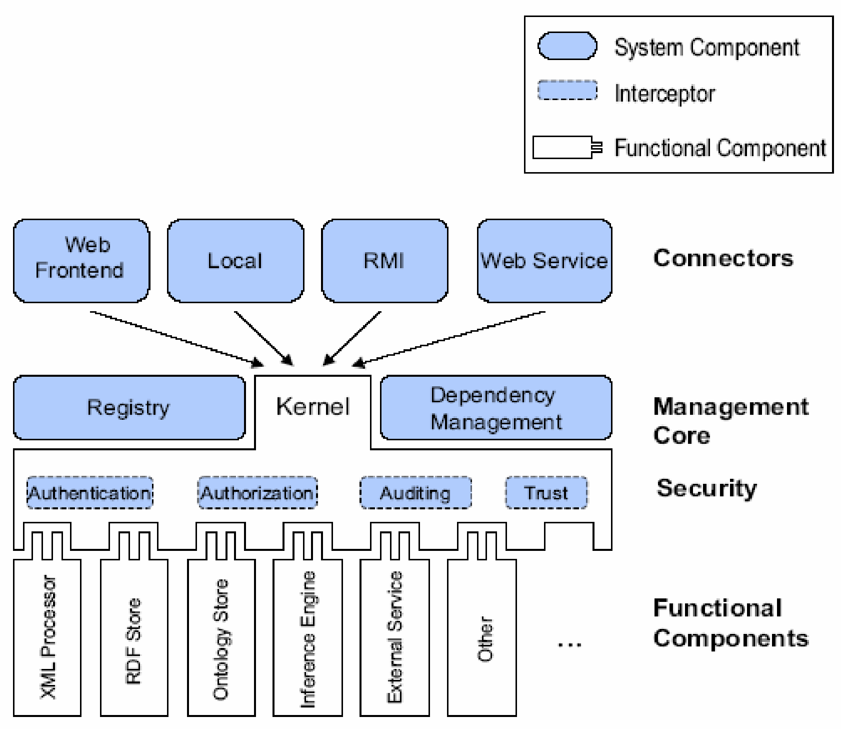


Figure 34: KAON SERVER Architecture

3.2.5.6.2 DR2 MAP

DR2 Map [BIZ2003] is a declarative and XML-based language. It allows describing mappings between relational database schemata and OWL/RDFS ontologies. With DR2, users can create flexible mappings of complex relational structures without having to change the existing database schema, which is achieved by applying SQL statements directly on the mapping rules.

The DR2 processor is responsible for the mapping process, which is performed in four logical steps:

1. A record set is selected from the database, based on class similarity.
2. The record set is grouped according to the groupBy columns.
3. Class instances are created.
4. The record set data is mapped to instance properties.

DR2 MAP is kept as simple as possible, expressing mappings with just three elements. Figure 38 shows the mapping process used in DR2 MAP.

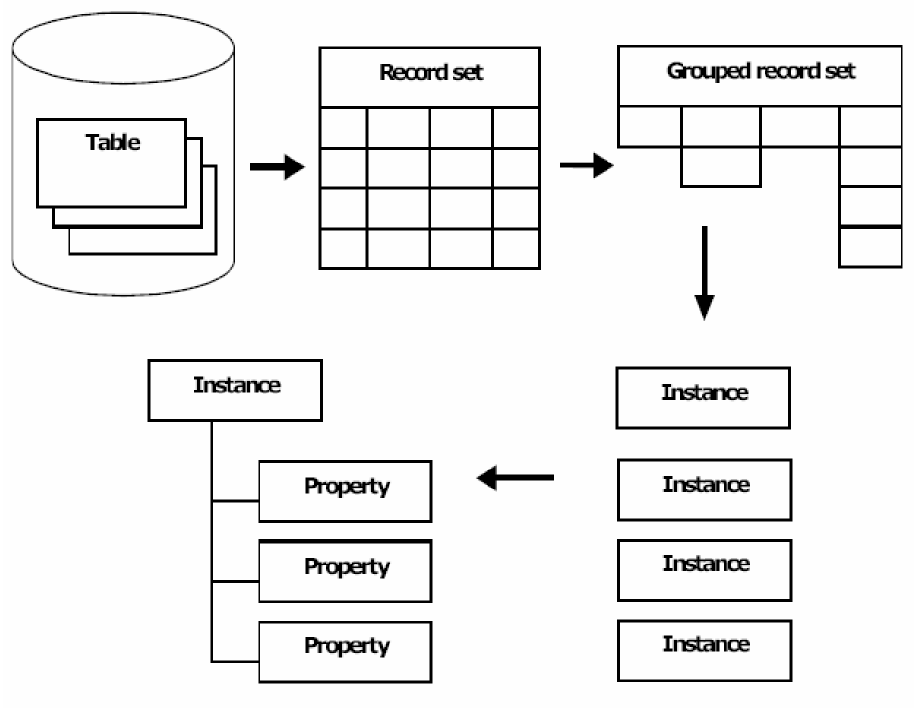


Figure 35: The DR2 mapping process

3.2.5.7 Comparison between Ontology-based Database Integration systems

In the following table it can be seen a comparison of features among several ontology-based integration systems:

	D2RMAP [1]	SEMEDA *	KAON Reverse *	INFOGENMED ONTOFUSION
Ontology Description Language	RDF	RDF	RDFS	DAML+OIL
OWL	YES	No	No	YES
Ontology Editor	No	YES	YES	YES
Ontology Graphical Browser	No	No	YES	YES
Public Database Integration	No	YES	No	YES
Physical Schema Re-design	No	No	No	YES
Virtual Schemata Unification	No	No	No	YES

3.2.5.8 References

[BIZ2003] Bizer C. “D2R MAP - A Database to RDF Mapping Language”. In Proceedings of the International World Wide Web Conference (WWW2003), Budapest, Hungary, 2003.

[BOZ2002] Bozsak E. et al. “KAON - Towards a Large Scale Semantic Web”. In K. Bauknecht, A. Min Tjoa, and G. Quirchmayr, editors, EC-Web 2002, volume 2455 of Lecture Notes in Computer Science, pages 304–313. Springer, September 2002.

[CRI1998] T. Critchlow, M. Ganesh, and R. Musick, "*Automatic generation of warehouse mediators using an ontology engine*," presented at the Proc. 5th KRDB Workshop, Seattle, WA, 1998.

[CRI1998] Critchlow, T., Ganesh, M., and Musick, R. 1998. Meta-Data Based Mediator Generation. In *Proceedings of the 3rd IFCIS international Conference on Cooperative information Systems* (August 20 - 22, 1998). COOPIS. IEEE Computer Society, Washington, DC, 168-176. 1998

[CRI2001] Critchlow, R. Musick, T. Slezak. Experiences applying meta-data to bioinformatics. *Inf. Sci.* 139, 1-2 (Nov. 2001), 3-17. 2001

[GRU1993] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications", Knowledge Acquisition, 5(2), 199-220, 1993

[HAK2001] Hakimpour, F. and Geppert, A. 2001. Resolving semantic heterogeneity in schema integration. In *Proceedings of the international Conference on Formal ontology in information Systems - Volume 2001* (Ogunquit, Maine, USA, October 17 - 19, 2001). FOIS '01. ACM Press, New York, NY, 297-308. 2001

[KIM1996] R. Kimball. *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*, John Wiley. 1996

[KOH2003] Jacob Kohler, Stephan Philippi and Matthias Lange. SEMEDA: ontology based semantic integration of biological databases, Bioinformatics, pp 2420--2427. 2003

[PER2006] D. Pérez-Rey, A. Anguita, V. Maojo. “OntoDataClean: Ontology-based Integration and Preprocessing of Biomedical Data”. Submitted to the VII International Symposium on Biological and Medical Data Analysis (ISBMDA 06). (Submitted)

[PER2005b] D. Pérez-Rey, V. Maojo, “Nuevo modelo basado en Ontologías para el KDD en Biomedicina”. TIC en Biomedicina. Colección Informática 15. ISBN 84-934497-3-3. pp. 157-176. Diciembre 2005.

[PER2004] D. Perez-Rey, V. Maojo, M. Garcia-Remesal, R. Alonso-Calvo, "Biomedical

Ontologies in Post-Genomic Information Systems," bibe, p. 207, Fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE'04), 2004.

[PER2005a] D. Pérez-Rey, V. Maojo, M. García-Remesal, R. Alonso-Calvo, H. Billhardt, F. Martin-Sánchez and A. Sousa, ONTOFUSION: Ontology-based integration of genomic and clinical databases, Computers in Biology and Medicine, In Press, Corrected Proof, Available online 6 September 2005

[SHE1990] A. P. Sheth e J. A. Larson. "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases", *ACM Computing Surveys*, 22(3): pp. 183-236. 1990

[VER2003] JL Verschelde, M Casella Dos Santos, T Deray, B Smith & W Ceusters Ref: Verschelde JL, Casella Dos Santos M, Deray T, Smith B and Ceusters W. Ontology-assisted database integration to support natural language processing and biomedical data-mining, Journal of Integrative Bioinformatics. 2003

[WIE1992] G. Wiederhold "Mediators in the Architecture of Future Information Systems", *IEEE Computer*, 25(3): pp. 38-49. 1992

3.3 BIOMEDICAL ONTOLOGIES

This document describes the commonly used biomedical ontologies, terminologies and databases (OTDs) for various purposes. The OTDs have various usages within the domain of biomedicine in general and in oncology and oncology-related biology in particular. The widest service which the OTDs provide is that of a good dictionary, where different classes, terms, entities are given unique identification codes and can be used in a way that they are univocal. Arguably this is the simplest service which OTDs can provide. Ability to draw inferences, relationships among entities at various levels of granularity, existential dependence, mereotopological formalizations etc. are the more advanced services which OTDs can provide. These services are used for life-science data integration, integration of Electronic Health Record data, patient status description, and drug delivery information provision in the domain of oncology. Specific features of these OTDs make them relevant for clinical practice in oncology and for oncology-related biomedical research.

3.3.1 GENERIC MEDICAL OTDS

3.3.1.1 Systematized Nomenclature of Medicine – Clinical Terms (Snomed CT)[2]

Developed by: College of American Pathologists & England and Wales National Health Service

Content: Snomed CT is a generic healthcare terminology together with various relations between it's over 300,000 concepts. There are about a million descriptions of those concepts and about a million semantic links between them. The Snomed CT core content consists of:

- Concepts Table
- Descriptions Table
- Relationship Table
- History Table
- ICD Mapping

Top Classes: The main top classes consist of Clinical Finding, Procedure, Observable Entity, Body Structure, Organism, Substance, Pharmaceutical/Biologic Product, Specimen and Events.

Attributes: Snomed CT classifies attributes according to the top classes. While some attributes are used across many top classes, there are many which are characteristically used within a single top class. For example, Clinical Finding top class is associated with attributes like Severity, Onset, Course, Episodicity, Stage and so on. Similar, for Procedure, the attributes include Procedure Site, Procedure Device, Procedure Morphology, Access and so on.

Availability: Snomed CT is available under license for the countries within the European Union.

[2] http://www.snomed.org/snomedct/

Tools: Clue-5[3] is a CIC Lookup engine for browsing SNOMED CT and for its integration with MS Windows-based clinical applications. The Clue-5 tool provides a reference and a browser server with an API for Snomed CT integration.

Relevance to Oncology: Since Snomed CT covers the generic medical domain, there are many areas where there are overlaps with the domain of carcinomas. In particular, the classification of procedures, medications and diseases are useful. Although Snomed CT also provides an anatomical classification, the FMA seems to be more useful for carcinomas. The advantage of using Snomed, as much as possible, is that the terms are connected together and come with unique IDs. However the problems with the classifications and relationship formalisms in Snomed could lead to some limitations in inference derivation.

3.3.1.2 Unified Medical Language System (UMLS)[4]

Developed by: National Library of Medicine

Content: UMLS consists of Metathesaurus, Semantic Network, SPECIALIST Lexicon and Metamorphosys.

- Metathesaurus is a vocabulary database of over a million terms dealing with the content of biomedical literature and Electronic Health Records. It consists of over 100 source vocabularies and tends to be univocal. When more than one meaning is assigned to a single vocabulary, then both the meanings of the term are represented within the Metathesaurus with the reference to specific source vocabularies. The source vocabularies integrated with the Metathesaurus includes ICD, Snomed, CPT codes, DSM, HUGO, MedDRA, NCI Thesaurus.
- The Semantic Network consists of # Semantic Types, that provide a consistent categorization of all concepts represented in the UMLS Metathesaurus a set of Semantic Relations, that exist between Semantic Types.
- The SPECIALIST Lexicon provides the lexical information needed for the SPECIALIST Natural Language Processing (NLP) System.

Availability: UMLS is available under license for the users within the European Union.

Tools: UMLS resources are used in informatics applications including information retrieval, natural language processing, creation of patient and research data, and the development of enterprise-wide vocabulary services. NLM's applications include PubMed, the NLM Gateway, ClinicalTrials.gov, and the Indexing Initiative. Other examples of UMLS-enabled applications include the National Cancer Institutes Enterprise Vocabulary Services and the Agency for Healthcare Research and Quality's National Guidelines Clearinghouse and National Quality Measures Clearinghouse. UMLS knowledge sources are distributed with flexible lexical tools and the MetamorphoSys install and customization program.

Relevance to Oncology: UMLS is a conglomerate where terms from over 100 OTDs can be queried for. The Metathesaurus has been extensively used for text mining and natural

[3] http://www.clininfo.co.uk/clue5/

[4] http://umlsinfo.nlm.nih.gov/

language processing in biomedical domain and thus is relevant for carcinomas. The UMLS Semantic Network and the Metathesaurus are not formalized ontologies, however, recently efforts are being made to formalize the Semantic Network in a way that inferences can be made based on it. UMLS has also been used to for mutant protein term identification from the natural text, something which helps in a semiautomatic extension of the existing mutant protein databases.

3.3.1.3 Generalized Architecture for Languages, Encyclopaedias and Nomenclatures in medicine (GALEN)[5]

Developed by: GALEN and related European Union Project Participants

Content: The GALEN project developed a Common Reference Model, a clinical terminology which can be applied to various medical domains. The GALEN project established the ontology and GRAIL formalism and demonstrated the feasibility of the concepts. GALEN-IN-USE developed the Common Reference Model (CRM) for Medical Procedures —a key element for architectures for interworking between medical records, decision support, information retrieval and natural language processing systems in healthcare. OpenGALEN was established in 1999 as a not-for-profit organisation to provide information on GALEN technologies and relevant software distributors and, in particular, to maintain and disseminate the CRM.

Availability: OpenGALEN is available for free use within the European Union within the terms of its license.

Tools: Common Reference Model; GALEN Representation and Integration Language (GRAIL); Knowledge Management Environment (OpenKnoME); GALEN Case Environment

Relevance to Oncology: Similar to Snomed CT, GALEN can be embedded as a generic clinical terminology with extensions for carcinomas. GALEN is better formalized compared to the other generic medical OTDs and using GRAIL, various kinds of inferences can be derived.

3.3.2 SPECIFIC MEDICAL OTDS

3.3.2.1 Foundational Model of Anatomy (FMA)[6]

Developed by: Structural Informatics Group, University of Washington.

Content: FMA is concerned with the representation of classes and relationships necessary for the symbolic representation of the structure of the human body in a form that is understandable to humans and is also navigable by machine-based systems. Specifically, the FMA is a domain ontology that represents a coherent body of explicit declarative knowledge about human anatomy. FMA has four interrelated components:

[5] http://www.opengalen.org/

[6] http://sig.biostr.washington.edu/projects/fm/AboutFM.html

Anatomy taxonomy: classifies anatomical entities according to the characteristics they share and by which they can be distinguished from one another.

Anatomical Structural Abstraction: specifies the part-whole and spatial relationships that exist between the entities represented in the taxonomy

Anatomical Transformation Abstraction: specifies the morphological transformation of the entities represented in the taxonomy during prenatal development and the postnatal life cycle

Metaknowledge: specifies the principles, rules and definitions according to which classes and relationships in the other three components of FMA are represented.

FMA contains approximately 72,000 classes, over 115,000 terms and over 2.1 million relationship instances from 168 relationship types.

Availability: FMA is available for free use within the European Union within the terms of its license. A contract must be individually signed and a download access asked for.

Tools: Foundational Model Explorer is an internet based FMA browser. FMA also allows StruQL queries which provide XML as output.

Relevance to Oncology: FMA is very useful while representing anatomical entities in Relevance to Oncology. These include carcinoma staging, locations for radiotherapy and surgery, access routes for various procedures, locations for drug actions, and so on. The robust formalism allows to derivation of inferences, especially for staging of carcinomas.

3.3.2.2 NCI Thesaurus[7]

Developed by: National Cancer Institute

Content: The NCI Thesaurus is an ontology-like vocabulary that includes broad coverage of the cancer domain, including cancer related diseases, findings and abnormalities; anatomy; agents, drugs and chemicals; genes and gene products and so on. In certain areas, like cancer diseases and combination chemotherapies, it provides the most granular and consistent terminology available. It combines terminology from numerous cancer research related domains, and provides a way to integrate or link these kinds of information together through semantic relationships. The Thesaurus currently contains over 34,000 concepts, structured into 20 taxonomic trees.

Availability: NCI Thesaurus is available for free use within the European Union within the terms of its license.

Tools: NCI Thesaurus browser is maintained by the NCI.

Relevance to Oncology: The terminology of NCIT has been built to deal with the specific domain of carcinomas and therefore it does play an important role as a common dictionary of terms used by specialists from different domains while dealing with carcinomas. Its over-

[7] http://nciterms.nci.nih.gov/NCIBrowser/Dictionary.do

reliance on the UMLS and in particular its semantic network and some otherwise inherent problems with the classification within NCIT leads to some limitations in inference derivations; however, the NCIT does play a very useful role as a common carcinoma terminology.

3.3.2.3 International Classification of Diseases (ICD)[8]

Developed by: World Health Organization

Content: ICD is designed to promote international comparability in the collection, processing, classification, and presentation of diagnostics in health epidemiology, health management and mortality statistics. These include the analysis of the general health situation of population groups and monitoring of the incidence and prevalence of diseases and other health problems in relation to other variables such as the characteristics and circumstances of the individuals affected. The top classes consist main of diseases classified according to the body system, though neoplasms, infectious diseases and injuries and poisonings have their own axes.

Availability: ICD is available for free use within the European Union within the terms of its license.

Tools: ICD browser is provided by the WHO. Many other browsers in different languages exist online.

Relevance to Oncology: To a lot of extent, ICD provides a disease classification on the basis of anatomy. Although not all the diseases within ICD are classified according to anatomy, the neoplasms are more or less classified within the anatomical partition. Thus, an ontology of carcinomas which follows the anatomical partition for classification of neoplasms and related diseases can use portions of ICD more easily than other disease classifications. However there are issues of misclassifications within ICD and also terms which do not represent a real disease. With certain modifications, integration of ICD with FMA related anatomy is possible in a way that inferences can be drawn on the basis of the anatomy ontology of FMA.

3.3.2.4 International Classification of Functioning, Disability and Health (ICF)[9]

Developed by: World Health Organization

Content: ICF is a classification of health and health related domains that describe body functions and structures, activities and participation. The domains are classified from body, individual and societal perspectives. Since an individual's functioning and disability occurs in a context, ICF also includes a list of environmental factors. The top classes of ICF are: Body Functions, Body Structures, Activities and Participation and Environmental Factors. Thus ICF provides terminology not just for functions, disability and Environmental factors, but also for

[8] http://www.who.int/classifications/icd/

[9] http://www3.who.int/icf/icftemplate.cfm

the body structures, although they are not formalized and detailed like other ontologies e.g. FMA.

Availability: ICF is available for free use within the European Union within the terms of its license.

Tools: ICD browser is provided by the WHO.

Relevance to Oncology: The classification of functioning and disability is useful to code patient status before and after therapy and also during the rehabilitation. ICF does provide a terminology which is useful for coding, however the classification is primitive and the relations between classes belonging to different axes does not exist. ICF's connection with ICD would improve the usage of both these terminologies.

3.3.2.5 Logical Observation Identifiers Names and Codes (LOINC)[10]

Developed by: The Regenstrief Institute and the LOINC committee.

Content: LOINC is a terminology primarily for laboratory results and also covers certain kinds of clinical observations. It contains over 40,000 terms out of which over 30,000 deal with the laboratory domain. The laboratory portion of the LOINC database contains the usual categories of chemistry, hematology, serology, microbiology (including parasitology and virology), and toxicology; as well as categories for drugs, the cell counts and antibiotic susceptibility. The clinical portion of the LOINC database includes entries for vital signs, hemodynamics, intake/output, EKG, obstetric ultrasound, cardiac echo, urologic imaging, gastroendoscopic procedures, pulmonary ventilator management, selected survey instruments, and other clinical observations.

Availability: LOINC is available for free use within the European Union within the terms of its license.

Tools: Windows-based mapping utility called the Regenstrief LOINC Mapping Assistant (RELMA)[11] facilitates searches through the LOINC database and to assist efforts to map local codes to LOINC codes. Like the LOINC database, this program is also available for free use.

Relevance to Oncology: The LOINC database provides a terminology source which is widely used in all aspects of healthcare and thus is also widely used with the domain of carcinomas, especially in the English-speaking countries. The connection between LOINC codes and certain EHR architectures increase its usage. Most of the specific laboratory tests which are useful for carcinomas are covered within LOINC. A lack of formal classification, a formal mechanism of term post-coordination and relations between various classes are known issues but LOINC tends to a database and not a full-fledged ontology and elaborates the necessary attributes for various laboratory tests and procedures in detail.

[10] http://www.regenstrief.org/loinc/

[11] http://www.regenstrief.org/loinc/relma/

3.3.2.6 Medical Subjects Headings (MeSH)[12]

Developed by: National Library of Medicine

Content: MeSH is a controlled vocabulary thesaurus consisting of sets of terms naming descriptors in a hierarchical structure that permits searching at various levels of specificity. The top-level classification includes: Anatomy, Organisms, Diseases, Chemicals and Drugs, Analytical, Diagnostic and Therapeutic Techniques and Equipment, Psychiatry and Psychology, Biological Sciences, Physical Sciences, and so on. MeSH is used on MEDLINE to index bibliographic citations and author abstracts from over 4,000 journals.

Availability: MeSH is available for free use within the European Union within the terms of its license.

Tools: MeSH Browser provides a searchable GUI for MeSH terms. PubMed uses MeSH as its terminology to search journal articles. HONSelect is a multilingual search tool which uses MeSH to link to various healthcare-related websites.

Relevance to Oncology: MeSH is useful for the carcinoma domain due to its usage within PubMed. All major carcinoma literature is classified within PubMed and is available to retrieval using the MeSH coding. Like many other OTDs, MeSH does not claim to be a full ontology and not all its axes are as complete in terms are others. However its usage within PubMed is wide and has been widely embraced in many text-mining systems.

3.3.2.7 Medical Dictionary for Regulatory Activities (MedDRA)[13]

Developed by: Developed by the International Conference on Harmonisation (ICH). It is owned by the International Federation of Pharmaceutical Manufacturers and Associations (IFPMA) acting as trustee for the ICH steering committee. Maintained by MSSO - Maintenance and Support Services Organization.

Content: MedDRA is a terminology for drug and medical device side-effects and malfunctions. It emphasizes on ease of use for data entry, retrieval, analysis, and display when dealing with registering, documenting, and safety monitoring of medical products. The top-level classification of MedDRA consists mainly of disorders classified according to various body systems: Respiratory disorders, Cardiac disorders, Gastrointestinal disorders, Immune system disorders, Endocrine disorders, and so on.

Availability: Annual subscription fee required for use within the European Union.

Tools: MedDRA browser comes with the license agreement.

Relevance to Oncology: MedDRA is used to code drug and medical device-side effects in all the medical domains and thus is also used for management of carcinomas. As far as the terminology is concerned, MedDRA tends to cover quite a generic domain similar to Snomed CT or UMLS. However the kinds of issues regarding the classification are similar to other OTDs. Not all the classes follow a classification on the basis of anatomy and this integration

[12] http://www.nlm.nih.gov/mesh/

[13] http://www.meddramsso.com/MSSOWeb/index.htm

of MedDRA with anatomy ontologies needs reclassification and introduction of new classes. Such an effort is needed in order to improve the derivation of inferences.

3.3.2.8 National Drug Code Directory[14]

Developed by: Food and Drug Administration (FDA)

Content: The Drug Listing Act of 1972 requires registered drug establishments to provide the FDA with a current list of all drugs manufactured, prepared, propagated, compounded, or processed by it for commercial distribution. Drug products are identified and reported using a unique, three-segment number, called the National Drug Code (NDC), which is a universal product identifier for human drugs. FDA inputs the full NDC number and the information submitted as part of the listing process into a database known as the Drug Registration and Listing System (DRLS). Several times a year, FDA extracts some of the information from the DRLS data base for publication in the NDC Directory.

Availability: NDC is available for free use within the European Union within the terms of its license.

Tools: No specific publicly available tools provided.

Relevance to Oncology: The usage of NDC is mandatory for coding related to medications and this applies to all the medical domains and thus is applicable to carcinomas. Although NDC usage is mandated only within the USA, many other countries have based their requirements in lines with what is proposed by NDC. Moreover, since most of the major Hospital Information Systems and Drug Databases are NDC compliant, these codes are embedded in systems used almost everywhere in the world. NDC is not an ontology and provides a very limited set of information regarding medication and for chemotherapy agents used in carcinomas these tend to be particularly deficient. An extension of NDC is possible and is implemented within various systems.

3.3.2.9 Online Mendelian Inheritance in Man (OMIM)[15]

Developed by: John Hopkins University and National Center for Biotechnology Information

Content: OMIM is a catalog of human genes and genetic disorders together with textual information and references. It illustrates the genes which have been associated with a particular disease in literature. OMIM focuses primarily on inherited or heritable, genetic diseases. It is also considered to be a phenotypic companion to the human genome project and is based upon the text Mendelian Inheritance in Man. Each entry is given a unique six-digit number whose first digit indicates the mode of inheritance of the gene involved

Availability: OMIM is available for free use within the European Union within the terms of its license.

[14] http://www.fda.gov/cder/ndc/

[15] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM

Tools: A searchable browser with basic and advanced functions is provided. The OMIM Gene Map presents the cytogenetic locations of genes that are described in OMIM. It is a single file, presented in tabular format, listing genes from the p telomere of chromosome 1 through the q telomere of chromosome 22, followed by genes on the X and Y chromosomes. The OMIM Morbid Map is an alphabetical list of diseases described in OMIM and their corresponding cytogenetic locations.

Relevance to Oncology: The connection between gene abnormalities and diseases is useful for almost all the diseases present within OMIM. However it is especially important for hereditary diseases and carcinomas. The sheer number of genetic abnormalities associated with carcinomas is the evidence that such associations are related to the various protein and pathway abnormalities forming a part of the pathologies within carcinomas.

3.3.2.10 International Classification of Nursing Practice (ICNP)[16]

Developed by: International Council of Nurses

Content: ICNP is a terminology which facilitates description and comparison of nursing practice. ICNP had three axes to begin with:

- Nursing phenomena (nursing diagnoses)
- Nursing actions
- Nursing outcomes

However recently the ICNP version 1 has evolved from the beta versions and consists of only one root axis – Nursing Phenomena, which in turn has seven axes – Client, Focus, Location, Judgment, Means, Time, and Action.

Availability: ICNP is available for free use within the European Union within the terms of its license.

Tools: Searchable browser provided by the ICN.

Relevance to Oncology: The nursing diagnoses, actions and procedures are important to every medical domain and therefore also to carcinomas. Given the large number of processes involved in management of cancer patients and the criticality of many of those procedures, a common coding system for nursing, ICNP, is of high usage. ICNP does not claim to be an ontology and the formalism behind the classification and relationships between the classes could be further improved.

[16] http://www.icn.ch/icnp.htm

3.3.3 GENE ANNOTATION OTDS

3.3.3.1 Gene Ontology (GO)17 and Gene Ontology Annotation (GOA)[18]

Developed by: GO is developed by the Gene Ontology Consortium, of which GOA@EBI is also a part of. GOA is developed by GOA@EBI group (European Bioinformatics Institute).

Content: The GO project is a collaborative effort to address the need for consistent descriptions of gene products in different databases. The GO project has developed three structured, controlled vocabularies (ontologies) that describe gene products in terms of their associated **biological processes**, **cellular components** and **molecular functions** in a species-independent manner. As of June 2006, GO contains 19861 terms of which 95.5% have definitions with10690 belonging to the biological process axis, 1740 to the cellular component axis and 7431 to the molecular function. Currently, GO has only is-a and part-of relations between terms belonging to a particular axis.

GOA provides assignments of gene products to the Gene Ontology (GO) resource. UniProtKB/Swiss-Prot has joined the Gene Ontology (GO) Consortium and has adopted its standard vocabulary to characterise the activities of proteins in the UniProtKB/Swiss-Prot, UniProtKB/TrEMBL and InterPro databases. It has initiated the GOA project to provide assignments of GO terms to gene products for all organisms with completely sequenced genomes by a combination of electronic assignment and manual annotation.

Availability: GO and GOA are available for free use within the European Union within the terms of its license.

Tools: There are a wide plethora of tools built around GO, some by the GO consortium and many out the consortium.

- The consortium tools consist of: AmiGO, a browser allowing search for a GO term and view all gene products annotated to it, or search for a gene product and view all its associations and OBO-Edit, an open source, platform-independent graph-based application for viewing and editing OBO ontologies.
- Non-Consortium tools for searching and browsing GO include: CGAP GO Browser, COBrA, EP GO Browser, GeneInfoViz, GeneOntology at RZPD, GenNav, GOblet, GoFish, MGI GO Browser, QuickGO at EBI, PANDORA, TAIR Keyword Browser, Tk-GO. Tools for annotation include GeneTools, GoAnnotator, GoFigure, GoPubMed, GOtcha, HT-GO-FAT, InGOt, JAFA, Manatee and PubSearch.
- Non-consortium tools for gene expression and microarray analysis include: BiNGO, CLENCH, DAVID, EASE, eGOn v2.0, ermineJ, FatiGO, FuncAssociate, FuncExpression, GARBAN, GeneMerge, GFINDer: Genome Function, GOArray, GOdist, GOHyperGAll, GoMiner and MatchMiner, GOODIES, GOstat, GoSurfer, GO Term Finder, GOTM (Gene Ontology Tree Machine), GOToolBox, L2L, Machaon Clustering and Validation Environment, MAPPFinder, NetAffx Gene Ontology Mining Tool, Onto-Compare, Onto-Design, Onto-Express, Onto-Miner, Onto-Translate, OntoGate, Ontologizer, Ontology Traverser, Probe Explorer, SeqExpress, SOURCE, STEM: Short Time-series Expression Miner, THEA, Avadis - gene expression analysis with GO browser and Spotfire Gene Ontology Advantage Application.

[17] http://www.geneontology.org/

[18] http://www.ebi.ac.uk/GOA/

Relevance to Oncology: GO and GOA provide annotations to various gene products which are directly associated with carcinomas. The mapping of those gene products to entities within Uniprot and pathway databases and that to OMIM further close the loop by which the various functions and effects of those gene products can be queried. GO terms themselves provide a rather primitive collection of relations between the classes. However the annotations to those terms and the subsumption relationships help provide certain kinds of inferences. Despite being called an ontology, GO is far from being a formal ontology.

3.3.4 PROTEIN OTDs

3.3.4.1 Universal Protein Resource (UniProt)[19]

Developed by:

Content: UniProt is a central repository of protein sequence and function created by joining the information contained in Swiss-Prot, TrEMBL, and PIR.

Availability: Uniprot is available for free use within the European Union within the terms of its license.

Tools: UniProt Reference Clusters (UniRef) databases combine closely related sequences into a single record to speed searches. UniProt Archive (UniParc) is a repository with the history of all protein sequences.

Relevance to Oncology: All the sources of Uniprot provide mutant protein databases with annotation to the diseases they are associated with. The number of mutant proteins associated with carcinomas form one of the largest portion of mutant protein databases. Together with the various links to DNA and RNA databases, pathways and to biomedical literature references, Uniprot plays an important in bridging together the gap between biological and medical information related to carcinomas.

3.3.4.2 Structural Classification of Proteins (SCOP)[20]

Developed by: MRC Centre for Protein Engineering, Cambridge, UK

Content: SCOP database, created by manual inspection and abetted by a battery of automated methods, provides a detailed description of the structural and evolutionary relationships between all proteins whose structure is known. It provides a broad survey of all known protein folds and detailed information about the close relatives of any particular protein. As of June 2006, SCOP has 25973 PDB Entries. The top hierarchy of SCOP includes alpha proteins, beta proteins, small proteins, multi-domain proteins, membrane and cell surface proteins, coiled coil proteins and peptides.

[19] http://www.pir2.uniprot.org/

[20] http://scop.mrc-lmb.cam.ac.uk/scop/

Availability: SCOP is available for free use within the European Union within the terms of its license.

Tools: SCOP browser is provided by the MRC group with pictographic representation. There are many tools developed by other groups which use SCOP database as their major inputs. There include: Structural similarity search of SCOP using SSM, Combinatorial Extension (CE) method for structural comparison, PALI pairwise and multiple alignments of SCOP families, SUPFAM structure/sequence relationships, Structural similarity search of SCOP using 3dSearch, Structural alignment of SCOP sequences (database + server), PINTS - Patterns In Non-homologous Tertiary Structures, Sequence similarity search of SCOP using FPS, CATH structural classification, Dali structural comparison and FSSP structural classification, PDB at a Glance, and 3Dee Protein Domain Definitions.

Relevance to Oncology: SCOP provides structures of proteins on the basis of which many different protein mutant structures have been predicted. The location and depiction of various functional groups within proteins helps provide the structure-functional relationship for normal proteins which malfunction and of mutant proteins. Given the large number of mutant proteins associated with carcinomas, this information is widely applied in determination of various pathways related to carcinomas.

3.3.5 PATHWAY AND INTERACTION OTDs

3.3.5.1 IntAct[21]

Developed by: Proteomics Services Team, European Bioinformatics Institute

Content: IntAct provides a database for protein interaction data derived from literature curation or direct user submissions. IntAct also incorporates the information within interaction databases like Database of Interacting Proteins (DIP) and Biomolecular Interaction Network Database (BIND).

Availability: IntAct is available for free use within the European Union within the terms of its license.

Tools: The tools developed as a part of the IntAct project include:

- ProViz: graph visualization system
- Targets: Predicts targets for pull-down experiments
- MiNe: Computes minimal connecting network for protein sets

Relevance to Oncology: The protein-protein interaction play an important role in the representation of various pathways associated with carcinomas. It not only tells about the malfunctioning related to carcinomas but also throws light on the various kinds of structural configurations of mutant proteins. Such information is also useful for drug development where agents targeting particular kinds of interactions and functional groups can be produced leading to increased efficacy and reduced adverse effects.

[21] http://www.ebi.ac.uk/intact/index.jsp

3.3.5.2 Reactome[22]

Developed by: Cold Spring Harbor Laboratory, European Bioinformatics Institute, Gene Ontology Consortium

Content: Reactome is a curated resource of core pathways and reactions in human biology. In addition to curated human events, inferred orthologous events in 21 non-human species including mouse, rat, chicken, fugu fish, worms, fly, yeast and E.coli are also available. The main pathways represented within Reactome include:

Apoptosis, Checkpoints, Mitotic Cell Cycle, DNA Repair, DNA Replication, Electron Transport Chain, Gene Expression, Hemostasis, HIV Infection, Hs Influenza Infection, Immune System Signaling pathways, Insulin receptor mediated signaling, Integration of pathways involved in energy metabolism, Lipid metabolism, Metabolism of amino acids and related nitrogen-containing molecules, Metabolism of glucose, other sugars, and ethanol, Notch Signaling Pathway, Nucleotide metabolism, Oxidative decarboxylation of pyruvate and TCA cycle, Post-translational modification of proteins, TGF-beta signaling pathway, Transcription, Translation, mRNA Processing.

Availability: Reactome is available for free use within the European Union within the terms of its license.

Tools: A browsable version with explanations in detail of all the steps is provided.

Relevance to Oncology: Reactome's emphasis on pathways related to transcription and translation and to receptor communication covers a lot of turf as far as processes related to carcinomas are concerned. Pathologies behind the initiation and spread of carcinomas involve some processes which are completely absent within the normal human body. However, a majority of the processes involved in carcinomas are those which are present within the normal human body and are either abnormally regulated, or over- or under-executed or take place at abnormal locations or time. The gene expression data from carcinomatous structures can be matched with respect to the expressions of various gene products.

3.3.5.3 Kyoto Encyclopedia of Genes and Genomes (KEGG)[23]

Developed by: Bioinformatics Center, the Institute for Chemical Research, Kyoto University

Content: KEGG is a suite of databases and associated software, integrating the function and utility of biological systems (PATHWAY and BRITE databases), genes and proteins (GENES database), and chemical compounds and reactions (LIGAND database). The PATHWAY database covers 37,869 pathways generated from 301 reference pathways, over a million genes in their GENES database and over 14000 compounds in their LIGAND Database. The main pathways covered include:

Metabolism (Carbohydrate, Energy, Lipid, Nucleotide, Amino acid, Glycan, PK/NRP, Cofactor/vitamin, Secondary metabolite, Xenobiotics), Genetic Information Processing,

[22] http://www.reactome.org/

[23] http://www.genome.jp/kegg/

Environmental Information Processing, Cellular Processes, Human Diseases and Drug Development.

Availability: KEGG is available for free usage within the European Union within the terms of licensing.

Tools: KEGG provides a searchable browser together with a pictographic representation of the various pathways.

Relevance to Oncology: Like Reactome, KEGG plays an important role in oncology research. The PATHWAY database provides information relevant to the pathological processes involved in carcinoma initiation and development. Apart from the pathway-related information, KEGG also provides information on carcinoma-relevant genes and proteins with their mutant variants.

3.3.6 DNA OTDs

3.3.6.1 Human Genome Project (HGP)[24]

Developed by: Human Genome Project Consortium

Content: Begun formally in 1990, the U.S. Human Genome Project was a 13-year effort coordinated by the U.S. Department of Energy and the National Institutes of Health. The project originally was planned to last 15 years, but rapid technological advances accelerated the completion date to 2003. Project goals were to

- identify all the approximately 20,000-25,000 genes in human DNA,
- determine the sequences of the 3 billion chemical base pairs that make up human DNA,
- store this information in databases,
- improve tools for data analysis,
- transfer related technologies to the private sector, and
- address the ethical, legal, and social issues (ELSI) that may arise from the project.

Availability: HGP sequences are available for free usage within the European Union within the terms of licensing.

Tools: Numerous projects have been spun off from the original HGP and has led to development of thousands of tools including those for analysis of gene expression, structure-function relations, transcription and translation simulators, public curation platforms like GenePoint and so on.

Relevance to Oncology: DNA structure, active zones, regulation and associated RNA transcription – all form the fundamentals of what gets coded into proteins. Given that mutant proteins and normal proteins behaving abnormally play an essential role in carcinoma initiation, development and spread, information coded within DNA molecules form the core of almost all pathologies associated with carcinomas.

[24] http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml

3.3.7 RNA OTDs

3.3.7.1 RNA Structure Database (RNABase)[25]

Developed by:

Content: RNABase is a database providing information regarding RNAs, especially its 3-dimensional structure in Cartesian coordinates. It uses a language of its own to define the dihedral angles of the various RNA bonds, which together provide the complete structure.

Availability: RNABase is available for free usage within the European Union within the terms of licensing.

Tools: A searchable browser is provided.

Relevance to Oncology: RNABase, similar to its DNA and protein counterparts, provide a possibility for structure-function comparisons. The transcription-translation process from DNA to proteins produces the catalysts and most of the participants for the biological pathways, which make them relevant for every pathological process, including those involved in the initiation and spread of carcinomas.

3.3.7.2 European Ribosomal RNA Database[26]

Developed by: European rRNA database was developed at the University of Antwerp, Belgium and since 2002 is maintained by the University of Ghent.

Content: European rRNA database compiles all complete or nearly complete SSU (small subunit) and LSU (large subunit) ribosomal RNA sequences. Sequences are provided in aligned format. The alignment takes into account the secondary structure information derived by comparative sequence analysis of thousands of sequences. Additional information such as literature references, taxonomy, secondary structure models and nucleotide variability maps, is also available.

Availability: European rRNA database is available for free usage within the European Union within the terms of licensing.

Tools: A BLASTable version of database is provided.

Relevance to Oncology: Ribosomal RNAs play an essential role within the process of translation which generates amino acid chains from the messenger RNA code. rRNAs are being actively researched for drug development in various clinical domains including oncology.

[25] http://www.rnabase.org/

[26] http://www.psb.ugent.be/rRNA/

3.3.8 SINGLE NUCLEOTIDE POLYMORPHISM (SNP) OTDs

3.3.8.1 NIH Single Nucleotide Polymorphism Database (dbSNP)[27]

Developed by: National Center for Biotechnology Information (NCBI)

Content: SNP stands for "single nucleotide polymorphism". SNPs are the most common genetic variations and occur once every 100 to 300 bases. A key aspect of research in genetics is the association of sequence variation with heritable phenotypes. It is expected that SNPs will accelerate the identification of disease genes by allowing researchers to look for associations between a disease and specific differences (SNPs) in a population. This differs from the more typical approach of pedigree analysis which tracks transmission of a disease through a family. It is much easier to obtain DNA samples from a random set of individuals in a population than it is to obtain them from every member of a family over several generations. Once discovered, these polymorphisms can be used by additional laboratories, using the sequence information around the polymorphism and the specific experimental conditions. The Single Nucleotide Polymorphism database (dbSNP) is a public-domain archive for a broad collection of simple genetic polymorphisms. As of June 2006, dbSNP contains over 12 million Human RefSNP clusters, over 6 million Mus musculus RefSNP clusters and over 3 million Canis familiaris and Gallus gallus clusters.

Availability: dbSNP is available for free usage within the European Union within the terms of licensing.

Tools: A searchable browser is provided through Entrez.

Relevance to Oncology: In the last few years, SNPs have gained a lot of importance in clinical research. The database information is compared to gene expression information of many carcinomas. Multispecies database allows comparison across different species and also make results from animal models comparable to the human case. SNPs are being widely used in chemotherapy drug development targeted against specific mutant proteins or protein complexes. Recently SNPs have also been applied for clinical research in radiotherapy.

3.3.8.2 Japanese Single Nucleotide Polymorphism (JSNP) Database[28]

Developed by: JSNP is developed by Human Genome Center, Institute of Medical Science, The University of Tokyo and Japan Science and Technology Agency.

Content: JSNP is the database for DNA sequence variations, polymorphic markers to investigate genes susceptible to diseases or those related to drug responsiveness. The 28th data release consists of 197,157 SNPs and 84,612 SNPs with allele frequency. SNPs will also be deposited in the public dbSNP and HGVbase (previously known as HGBASE) under the bi-directional data exchange between dbSNP and HGVbase.

[27] http://www.ncbi.nlm.nih.gov/projects/SNP/

[28] http://snp.ims.u-tokyo.ac.jp/

Availability: JSNP is available for free usage within the European Union within the terms of licensing.

Tools: A searchable browser enabling gene search is provided by the developers.

Relevance to Oncology: Similar to dbSNP, JSNP database information is useful for gene expression studies and drug development.

3.4 ONTOLOGIES AND IMAGES

3.4.1 INTRODUCTION

This chapter presents a survey of the most relevant methods and models used for applying ontologies to image processing. This issue is potentially relevant for ACGT, to link WP7 with other WPs where images must be used or processed. This chapter is divided in three sections: (2) *Methods and Models, (3) Glossary and (4) Bibliography*. The reader will found that section (2) is divided into two classes: *Ontology-Based Image Annotation* and *Ontologies in Content-Based Image Retrieval.* These two are the main research lines in which ontologies have been applied in relation with images.

3.4.2 METHODS & MODELS

3.4.2.1 Ontology-based Image Annotation

3.4.2.1.1 Introduction

Text based retrieval and classification of documents is a problem that has been approached with very good results. This is not the case with images. The information contained in an image that can be easily extracted, concerns mainly to low level features, such as colours and primitive shapes. It is a hard task to relate this kind of low level information with high level semantic concepts that could, for example, describe a class, or be present in a query.

Image annotation approaches the task of attaching text to an image that describes it in different ways, making easier to find it within a database or to classify it. With good annotations, text retrieval and classification techniques can be used to manage large image databases (there is no need of actual image retrieval). Unfortunately, good annotations have to be made by humans. This task is very difficult to be addressed when large image sets need to be annotated.

Without the proper guide, human annotations of images can be a complex and difficult task. Ontologies have been used to assist human annotation of images. An ontology introduces standard vocabularies, and, because of their richness in terms of structuring of knowledge, can provide richer level of annotation as well.

This section shows different methods and tools that have exploited ontologies to aid the process of image annotation in different domains. All these methods and approaches are ordered in an inverse chronological sort, in order to keep the newest on the top. Readers of this deliverable may notice that newer methods are usually based on older ones.

3.4.2.1.2 Methods, Approaches and Tools in Ontology-based Image Annotation

In this section, the reader can find a summary the latest methods, approaches and tools that make use of ontologies to aid the task of annotating images.

3.4.2.1.2.1 Silva et al's Method [CHA1999]

In this work, two levels of annotation are proposed:

- The first one allows the selection of atomic concepts that the user identifies in the image

- The second level captures abstract concepts that an expert identifies belonging to the domain.

An ontology can contribute to make the knowledge of the domain (which the image belongs to) explicit.

A knowledge-based model to describe visual features according to different annotations strategies and levels of granularity is introduced. In this work two levels of granularity are proposed, which require different annotation approaches. There exists a mapping between these approaches to present an ontology-based annotation framework, with which different users (from novices to experts) can refine the annotations.

In Low-Level feature annotation, the user browses the concepts, attributes and values from a given ontology on the domain. Multiple image descriptions can be produced according to different task requirements. Figure 54 shows how rock features from an Oil-Reservoir ontology are associated to an image.

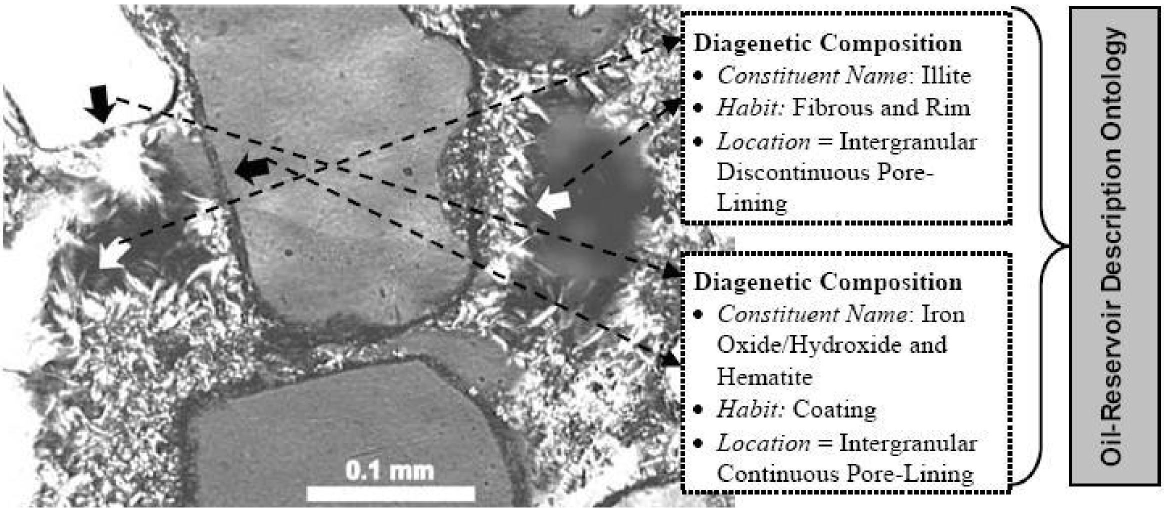

Diagenetic Composition
• Constituent Name: Illite
• Habit: Fibrous and Rim
• Location = Intergranular Discontinuous Pore-Lining
Diagenetic Composition
• Constituent Name: Iron Oxide/Hydroxide and Hematite
• Habit: Coating
• Location = Intergranular Continuous Pore-Lining
Oil-Reservoir Description Ontology
0.1 mm

Figure 36: Association of low-level concepts to an image

3.4.2.1.2.2 *Soo et al's Method [SOO2003]*

In this work a method for making semantic annotation with the aid of an ontology is presented. The images are annotated with semantic tags that are derived from a domain ontology and thesaurus. This leads to an information retrieval approach richer than another one that only takes into account syntactic keyword matching.

Here a semi-automatic annotation technique is presented. This method translates queries formulated by users into a RDF query instance. Images need to be annotated in natural language by humans, and the algorithm translates such descriptions into RDF instances automatically.

According to this approach, semantic annotation assigns domain concepts in terms of semantic tags that are well defined in the ontology or thesaurus, but the process of annotating images with semantic tags is impractical with large datasets (is even difficult with a single image).

This method leaves the responsibility of annotating the images in a non-formal way (natural language) to humans, and makes the translation by itself.

3.4.2.1.2.3 Hu et al's Method [HU2003]

In this work a system to formally annotate medical images is presented. This approach is based on a well constructed image ontology specifying the domain knowledge, and introduces a Description Logic taxonomic inferential engine, responsible for semantics-based reasoning and image retrieval.

This approach is based on the assumptions that:

1. Expressing all the desired features using domain knowledge is feasible.
2. Manual annotation is practical.
3. Representing and reasoning about the textual description are performed with a reasonable complexity.

The ontology is used here to select the correct terms to describe features or objects present in the image (such as an abnormality in a radiography), and include new annotations based on these concept. An algorithm for controlling image annotation and retrieval can be seen in Figure 56.

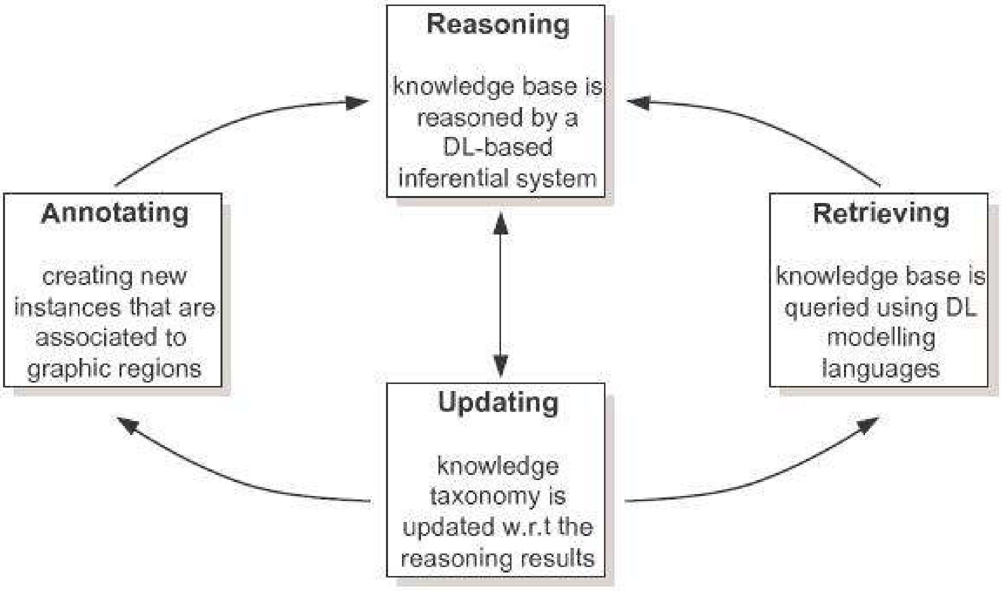

Reasoning
knowledge base is reasoned by a DL-based inferential system
Annotating
creating new instances that are associated to graphic regions
Retrieving
knowledge base is queried using DL modelling languages
Updating
knowledge taxonomy is updated w.r.t the reasoning results

Figure 37: DL-based inferential engine

It is shown that, by annotating an image using defined descriptors, a DL-based engine can answer queries properly. In this work an initial ontology for images is also presented. This ontology is considered to be an initial effort towards a uniform and standard reporting system.

3.4.2.1.2.4 Hollink et al's Method [HOL2003]

This work discusses a tool for semantic annotation and search of art images. Four different ontologies are used for annotation.

For annotating images, the user is provided a template derived from VRA (a specialization of the Dublin Core set of metadata elements for art images). This template includes:

Production-related descriptors: such as title, date or technique.

Physical descriptors: such as materials, measurements or type.

Administrative descriptors: location, collection ID, source or rights.

Two VRA data elements are not include in this template: description and subject. They are used to describe the content of the image. Subject of an image is described with a collection of statements, whose terms have being previously selected from the four ontologies provided. Each statement must have at least an agent (e.g. a portrait) on an object (e.g. still life).

The RDF Schema specifications of the ontologies, of the ontology links and of the annotation template are parsed into the tool. It generates an annotation and search interface for these specifications. The interface is used to annotate and query images.

It is shown that semantic annotation allows users to do concept search instead of a simple keyword search.

3.4.2.1.2.5 Gertz et al's Method [GER2003]

In this work a model and realization of an annotation framework for scientist to enrich different kind of documents is presented. The efforts are focused on annotation of scientific images. Concepts from given ontologies (on the domain) are used to construct metadata schemes for annotations.

The model presented in this work allows scientists to

1. Define semantic rich metadata for a particular domain.
2. Use such metadata to annotate images.
3. Use the metadata associated with images in data retrieval tasks on an image repository.

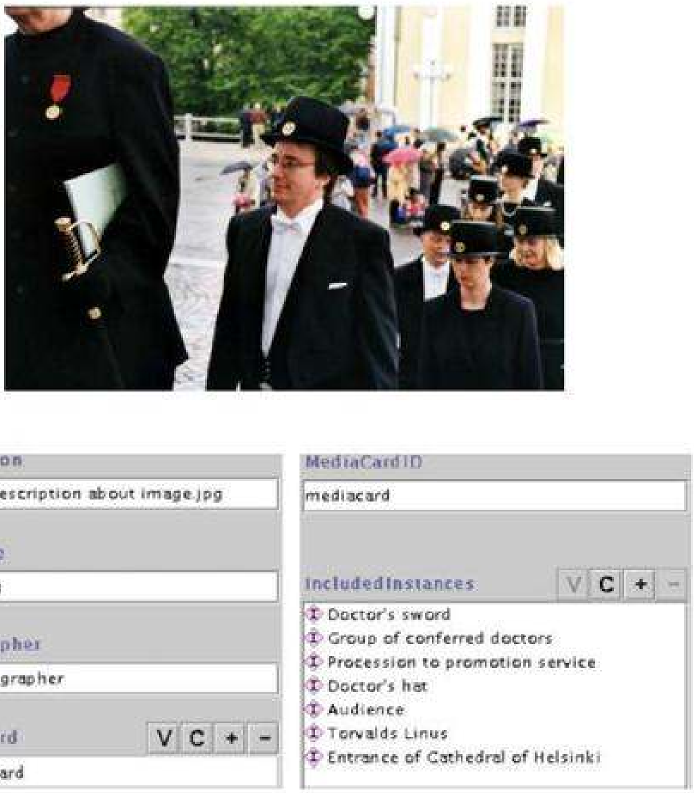
The annotation process is based on the discovering of “regions of interest” to be annotated with well known concepts in the domain. Since this is a model to serve in scientific tasks, is important to keep annotations separated from the actual images, in order to allow different people to annotate the same image, perhaps using different concepts. Regions of interest are specified spatial structures. This allows “fine grain” annotations, instead of just annotating the whole image. This model allows various text-based data retrieval scenarios.

3.4.2.1.2.6 Hyvönen et al's Method [HYV2002]

In the proposed system images are annotated according to ontologies, and the same information is offered to the user for him to build better queries in information retrieval.

Hyvönen follows this annotation scheme: every image is associated with a set of instances of the ontology. These instances occur in the image, and characterize its content. For including new metadata in an image, the annotator browses the ontology and, starting from the top level, he searches for the instances to be attached. If no previous instances are found, the annotator creates the new one and attaches it to the image. The new instance enriches the ontology, so the more images are annotated, the more rich the ontology is.

Figure 59 shows an example of annotation of a photograph:



Description: Textual description about image.jpg

ImageFile: image.jpg

Photographer: Mr. Photographer

Media Card: mediacard

MediaCardID: mediacard

IncludedInstances:
- Doctor's sword
- Group of conferred doctors
- Procession to promotion service
- Doctor's hat
- Audience
- Torvalds Linus
- Entrance of Cathedral of Helsinki

Figure 38: Example of annotation of an image

In image retrieval, the system provides the user a GUI containing the following semantic functionalities:

View-based filtering: the user can open ontologies to filter pictures of interest.

Image recommendations: When a query is performed, the image is semantically linked to other images, using its ontological definitions and annotations.

With this approach, more meaningful answers than just hit-lists are provided.

3.4.2.1.2.7 Schreiber et al's Method [SCH2001]

This work introduced the concept of *ontology-based image annotation*. In this article the use of background knowledge contained in ontologies to index and search collections of photographs is explored. An annotation strategy and tool to help users to annotate and search for specific images are developed.

Domain ontology and annotation ontology are separated concepts in this work. Annotation ontology is constructed to approach the annotation of a given image set. A mapping to link this one with the domain ontology needs to be developed.

Annotation process is simple: the user just selects a collection of concepts from the ontology to describe attributes in the image that he understands have to be annotated.

3.4.2.2 Ontologies in Content Based Image Retrieval

3.4.2.2.1 Introduction

Content based image retrieval (CBIR) aims to solve the problem of searching for digital images in large databases. Queries in CBIR make use of the features contained in the images themselves, rather than of annotations made by humans in every image (that is barely unfeasible with very large databases).

In the ideal CBIR, user formulates a semantic query, in natural language, and the system is able to retrieve "not annotated" images related to such query (e.g. the user gives a query such as "images of lung cancer", and the system retrieves the correct images contained in the database). This is an end goal that nowadays seems very far to achieve, since the information present in a single picture, with no added annotations, relates mainly to low level features (such as colours and primitive shapes). It is a very difficult task to translate a natural language query into this kind of low level features.

There are other approaches for CBIR that differs in the kind of query, such as:

Query by example. An example (a picture) is used as, or to complement, a query. The system should retrieve a set of pictures similar to the one used in the query.

Query by sketch. The user draws something that represents a feature related to the pictures he wants to be returned. This type of query does not need an example, but a drawing with some semantics in common with the pictures to be retrieved.

Low-level feature-based query. The user specifies a set of values (or ranges) for low level features of the picture (e.g. "60%-80% of colour black").

It is important to point out that the term CBIR refers only to image retrieval based on the contents of the pictures themselves, without taking into account textual (or any other kind of) information that can be artificially attached to them. Nevertheless, a CBIR method could include an image annotation step, and apply a textual information retrieval method later.

The latest approaches of CBIR use ontologies to save the gap between semantic queries and image low-level features, and also for assisting the users in query formulation. Some relevant examples of these approaches are described below.

The next section shows different methods and tools that have exploited ontologies to aid the process of content-based image retrieval. All these methods and approaches are ordered in an inverse chronological sorting, in order to keep the newest on the top. In contrast to this previous subsection, readers may notice the strong differences between newer systems and the older ones.

3.4.2.2.2 Methods, Approaches and Tools in Ontology-based Image Annotation

3.4.2.2.2.1 Vompras's Method [VOM2005]

In this work, CBIR is enhanced through integration of spatial context and semantic concepts into the feature extraction and retrieval process using the relevance feedback procedure [He, 2003] in a pre-processing step in the image mining process (feature selection and extraction).

In the semantic feature space, each image is represented by a set of characteristics and their weights. This semantic space is formed out of the mapping of low level feature space into high level semantic space. A visual representation of how a correspondence can be established between image primitives and concepts in an ontology can be seen in figure 61.

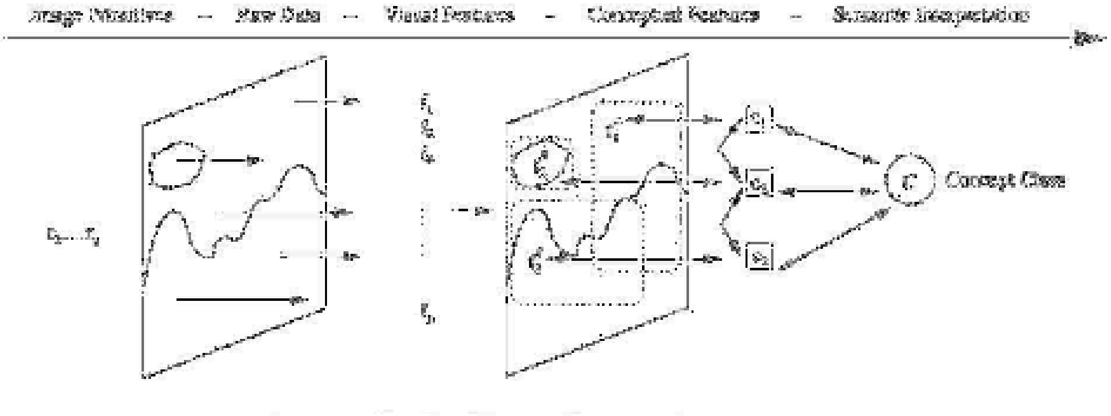


Figure 39: Levels of image content representation

There is a learning algorithm that is able to learn the classification of image objects into concepts (part of the given ontology). This algorithm requires human involvement, since a relevant feedback method is used to refine the mapping that needs to be constructed. The information retrieval process takes part in the algorithm, since users provide their relevant feedback after checking the results of a query. This information is used in the later process of information retrieval, and could be useful to for picture annotation tasks.

3.4.2.2.2.2 Mezaris et al. Method [MEZ2004]

This work attempts to address the problem of image retrieval in generic image collections, where no possibility of structuring a domain-specific knowledge base exists. To take further advantage of the human-friendly aspects of the region-based approach, low-level indexing features for the spatial regions are associated with high-level concepts humans are more familiar with.

No manual annotation of images is done in this approach. An ontology is employed to allow the user to define semantic (intermediate-level) objects that can be present in images, and the relationships among them. With this kind of information present in an image, retrieval becomes more effective and efficient.

A tool has been developed within this work to give support to all the process, from image-description to final queries.

The tool has a user-friendly interface for retrieval of colour images. The user does not have to be involved in technical matters (there is no need for doing manual tuning of weights, for example), and does not need image captions, either. Relevance feedback is used to refine image descriptions through user interaction as well.

3.4.2.2.2.3 Torres et al. Method [TOR2003]

This work studies the connection between a concept (human) level and a feature level in image retrieval. The method purposed uses a thesaurus to explore associations between text and image content.

The method embodies the assumption that the target images of the user are associated with concepts, instead of low-level features, such as colour or primitive shapes. Two levels of similarity are discussed in this work: *conceptual* and *visual*. These two levels are separated in two different layers of abstraction. One or more visual categories can be included in a conceptual one, and can be present in more than one either. In this approach, a one-to-one mapping is established between categories at the visual level and at the conceptual level. Structure of concepts can be learnt using relevance feedback.

In the adopted approach each concept is associated with a textual term in a one-to-one mapping. These textual labels are extracted from a formal thesaurus, so inconsistency is reduced. It is an assumption that the image has been divided in regions before beginning with the association of concepts with visual features.

Query formulation is based on the same thesaurus used for annotation of concepts. The user must select the concepts he needs to be present in the images to be retrieved from this formal vocabulary.

3.4.2.2.2.4 Town et al. Method [TOWI2004]

This work approaches the design of a specialized query language for content-based image retrieval. This language is based on an extensible ontology, for bridging the semantic gap between user language and image retrieval models.

OQUEL query language design and use is presented as an example of specific retrieval language for images. This language does not rely in any kind of annotation of images. Thus, format for the queries can be very flexible. A graphical representation of the process proposed can be seen in the figure below.

They use an ontological query language for providing an integrated query and retrieval framework. By basing query language on an ontology, one can capture both concrete and abstract relationships between images in a more powerful way. Because of the language is used to describe queries rather than describe image content, such relationships can be represented without prior constraints.

There is no need of examining exhaustively all the relations in a given image. One image can be considered irrelevant with a quick examination, avoiding incur in a combinatorial explosion.

3.4.2.2.2.5 Schober et al. Method [SCH2004]

They present a supervised Learning system, named OntoPic. This system is based on well known ontologies coded in DAML+OIL for providing domain knowledge. The OntoPic system provides an automated keyword annotation for images and content-based image retrieval on a semantic level. It uses a reasoner as a classifier, enabling a dual use of the ontology.

The OntoPic tool has three components:

1. Supervised training.
2. Analysis.
3. Retrieval.

The first task to use OntoPic is to design the ontology. After being constructed, the ontology can be enriched with knowledge for its usage in the domain, its concepts being mapped with image objects.

The training phase needs human interaction. The trainer has to assign some images to the training set and mark the images in order to obtain a semantic meaningful segmentation. After the training is complete, the ontology can be updated with training results.

The analysis phase deals with low-level feature extraction within a region, that have to be discretized, and the definition of spatial relations between different concepts. In this approach, the next spatial relations are taken into account: *isAvobe, isBelow and liesBeneath.*

The retrieval process is also supported by the ontology. The ontology itself provides a thesaurus to be used by the user to formulate queries. Terms in the query, then, match perfectly with those detected in the images.

3.4.2.2.2.6 Cha et al's method [CHA1999]

Cha et al. proposed a scheme based on a domain ontology to represent the situational meaning of an image. By situation, we mean something related to questions such as "What is the image about?", "Where is the location of an object in an image?", "What are people in the image doing?" or "Who is the one standing on the platform?", etc. The description scheme is able to answer this type of questions effectively. Typical examples include: "I want images of Bill Clinton at the White House" and "Give me images of Michael Jordan dunking in games". Cha gave a definition of ontology as the specification of relationships among notions or concepts of words through hierarchical classification.

The proposed solution represents features related to situational meaning of a still image, and consists of five descriptors which are geographical, component (different objects in an image), context, relational (spatial relationship) and temporal information.

3.4.3 TOOLS

3.4.3.1 Ontology-based Image Annotation tools

3.4.3.1.1 AKTive Media - Ontology based annotation system

This system contributes to the annotation process by suggesting the users in an interactive way.

Features supported by AKTive Media during image annotation are:

- Ability to import multiple ontologies represented in various ways (RDFS, OWL, DAML, etc)
- Support for all types of image formats (JPG, GIF, BMP, PNG, TIFF)
- Supports regional annotation, i.e. the ability to highlight and annotate specific regions of the image.
- Batch annotation, to annotate an entire collection of images at the same time.
- Integration with web services, to find relevant images.
- Knowledge suggestion using the smart SPARQL search facility. This is an experimental prototype for their Dissapearing Ontology technique, which minimizes the user effort in dealing with complex ontologies. It presents a rather generic and simple ontology to the user and populates the specialised underlying ontology while the user is annotating.
- Relational annotation of images.
- 2 Step persistance for storing annotation RDF graphs:The Local Store and the Central Triple Store.
- EXIF Metadata extraction.
- Easy to use interface.

Auto RDF import and export facility, export the annotated data to RDF for later access or to publish this information into the semantic web

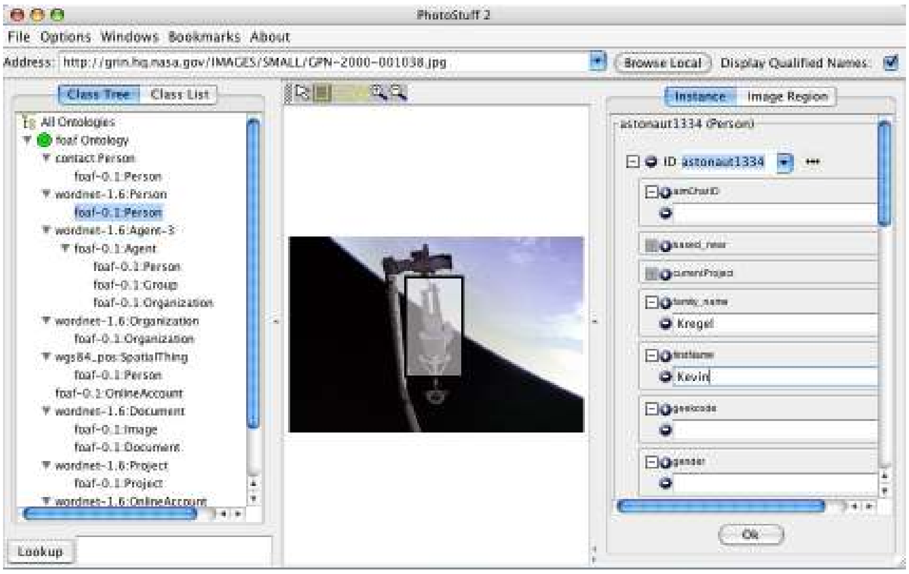
Link: http://www.dcs.shef.ac.uk/~ajay/html/cresearch.html

3.4.3.1.2 PhotoStuff.

This image annotation tool allows the user to annotate contents of specific regions in some images. It is based on several domain ontologies coded in OWL.

The Figure below shows the interface of PhotoStuff

Link: http://www.mindswap.org/2003/PhotoStuff/



Address: http://grin.hq.nasa.gov/IMAGES/SMALL/GPN-2000-001038.jpg

All Ontologies
- foaf Ontology
  - contact:Person
    - foaf-0.1:Person
  - wordnet-1.6:Person
    - foaf-0.1:Person
  - wordnet-1.6:Agent-3
    - foaf-0.1:Agent
      - foaf-0.1:Person
      - foaf-0.1:Group
      - foaf-0.1:Organization
  - wordnet-1.6:Organization
    - foaf-0.1:Organization
  - wgs84_pos:SpatialThing
    - foaf-0.1:Person
  - foaf-0.1:OnlineAccount
  - wordnet-1.6:Document
    - foaf-0.1:Image
    - foaf-0.1:Document
  - wordnet-1.6:Project
    - foaf-0.1:Project
  - wordnet-1.6:OnlineAccount

astonaut1334 (Person)

ID astonaut1334

aimChatID:

based_near

currentProject

family_name: Kregel

firstName: Kevin

geekcode:

gender:

Figure 40: PhotoStuff interface

3.4.3.1.3 M-OntoMat-Annotizer

This tool allows the users to annotate web sites and all their contents, images included, attaching OWL metadata to them.

Link: http://annotation.semanticweb.org/ontomat/index.html

3.4.4 REFERENCES

[CHA1999] Cha, K. H., Lee, H. A., Park, J. D., Ryu P.-M., Chae Y.-S., and Park S.-Y., “Representation of the situational meaning of an image based on domain ontology” ISO/IEC/JTC1/SC29/WG11 pp. 331, Lancaster, UK, Feb. (1999).

[CHA1999] Cha, K. H., Lee, H. A., Park, J. D., Ryu P.-M., Chae Y.-S., and Park S.-Y., “Representation of the situational meaning of an image based on domain ontology” ISO/IEC/JTC1/SC29/WG11 pp. 331, Lancaster, UK, Feb. (1999).

[GER2003] Gertz, M., Sattler, K., Gorin, F., Hogarth, M., and Stone, J. 2002. Annotating Scientific Images: A Concept-Based Approach. In Proceedings of the 14th international Conference on Scientific and Statistical Database Management (July 24 - 26, 2002). SSDBM. IEEE Computer Society, Washington, DC, 59-68.

[HE2003] Xiaofei He, Oliver King, Wei-Ying Ma, Mingjing Li, and Hong-Jiang Zhang. Learning a semantic space from user's relevance feedback for image retrieval. EEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 13, NO. 1, JANUARY 2003

[HOL2003] L. Hollink, A.Th. Schreiber, J. Wielemaker, and B. Wielinga. Semantic annotation of image collections. In Proc. of the Semannot Workshop, USA, 2003.

[HU2003] Hu, B., Dasmahapatra, S., Lewis, P., and Shadbolt, N. 2003. Ontology-Based Medical Image Annotation with Description Logics. In Proceedings of the 15th IEEE international Conference on Tools with Artificial intelligence (November 03 - 05, 2003). ICTAI. IEEE Computer Society, Washington, DC, 77.

[HYV2002] E. Hyv¨onen, A. Styrman, and S. Sarrela. Ontology- based image retrieval. In Towards the semantic web and web services, Proceedings of XML Finland 2002 Conference, pages 15–27, 2002.

[MEZ2004] V.Mezaris, I.Kompatsiaris, and M.G.Strintzis: "Region-based Image Retrieval using an Object Ontology and Relevance Feedback", EURASIP Journal on Applied Signal Processing, vol. 2004, no. 6, pp. 886-901, June 2004.

[PIE2004] Jean-Pierre Schober and Thorsten Hermes and Otthein Herzog (2004):Content-based Image Retrieval by Ontology-based Object Recognition.In V. Haarslev and C. Lutz and R. Möller (eds.), Proceedings of the KI-2004 Workshop on Applications of Description Logics (ADL-2004), Ulm, Germany, September 2004

[SCH2001] Schreiber, A. T., Dubbeldam, B., Wielemaker, J., and Wielinga, B. 2001. Ontology-Based Photo Annotation. IEEE Intelligent Systems 16, 3 (May. 2001), 66-74. DOI= http://dx.doi.org/10.1109/5254.940028

[SCH2004] Jean-Pierre Schober and Thorsten Hermes and Otthein Herzog (2004):Content-based Image Retrieval by Ontology-based Object Recognition.In V. Haarslev and C. Lutz and R. Möller (eds.), Proceedings of the KI-2004 Workshop on Applications of Description Logics (ADL-2004), Ulm, Germany, September 2004

[SIL2004] SILVA, Luis Alvaro Lima ; MASTELLA, Laura Silveira ; ABEL, Mara ; GALANTE, Renata de Matos ; ROS, Luiz F. de . An Ontology-Based Approach for Visual Knowledge: Image Annotation and Interpretation. In: Workshop on Ontologies and their Applications - SBIA2004, 2004, São Luis. Workshop on Ontologies and Their

Applications - WONTO'2004, 2004. p. 43-54.

[SOO2003] Von-Wun Soo , Chen-Yu Lee , Chung-Cheng Li , Shu Lei Chen , Ching-chih Chen, Automated semantic annotation and retrieval based on sharable ontology and case-based learning techniques, Proceedings of the thirdd ACM/IEEE-CS joint conference on Digital libraries, May 27-31, 2003, Houston, Texas

[TOR2003] Torres, José, Parkes, Alan, Corte-Real, Luis. Region-Based Relevance Feedback in Concept-Based Image Retrieval. Proc. of the 5th International Workshop on Image Analysis for Multimedia Interactive Services, Lisboa, Portugal, Abril (2004).

[TOW2004] Town, C., Sinclair, D. (2004), “Language-based querying of image collections on the basis of an extensible ontology”, Data Knowl. Eng, 48, nr. 2, 2004, pp. 251-267, Elsevier Science Publishers B.V.

[TOWI2004] Town, C., Sinclair, D. (2004), “Language-based querying of image collections on the basis of an extensible ontology”, Data Knowl. Eng, 48, nr. 2, 2004, pp. 251-267, Elsevier Science Publishers B.V.

[VOM2005] Johanna Vompras. Towards Adaptative Ontology-Based Image Retrieval. 17th GIWorkshop on the Foundations of Databases, Wörlitz, Germany. 2005. Pages 148-152

3.5 ONTOLOGIES AND GRID

3.5.1 INTRODUCTION

This subchapter presents a survey of the most relevant methods and models used for using ontologies and GRID together. This subchapter is divided in four sections: (1) Languages and Standards, (2) Platforms and Frameworks, (3) Projects and (4) Bibliography. All elements listed in the sections refer to elements (methods, tools, platforms…) that use GRID and ontologies in a combined way.

3.5.2 LANGUAGES & STANDARDS

3.5.2.1 WSDL (Web Service Description Language)

WSDL is an XML-based language to describe the public interface of the Web Services. A WSDL document contains all the information needed to invoke and consume automatically a service. WSDL documents are structured in seven major elements:

- **Types.** Definition of the new (complex) data types to be exchanged in the messages.
- **Messages**. Abstract and typed definition of the exchanged data.
- **Operations**. Abstract description of the actions supported by the service.
- **Port Types**. Set of abstract operations and the messages used.
- **Bindings**. Define message format and concrete protocols for a particular port type.
- **Ports**. Define individual endpoints by specifying a single address for a binding.
- **Services**. Set of related ports.

The latest version accepted is WSDL 1.1 but WSDL 2.0 has been proposed for recommendation by W3C on March 27th 2006. Some interesting links are:

- http://www.w3.org/TR/wsdl
- http://www.w3.org/TR/wsdl20/
- http://www.w3.org/2002/ws/desc/

3.5.2.2 GWSDL (Grid WSDL)

GWSDL is an extension of WSDL to support explicitly management of service lifecycles.

GWSDL provides the necessary mechanisms for architectural constructs of OGSI, namely definition of service data and inheritance of base interfaces.

3.5.2.3 GSFL (Grid Services Flow Services)

The Grid Services Flow Language (GSFL) is an XML based language that allows the specification of workflow descriptions for Grid services in the OGSA framework. Its architecture is composed by:

- **Service Providers**, the list of services taking part in the workflow.
- **Activity Model**, describes the list of important activities in the workflow.
- **Composition Model**, describes the interactions between the individual services.
- **Lifecycle Model**, describes the lifecycle for the various activities and the services which are part of the workflow.

3.5.2.4 SOAP (Simple Object Access Protocol)

SOAP is a lightweight protocol which defines how different processes can exchange structured data in XML format in a distributed environment. It has been designed following two principles: simplicity and extensibility. SOAP follows a one-way message exchange paradigm. The message framework developed is independent from the programming language used to implement the applications. But this framework does not directly provide security mechanisms, such as access control, integrity, confidentiality and non-repudiation. They should be provided using extensions of the SOAP model.

Some interesting links are:

- www.w3.org/TR/soap/
- http://ws.apache.org/soap/index.html

3.5.2.5 UDDI 3.0 (Universal Description Discovery & Integration)

UDDI is a XML-based and platform-independent registry for Web Services. This registry offers a standard mechanism to classify, catalog, discover, manage and consume Web services. UDDI defines a set of services to support the discovery, description and technical interfaces of Web Services (public or private) provided by organizations.

The UDDI Registry offers to the users the possibility to:

- **Find** Web Services providers and implementations.
- **Know** the transport protocols supported by the services and their security.
- **Search** services using keywords.

More information at http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm

3.5.2.6 WSML (Web Service Modeling Language)

The Web Service Modeling Language (WSML) provides a framework of different language variants to describe semantic Web services. WSML is a frame based language with an intuitive human readable syntax and XML and RDF exchange syntaxes, as well as a mapping to OWL.

WSML provides a formal syntax and semantics for the Web Service Modeling Ontology (WSMO). WSML is based on different logical formalisms, namely, Description Logics, First-Order Logic and Logic Programming, which are useful for modeling Semantic Web services.

More information about WSML in:

- www.wsmo.org/wsml/
- www.w3.org/Submission/WSML/
- www.w3.org/2004/12/rules-ws/paper/44/

3.5.2.7 WSMO (Web Service Modelling Ontology)

The Web Service Modelling Ontology (WSMO) provides a conceptual framework and a formal language for semantically describing all relevant aspects of Web services in order to facilitate the automation of discovering, combining and invoking electronic services over the Web.

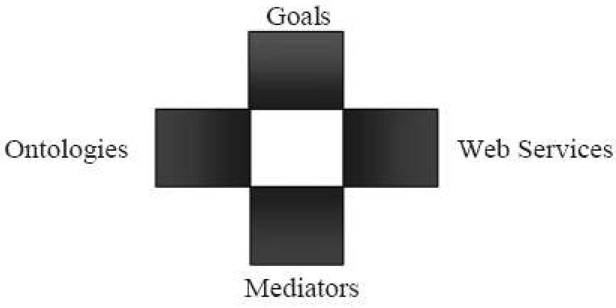


Figure 41: WSMO Core elements

WSMO is composed by four main elements:

- **Ontologies,** provide the terminology used by other WSMO elements
- **Web service descriptions**, describe the functional and behavioral aspects of a Web service
- **Goals** that represent user desires
- **Mediators**, aim at automatically handling interoperability problems between different WSMO elements.

One of the main characterizing features of WSMO is that goals, ontologies and Web Services are linked by mediators. Four kinds of mediators have been defined:

- **OO-mediators** resolve mismatches between ontologies and provide mediated domain knowledge specifications to the target component. They are used for ontology integration (aligning, merging and mapping ontology definitions).
- **WW-mediators** are used for establishing interoperability (choreography) among Web Services.
- **WG-mediators** link Web Services to Goals resolving terminological and functional differences.
- **GG-mediators** connect Goals and allow the creation of new goals from existing ones.

Some links to WSMO information:

- www.wsmo.org
- www.w3.org/Submission/WSMO/

3.5.2.8 XLANG (Web Services for Business Process Design)

XLANG was proposed by Microsoft as an extension of WSDL, used to model business processes as autonomous agents. It provides both the model of an orchestration of services as well as collaboration contracts between orchestrations. XLANG, like BPML, were designed with an explicit -calculus theory foundation.

XLANG defines the following set of operations as extensions to the standard WSDL operations:

- **delays** allow a thread to stall for a specified time period, or until another condition is met
- **raise** is a method to raise exceptions for certain actions
- **process control** combines actions together with conditional and iterative statements
- **correlation** provides a method for declaring longer running conversations
- **transaction support** allows definition of rollback procedures if one action in the execution fails
- **contracts** create agglomerate services by facilitating one way bindings between ports

More information about XLANG might be found at:
http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm

3.5.2.9 OWL-S (OWL for Services), also called DAML-S

OWL-S (previously DAML-S) is a OWL-based Web service ontology for describing the properties and capabilities of Web services in an unambiguous and computer-readable way, in order to facilitate automation of Web service tasks, including automated Web service discovery, execution, composition and interoperation.

A service description is composed by four ontologies:

- **Service** links to the other ontologies that make up the semantic description.
- **Service Profile** provides a description of what the service does (by exposing inputs, outputs, preconditions and effects), enabling advertising and discovery.
- **Service Model** provides a detailed description of a service's operation or how it works.
- **Service Grounding** provides details of how to interoperate with or access a service using messages. It defines the mapping of the atomic processes to the operations and message parts declared in the WSDL.

Some interesting links are:

- www.w3.org/Submission/OWL-S/
- www.daml.org/services/owl-s/

3.5.2.10 METEOR-S

The METEOR-S project at the LSDIS Lab, University of Georgia aims to extend these standards with Semantic Web technologies to achieve greater dynamism and scalability. In METEOR-S two approaches for annotating services are implemented and an algorithm is used for semantic publication of Web services in UDDI, and later during service discovery. METEOR-S defines its own infrastructure to publish services. The architecture of this publication service is divided into four layers, as shown in the following figure:

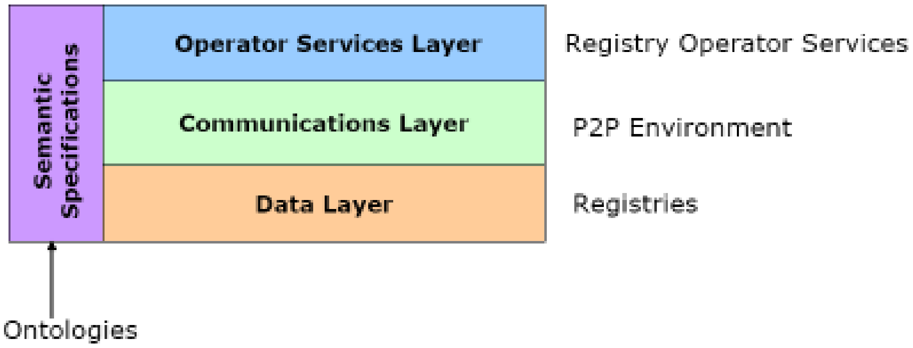


Figure 42: Architecture of METEOR-S Web Services Discovery Infrastructure

Web service registries are allocated in the *Data Layer*. All components communicate with each other using de *Communications Layer*. The *Operator Services Layer* enables registry operators to support various kinds of services that operate on their registries. The *Semantic Specifications Layer* provides with semantics, by using ontologies, at two levels: registries and individual Web services in the registries.

More information on http://lsdis.cs.uga.edu/projects/meteor-s/

3.5.3 PLATFORMS & FRAMEWORKS

3.5.3.1 WSRF (Web Service Resource Framework)

WSRF is an open framework proposed by OASIS (Organization for the Advancement of Structured Information Standards) for modelling and accessing resources using Web services. This includes mechanisms to describe views on the state, to support management of the state through properties associated with the Web service, and to describe how these mechanisms are extensible to groups of Web services.

Some interesting links are:

- www.oasis-open.org/committees/wsrf/
- www.globus.org/wsrf/
- www.cs.virginia.edu/~gsw2c/wsrf.net.html
- IBM WSRF
- http://ws.apache.org/wsrf/ (resources for implementation, APIs)

3.5.3.2 IRS III

The Internet Reasoning Service (IRS) project provides a framework and implemented infrastructure which supports the creation of semantic web services according to a WSMO ontology. It allows publication, composition, execution of different and heterogeneous web services which were built with the previous versions. Standalone software is automatically transformed into web services by creating wrappers which make them available on the Internet.

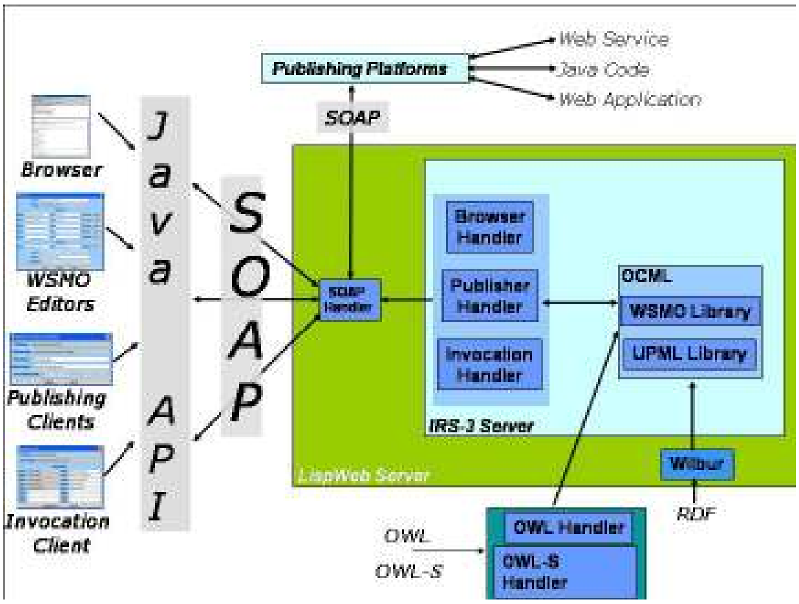


Figure 43: The IRS-III Architecture

The system architecture is composed by three main components: a HTTP Server implemented in LISP, a Publis her where services are registered, and finally, Clients who ask for a problem to be solved. When clients make requests, a broker decides what services are invoked. For IRS-III a WSMO specific Japa API and a browser have been developed. The IRS-III Ontology is implemented on WSMO where a class is used to represent each main concept.

3.5.3.3 SEWSIP (Semantic Web Services Integration Platform)

SEWSIP (SEmantic Web services Integration Platform) is a platform which uses a domain ontology to implement the service discovery, evaluation, selection and semi-automatic composition. It has been developed using Semantic Web technologies and Peer-to-peer technologies.

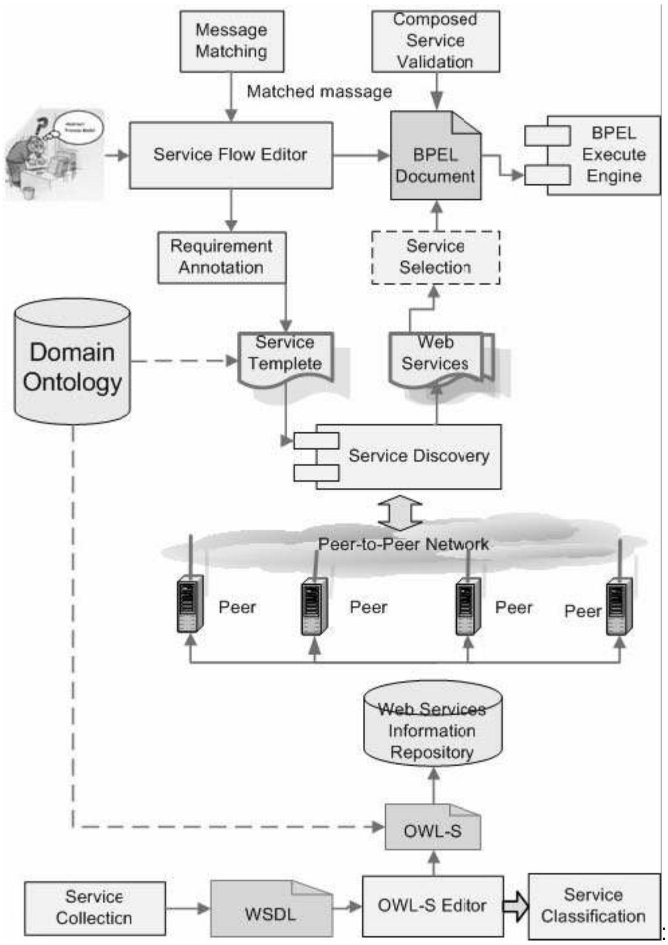


Figure 44: Diagram of SEWSIP Framework

Web services collected from the Internet or UDDI registries are annotated into OWL-S and then, the services are deployed in a P2P environment.

3.5.3.4 WSMX (Web Service Execution Environment)

WSMX is an open-source development of an execution environment for WSMO-based Semantic Web Services. It is an execution environment for business application where enhanced web services are integrated for various business applications. WSMX internal language is WSML (Web Service Modelling Language).

More information at www.wsmx.org

3.5.3.5 WSMT (Web Services Modelling Toolkit)

The Web Services Modeling Toolkit (WSMT) is a lightweight framework for the rapid creation and deployment of the tools for Semantic Web Services. This toolkit provides a collection of tools for describing Web services to use with the WSMO (Web Service Modeling Ontology), WSML (Web Service Modeling Language) and WSMX (Web Service Execution Environment).

3.5.3.6 SEAGRIN (SEmantic Adaptive Grid INfrastructure)

The infrastructure is based on wrappers and workflows.

The main objectives are:

- To integrate easily into existing infrastructure based on Web services, without imposing any additional implementation overhead on the existing services themselves.
- To provide robustness capabilities, such as fault tolerance, adaptation and dynamic reconfiguration.
- To incorporate dynamic changes to the system and allow integration of the SEAGRIN infrastructure with foreign systems based on Web Services technology.
- To define responsibilities of individual components comprised by the infrastructure, separate their concepts and thus aid the overall robustness of the system.

Four kinds of wrappers have been defined regarding their functionality:

- The Translator implements syntactic transformations of messages by applying XSLT transformation templates.
- The Converter implements semantic transformations (i.e. conversions) on messages, based on semantic annotations of primary services. These conversions do not alter the structure of data, but the data itself based on their interpretation by a particular service, for example converting units between various measurement systems.
- The Merger gathers several incoming messages from various sources into one input message to be passed to the encapsulated primary service.
- The Splitter duplicates an outgoing message, passing it to specified successors.

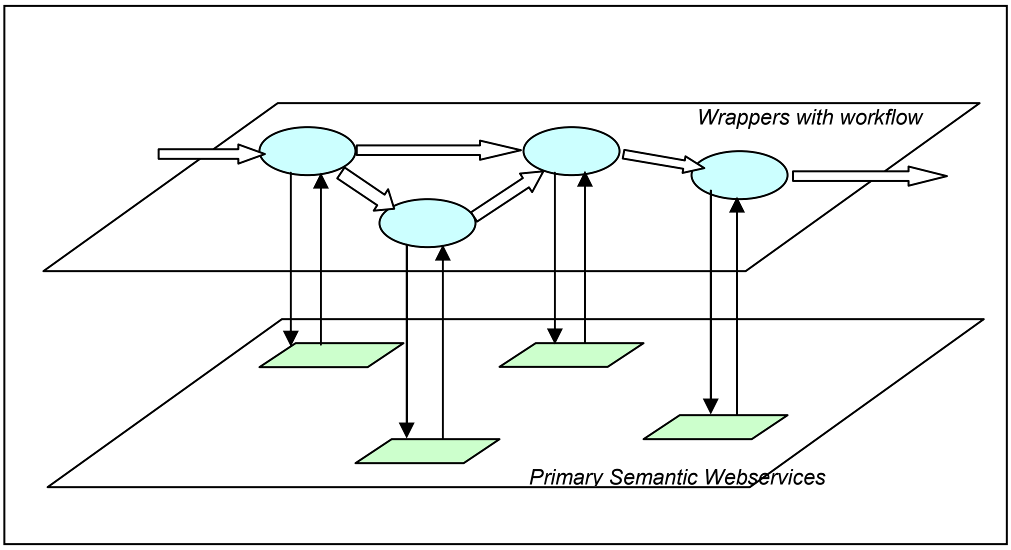


Figure 45: Overlay Grid

The design of the wrapper infrastructure is still work in progress and the concept is being refined further.

3.5.3.7 Jena

Jena is an open source Java framework for building Semantic Web applications, using RDF, RDFS or OWL, and includes a rule-based inference engine. Jena provides APIs to manage RDF and OWL.

Information about Jena at http://jena.sourceforge.net/

3.5.4 PROJECTS

3.5.4.1 TAVERNA

http://taverna.sourceforge.net/

Present version 1.3.1

The Taverna project aims to provide a language and software tools to facilitate easy use of workflow and distributed compute technology within the eScience community. As a component of the EPSRC funded myGrid project, Taverna is being developed by a consortium composed by the EBI (European Bioinformatics Institute), IT Innovation, the School of Computer Science, University of Newcastle, Newcastle Centre for Life, School of Computer Science at the University of Manchester and the Nottingham University Mixed Reality Lab.

Taverna is a multiplatform system, available freely under the terms of the GNU Lesser General Public License (LGPL), which provides support to the users to construct highly complex analyses over public and private data and computational resources. Only a computer with Java and network connection is needed to run the system. The most relevant components of Taverna are:

- SCUFL (Simple Conceptual Unified Flow Language). It is a language for workflow definition.
- Taverna Workbench. An environment for the creation, modification and execution of workflows.
- Freeflow. It is a workflow orchestration tool for web services.
- Taverna Data Model. It defines the data structure and types that are exchanges among services.
- MIR (myGrid Information Repository). A collection of experimental data and metadata for a community of users.
- Feta service discovery component. It allows users to find services over a registry of services.

3.5.4.2 PROTEUS

Proteus is a software environment for composing and running bioinformatics applications on the Grid. Workflow techniques are applied for designing and scheduling new applications. Metadata and ontologies are used to define bioinformatics processes. In order to combine different data sources and components, Proteus includes a Problem Solving environment.

One of the objectives is to build a reusable Knowledge Base of applications and results. The current implementation of PROTEUS is based on the KNOWLEDGE GRID but with addition of ontology modules.

PROTEUS is composed by a set of services organized in two layers:

- Core Services oriented to deal with Grid issues at low-level (to submit, execute and control of jobs over the Grid). These services also include discovery mechanisms. The information about services is coded in DAML+OIL.
- Ontology-based services that are the set of components ready to be composed into new applications. There exists a graphical browser to manage with the ontologies and different tools to define execution plans and visualize the final results.
-

3.5.4.3 Knowledge Web

The Knowledge Web is a 4 year Network of Excellence project funded by the European Commission 6th Framework Programme, started on January 1st, 2004. Its major goal is to support the transition process of Ontology technologies from Academia to Industry.

Official web site: http://knowledgeweb.semanticweb.org/

3.5.5 REFERENCES

[CHA1999] Dumitru Roman, Holger Lausen, and Uwe Keller. Web Service Modeling Ontology – Standard (WSMO-Standard). Working draft, Digital Enterprise Research Insitute (DERI), September 2004. Available from http://www.wsmo.org/2004/d2/v1.0.

[CHA1999] T. Tsai, H. Yu, H. Shih, P. Liao, R. Yan, S. Chou, “Ontology-Mediated Integration of Intranet Web Services”, Computer No 10, Volume 36, October 2003 http://computer.org

[GER2003] J. Cardoso, A. Sheth: “Introduction to Semantic Web Services and Web Process Composition”, Lecture Notes in Computer Science, Springer-Verlag, Volume 3387 / 2005, ISBN 3-540-24328-3 (2005)

[HE2003] Czajkowski, K., Ferguson, D. F., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S., Vambenepe, W.: The WS-Resource Framework, Version 1.0, available at www.globus.org/wsrf/specs/ws-wsrf.pdf (2004)

[HOL2003] Web Service Semantics – WSDL-S http://lsdis.cs.uga.edu/Projects/METEOR-S/WSDL-S/ http://lsdis.cs.uga.edu/proj/meteor/SWP.htm

[HU2003] OWL-based Web Service Ontology, http://www.daml.org/services/owl-s/

[HYV2002] I. Blanquer et al.: HealthGrid Whitepaper, http://whitepaper.healthgrid.org

[MEZ2004] Roman, D., Lausen, H., Keller, U., Oren, E., Bussler, C., Kifer, M., Fensel, D.: The Web Service Modelling Ontology. V1.0, 20 September 2004, WSMO Working Draft

[PIE2004] The OWL Services Coalition, “OWL-S: Semantic Markup for Web services” July

2004 http://www.daml.org/services/owl-s/

[SCH2001] Roman, D., Scicluna, J., Feier, C., Stollberg, M., Fensel, D.: Ontology-based Choreography and Orchestration of WSMO Services. V0.1, 1 March 2004, WSMO Working Draft

[SCH2004] Jos de Bruijn, Holger Lausen, and Dieter Fensel. The WSML Family of Representation Languages. Working draft, Digital Enterprise Research Insitute (DERI), November 2004.

[SIL2004] D. Wu, B. Parsia, E. Sirin, J. Hendler and D. Nau, “Automating DAML-S Web Services Composition Using SHOP2”, 2nd International Semantic Web Conference, ISWC 2003, Sanibel Island, Florida, USA, October 2003 http://www.mindswap.org/papers/ISWC03-SHOP2.pdf

[SOO2003] L. Childers et al.: “AccessGrid: Immersive Group-to-Group Collaborative Visualisation”, In: Proceedings of Immersive Projection Technology (2000)

[TOR2003] Domingue, J., Cabral, L., Hakimpour, F., Sell, D., Motta, D.: IRS III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services. Proceedings of the Workshop on WSMO Implementations (WIW 2004). Frankfurt, Germany. CEUR Workshop Proceedings (online), Vol. 113, 2004

[TOW2004] Massimo Paolucci, Katia Sycara, Takuya Nishimura, Naveen Srinivasan, “Using DAML-S for P2P Discovery”, International Conference on Web Services, ISWS 2003, Las Vegas, Nevada, USA, June 2003 http://www-2.cs.cmu.edu/~softagents/papers/p2p_icws.pdf

[TOWI2004] Cimpian, E., Moran, M., Oren, E., Vitvar, T., Zaremba, M.: Overview and Scope of WSMX. WSMO Working Draft. V0.2, 08 February 2005

[VOM2005] de Bruijn, J: The WSML Specification, WSML Working Draft, 3 February 2005

4 Technical Annex: Use Cases

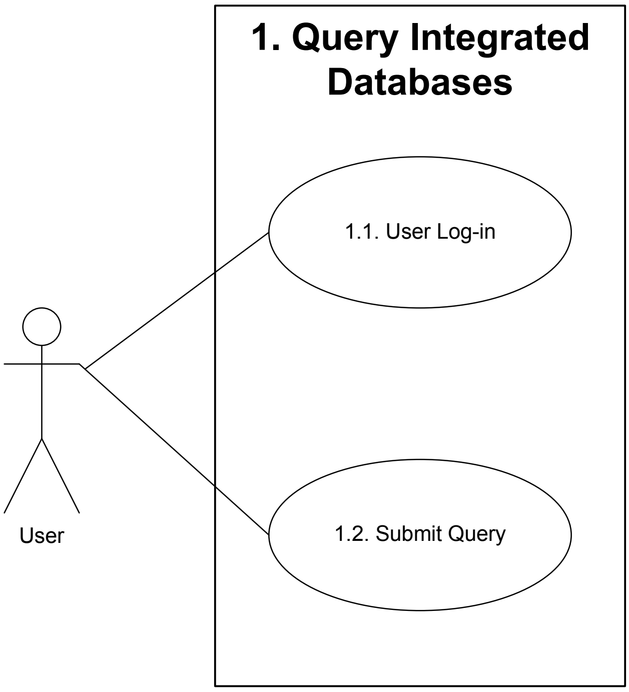


Figure 46: Feature 1- Query Integrated Databases

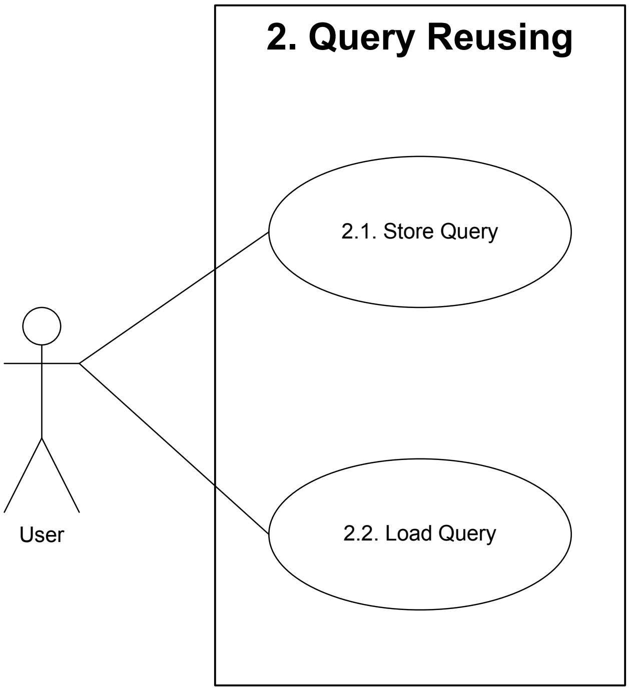


Figure 47: Feature 2 - Query Reusing

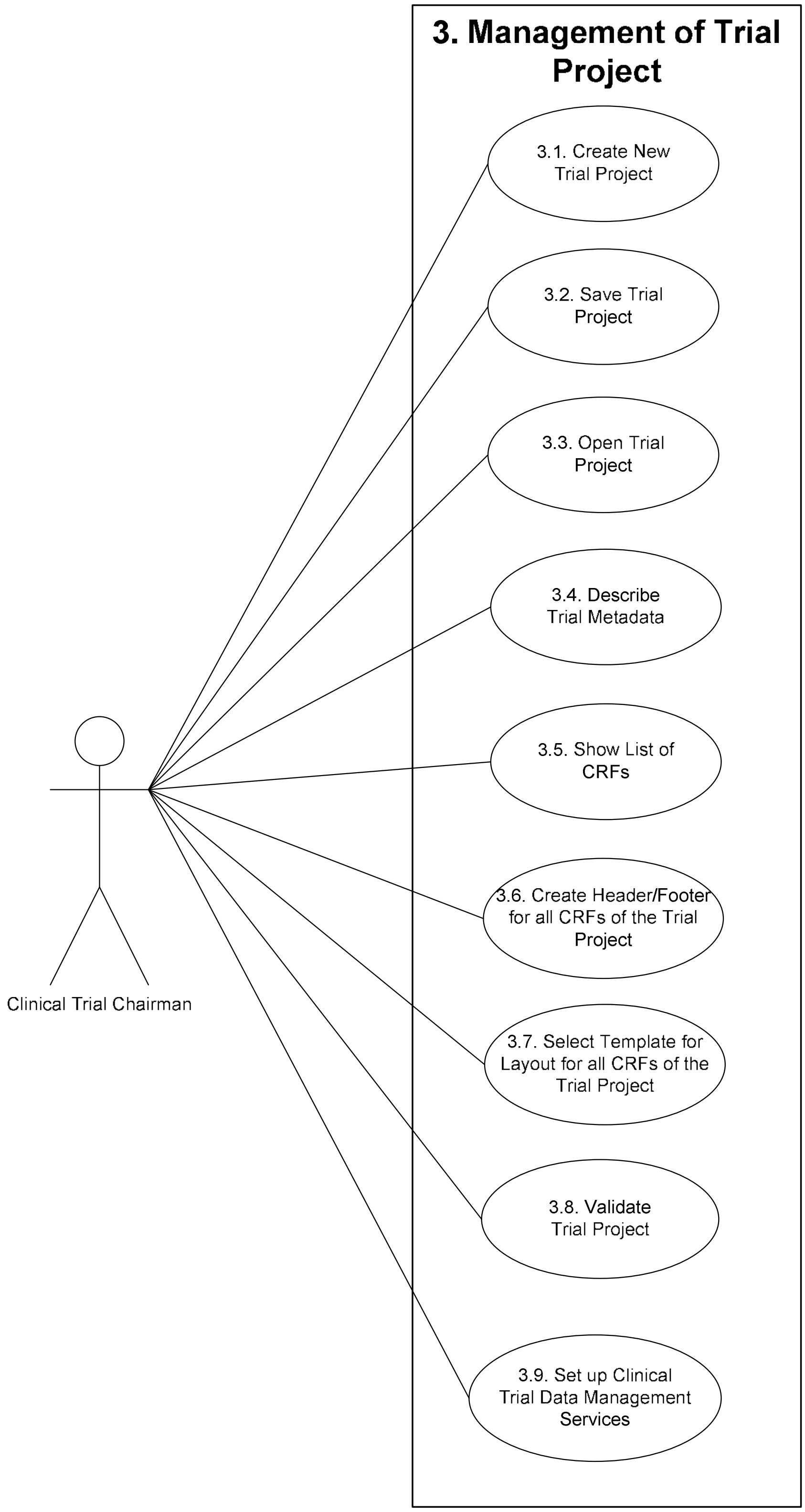

3. Management of Trial Project
3.1. Create New Trial Project
3.2. Save Trial Project
3.3. Open Trial Project
3.4. Describe Trial Metadata
3.5. Show List of CRFs
3.6. Create Header/Footer for all CRFs of the Trial Project
3.7. Select Template for Layout for all CRFs of the Trial Project
3.8. Validate Trial Project
3.9. Set up Clinical Trial Data Management Services
Clinical Trial Chairman

Figure 48: Feature 3 - Manage Trial Project

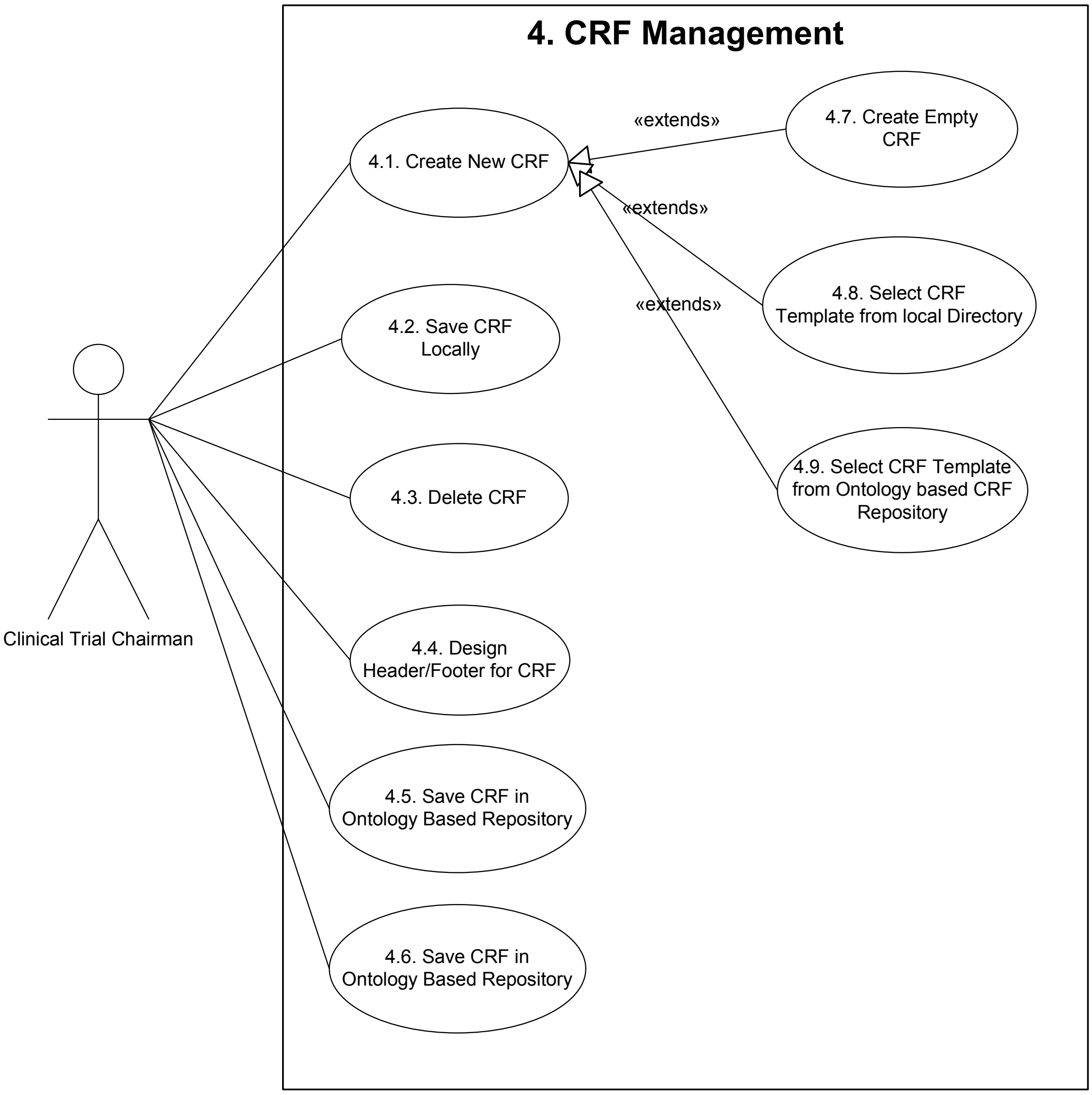

4. CRF Management
4.1. Create New CRF
4.2. Save CRF Locally
4.3. Delete CRF
4.4. Design Header/Footer for CRF
4.5. Save CRF in Ontology Based Repository
4.6. Save CRF in Ontology Based Repository
4.7. Create Empty CRF
4.8. Select CRF Template from local Directory
4.9. Select CRF Template from Ontology based CRF Repository
«extends»
«extends»
«extends»
Clinical Trial Chairman

Figure 49: Feature 4 - CRF Management

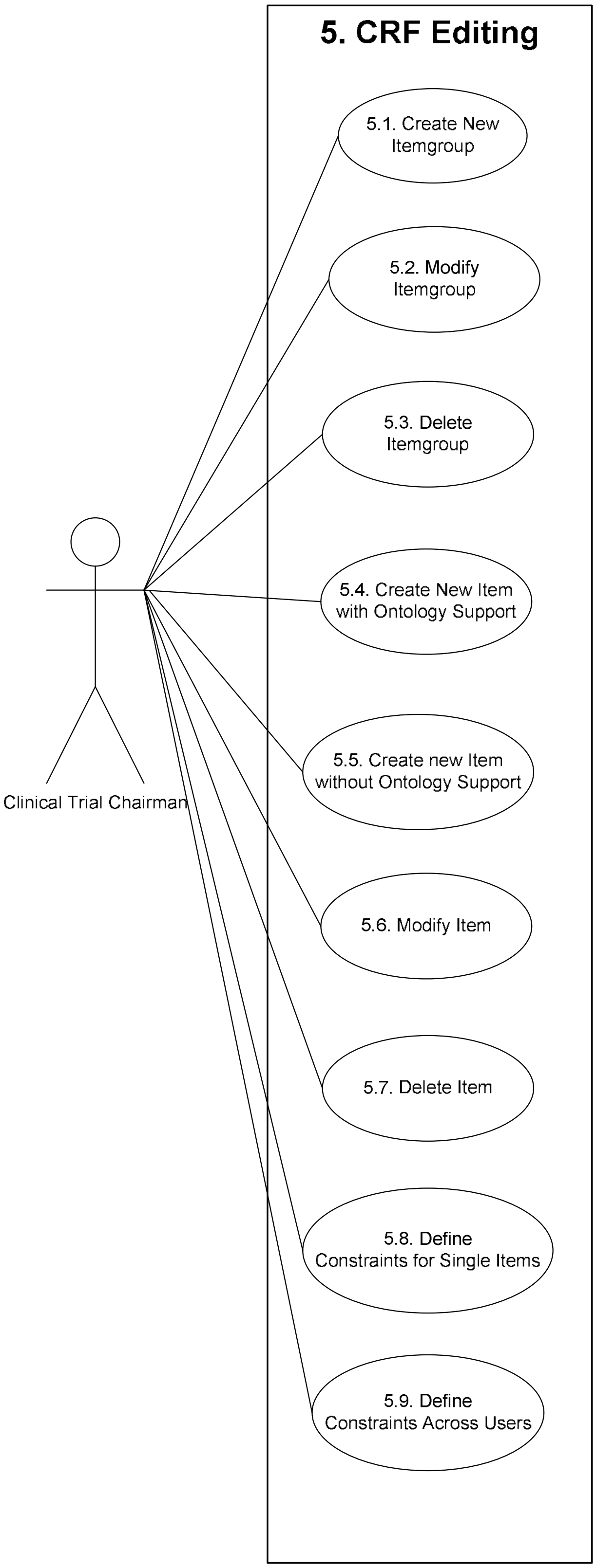

5. CRF Editing
5.1. Create New Itemgroup
5.2. Modify Itemgroup
5.3. Delete Itemgroup
5.4. Create New Item with Ontology Support
5.5. Create new Item without Ontology Support
5.6. Modify Item
5.7. Delete Item
5.8. Define Constraints for Single Items
5.9. Define Constraints Across Users
Clinical Trial Chairman

Figure 50: Feature 5 - CRF Editing

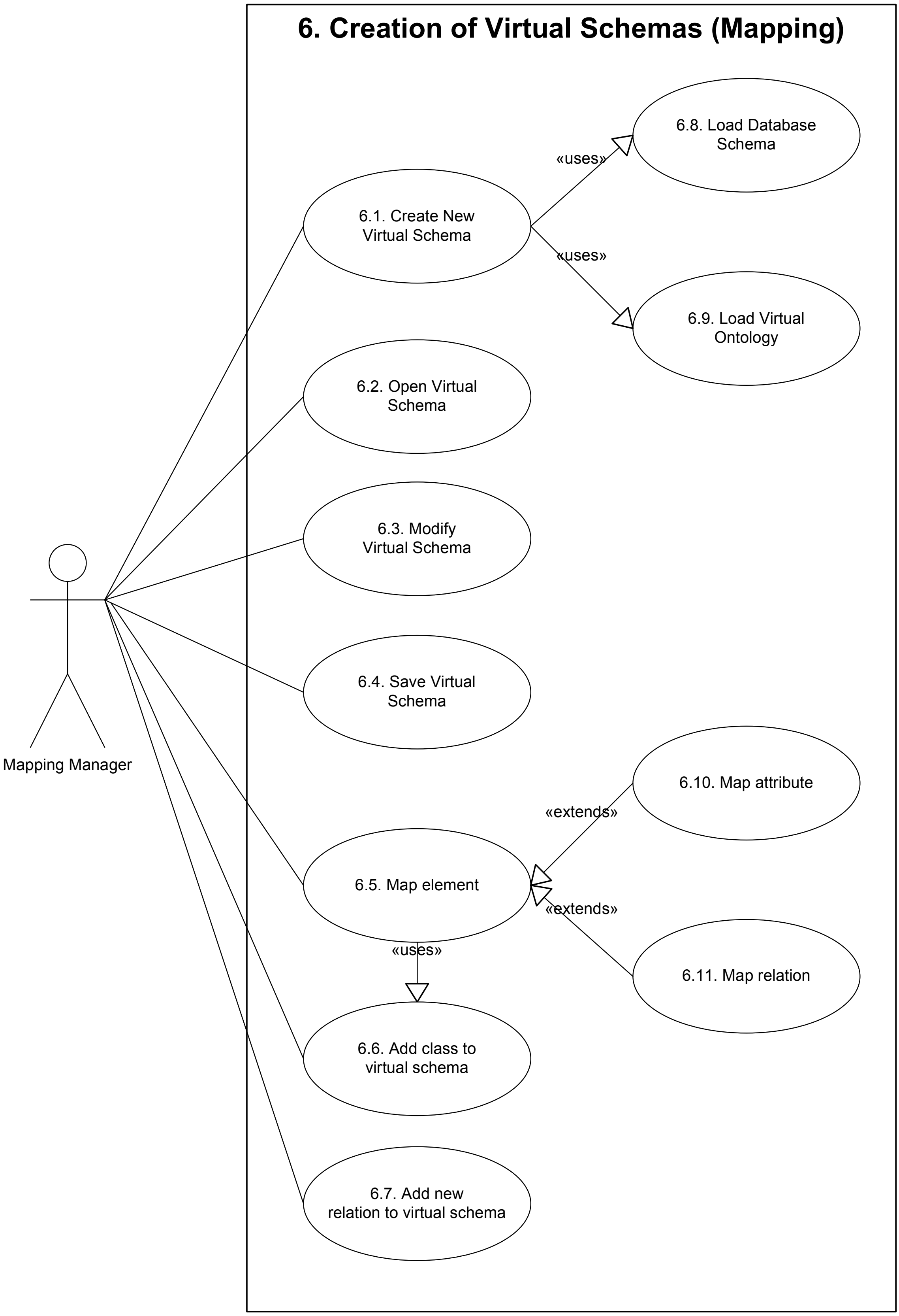

6. Creation of Virtual Schemas (Mapping)
6.1. Create New Virtual Schema
6.2. Open Virtual Schema
6.3. Modify Virtual Schema
6.4. Save Virtual Schema
6.5. Map element
6.6. Add class to virtual schema
6.7. Add new relation to virtual schema
6.8. Load Database Schema
6.9. Load Virtual Ontology
6.10. Map attribute
6.11. Map relation
«uses»
«uses»
«extends»
«extends»
«uses»
Mapping Manager

Figure 51: Feature 6 - Creation of Virtual Schemas (Mapping)

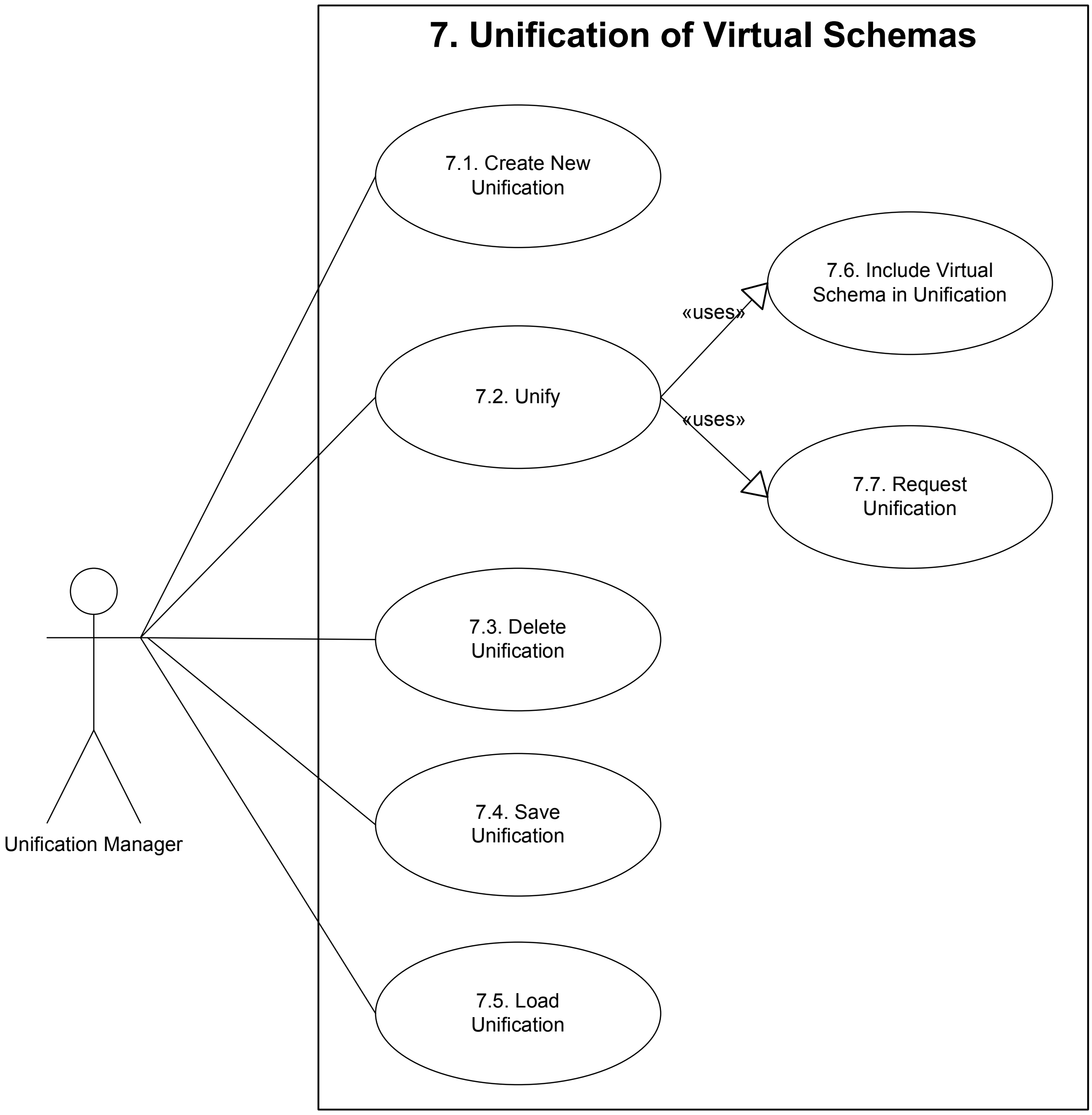


Figure 52: Feature 7 - Unification of Virtual Schemas

Feature 1: Query Integrated Databases

Use Case:	**1.1. USER LOG-IN**	
Actors:	User	
Purpose:	To get the user logged into the system	
Summary:	The user gives user name and password and gets access to the repositories associated with his account.	
Preconditions:		
Main Flow:	**User**	**System**
	1.- The user submits the user name and the password	
		2.- The system gives access to the user
Exceptions:	If user name and password are invalid, the system request the user to re-input them.	

Use Case:	**1.2. SUBMIT QUERY**	
Actors:	User	
Purpose:	Retrieve data and metadata in response to a query	
Summary:	A user performs a query against a unified virtual schema representing an integration of databases	
Preconditions:	The user must be logged into the system.	
Main Flow:	**User**	**System**
	1.- The user submits a string representing the query to the system	
		2.- The system retrieves the data and metadata associated
Exceptions:	None	

Feature 2: Query Reusing

Use Case:	**2.1. STORE QUERY**	
Actors:	User	
Purpose:	To store the query in a repository for future reuse	
Summary:	A query is stored in a repository for possible future reuse.	
Preconditions:	The query must be well formatted	
Main Flow:	**User**	**System**
	1.- The user provides the query as a string, with an optional description of its purpose in natural language	
		2.- The system returns a report message, confirming that the query has been stored in the repository
Exceptions:		

Use Case:	**2.2. LOAD QUERY**	
Actors:	User	
Purpose:	To retrieve a query stored in the repository	
Summary:	The user selects a query stored in the repository	
Preconditions:		
Main Flow:	**User**	**System**
	1.- The user selects the query to be retrieved from the repository	
		2.- The system provides the user with the string representing the query
Exceptions:		

Feature 3: Management of Trial Projects

Use Case:	**3.1. CREATE NEW TRIAL PROJECT**	
Actors:	Clinical trial chairman	
Purpose:	The creation of a new clinical trial project	
Summary:	The user requests the system to create a new trial project. The trial project is created	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The Clinical Trial Chairman request the system to create a new Trial Project	
		2.- The system requires the user to give a name to the clinical trial project
	3.- The Clinical Trial Chairman gives a name to the clinical trial project	
		4.- The system reports that the clinical trial project has been created, and shows the interface for describing metadata
Exceptions:		

Use Case:	**3.2. SAVE TRIAL PROJECT**	
Actors:	Clinical trial chairman	
Purpose:	Store the trial project	
Summary:	The user saves the trial project for future management and reusing	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The Clinical Trial Chairman request the system to save the clinical trial project	
		2.- The system reports that the clinical trial project has been saved
Exceptions:		

Use Case:	**3.3. OPEN TRIAL PROJECT**	
Actors:	Clinical trial chairman	
Purpose:	Load a trial project into the system	
Summary:	A trial project from a local repository can be selected and will be opened for further editing.	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The Clinical Trial Chairman request the system to open a stored trial project	
		2.- The system shows the list of stored projects
	3.- The user selects the project to be opened	
		4.- The system reports that the project has been opened, and gives access to it to the user
Exceptions:		

Use Case:	**3.4. DESCRIBE TRIAL PROJECT WITH METADATA**	
Actors:	Clinical trial chairman	
Purpose:	Describe the trial project with proper metadata	
Summary:	The user describes the trial project with metadata using an interfaced intended for it	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The Clinical Trial Chairman provides the metadata description	
		2.- The system reports that the clinical trial project has been annotated
Exceptions:		

Use Case:	**3.5. SHOW LIST OF CRFs**	
Actors:	Clinical trial chairman	
Purpose:	To show the list of available CRFs	
Summary:	List of all CRFs with their name and description which are part of the current trial project is shown	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user request the system to list the CRFs	
		2.- The system shows the list of CRFs
Exceptions:		

Use Case:	**3.6. CREATE HEADER/FOOTER FOR ALL CRFs OF THE TRIAL PROJECT**	
Actors:	Clinical trial chairman	
Purpose:	To design the header and footer for all CRFs within this project	
Summary:	The user creates a template for all CRF's within the current trial project.	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The clinical trial chairman request the system to design a common header/footer	
		2.- The system gives access to the header/footer template
	3.- The user modify the header/footer	
		4.- The system reports that the header/footer has been set
Exceptions:		

Use Case:	**3.7. SELECT TEMPLATE FOR LAYOUT FOR ALL CRFs OF THE TRIAL PROJECT**	
Actors:	Clinical trial chairman	
Purpose:	Use a template for designing CRFs within the trial project	
Summary:	A template for the layout (specifying colour, font size…) of all CRFs for the clinical trial project can be chosen	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user request the system to select a template	
		2.- The system shows the list of available templates
	3.- The system selects the template to be used	
		4.- The system loads the template into the project
Exceptions:		

Use Case:	**3.8. VALIDATE TRIAL PROJECT**	
Actors:	Clinical trial chairman	
Purpose:	Use a template for designing CRFs within the trial project	
Summary:	It will be checked if the current trial project is valid and complete. Only when a trial project is valid and complete Data Management Services can be created to conduct the clinical trial	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user request the system to validate the project	
		2.- The system reports the state of the trial project

Use Case:	**3.9. SET UP CLINICAL DATA MANAGEMENT SERVICES**	
Actors:	Clinical trial chairman	
Purpose:	Setting up of clinical data management services.	
Summary:	The user sets up the clinical data management services for this trial project	
Preconditions:	The clinical trial project must be valid and complete	
Main Flow:	**Clinical Trial Chairman**	**System**
Exceptions:		

Feature 4: CRF Management

Use Case:	**4.1. CREATE EMPTY CRF**	
Actors:	Clinical trial chairman	
Purpose:	Create a new empty CRF inside the project	
Summary:	An empty CRF template will be added to the current trial project	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The clinical trial chairman decides to add an empty CRF to the project	
		2.- The system request the user to give a name to the new CRF
	3.- The clinical trial chairman gives a name to the CRF	
		4.- The system reports that the new CRF has been added to the project
Exceptions:		

Use Case:	**4.2. SAVE CRF LOCALLY**	
Actors:	Clinical trial chairman	
Purpose:	To store the CRF locally	
Summary:	Current CRF can be saved in a local directory to use it as a template in other trial projects	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user requests the system to save the CRF locally	
		2.- The system reports that the CRF has been saved
Exceptions:		

Use Case:	**4.3. DELETE CRF**	
Actors:	Clinical trial chairman	
Purpose:	To delete the CRF	
Summary:	Current CRF is deleted from the repository.	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user requests the system to delete current CRF	
		2.- The system reports that the CRF has been deleted
Exceptions:		

Use Case:	**4.4. DESIGN HEADER/FOOTER FOR CRF**	
Actors:	Clinical trial chairman	
Purpose:	Design header and footer for all CRFs	
Summary:	Design header and footer for all CRFs in the trial project	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user designs header and footer using an interface	
		2.- The system reports that the template has been applied
Exceptions:		

Use Case:	**4.5. SAVE CRF IN LOCAL REPOSITORY**	
Actors:	Clinical trial chairman	
Purpose:	To store the CRF in the local repository	
Summary:	Save current CRF in the repository for future reusing	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user requests to store the CRF in the repository	
		2.- The system reports that the CRF has been stored
Exceptions:		

Use Case:	**4.6. SAVE CRF IN ONTOLOGY-BASED REPOSITORY**	
Actors:	Clinical trial chairman	
Purpose:	To store the CRF in the ontology driven repository	
Summary:	CRFs or parts of CRFs (Items or Item groups) can be saved in an ontology driven CRF repository. Only valid CRFs or CRF parts can be saved	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user requests to store the CRF in the repository	
		2.- The system reports that the CRF has been stored
Exceptions:		

Use Case:	**4.7. CREATE EMPTY CRF**	
Actors:	Clinical trial chairman	
Purpose:	Create and empty CRF	
Summary:	The user creates an empty CRF	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user requests to create and empty CRF	
		2.- The system reports that the CRF has been created
Exceptions:		

Use Case:	**4.8. SELECT CRF TEMPLATE FROM LOCAL DIRECTORY**	
Actors:	Clinical trial chairman	
Purpose:	Use a template for designing new CRFs	
Summary:	User can select a CRF template from a local directory. The selected template will be added to the current trial project.	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user requests the system to show the template list	
		2.- The system shows the list of available templates
	3.- The user selects the template to be included	
		4.- The system reports that the template has been included in the project
Exceptions:		

Use Case:	**4.9. SELECT CRF TEMPLATE FROM ONTOLOGY BASED CRF REPOSITORY**	
Actors:	Clinical trial chairman	
Purpose:	Use a template for designing new CRFs	
Summary:	The user looks for a stored template using a ontology-based search tool. When retrieved, the system request the user to configure it. The template is applied to CRFs in the clinical trial.	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user requests the system to show the query system to search a template	
		2.- The system gives access to the user to the query tool
	3.- The user configures the query and submits it	
		4.- The user retrieves the templates
	5.- The user selects the template to be included in the project	
		6.- The system reports that the template has been included in the project
Exceptions:		

Feature 5: CRF Editing

Use Case:	**5.1. CREATE NEW ITEMGROUP**	
Actors:	Clinical trial chairman	
Purpose:	Create a new itemgroup for the CRF	
Summary:	A new Itemgroup for a CRF is created by the user.	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user requests the system the creation of a new itemgroup	
		2.- The system requests to user to give a name to the itemgroup
	3.- The user gives a name to the new itemgroup, as well as the properties of it	
		4.- The system reports that the itemgroup has been created
Exceptions:		

Use Case:	**5.2. MODIFY ITEMGROUP**	
Actors:	Clinical trial chairman	
Purpose:	To modify the properties of the itemgroup	
Summary:	The user modifies the properties of the itemgroup	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user requests the system to modify the properties of the itemgroup	
		2.- The system gives access to the itemgroup properties to the user
	3.- The user modifies the itemgroup properties	
		4.- The system reports that the itemgroup has been modified
Exceptions:		

Use Case:	**5.3. DELETE ITEMGROUP**	
Actors:	Clinical trial chairman	
Purpose:	To delete this itemgroup	
Summary:	To delete this itemgroup	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user selects the itemgroup to be deleted	
		2.- The system reports that the itemgroup has been deleted
Exceptions:		

Use Case:	**5.4. CREATE NEW ITEM WITHOUT ONTOLOGY SUPPORT**	
Actors:	Clinical trial chairman	
Purpose:	To create new item, without ontological basis	
Summary:	A new item is created by the user without ontology support, setting up the properties.	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user selects an itemgroup, and request the system to create a new item	
		2.- The system requests the user to set up the properties of the item
	3.- The user sets the properties of the item	
		4.- The system reports that the item has been created
Exceptions:		

Use Case:	**5.5. CREATE NEW ITEM WITH ONTOLOGY SUPPORT**	
Actors:	Clinical trial chairman	
Purpose:	To create new item, based on an ontology	
Summary:	The user creates a new Item with ontology support, setting up the properties using the ontology.	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user selects an itemgroup, and request the system to create a new item	
		2.- The system shows the ontology visualizer
	3.- The user selects the terms to be included in the item description	
		4.- The system reports that the item has been created
Exceptions:		

Use Case:	**5.6. MODIFY ITEM**	
Actors:	Clinical trial chairman	
Purpose:	To modify item's properties	
Summary:	User modifies item's properties	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user selects an item and request the system to modify its properties	
		2.- The system asks the user what are the properties to be modified
	3.- The user modifies the properties of the item	
		4.- The system reports that the item has been modified
Exceptions:		

Use Case:	**5.7. DELETE ITEM**	
Actors:	Clinical trial chairman	
Purpose:	To delete an item within an itemgroup	
Summary:	The user deletes an item from an itemgroup	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user selects an item to be deleted	
		2- The system reports that the item has been deleted
Exceptions:		

Use Case:	**5.8. DEFINE CONSTRAINTS FOR SINGLE ITEM**	
Actors:	Clinical trial chairman	
Purpose:	To define constraints	
Summary:	The user defines the constraints for a single item.	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user selects the item and submits the constraint	
		2.- The system reports that the constraint has been set
Exceptions:		

Use Case:	**5.9. DEFINE CONSTRAINTS ACROSS ITEMS**	
Actors:	Clinical trial chairman	
Purpose:	To define constraints	
Summary:	It can be specified that when an item/some items has/have a particular value/particular combination of values, another item/itemgroup/error message will be shown	
Preconditions:		
Main Flow:	**Clinical Trial Chairman**	**System**
	1.- The user submits the constraint	
		2.- The system reports that the constraint has been set
Exceptions:		

Feature 6: Creation of Virtual Schemas (Mapping)

Use Case:	**6.1. CREATE NEW VIRTUAL SCHEMA**	
Actors:	Mapping Manager	
Purpose:	To define a virtual schema that guarantee semantic integration	
Summary:	The user builds a virtual schema for one database, based on the mapping of elements of it into elements from the domain ontology	
Preconditions:		
Main Flow:	**Mapping Manager**	**System**
	1.- The user requests the system the creation of a new virtual schema	
		2.- The system asks the user to give a name to the new virtual schema, and load the database schema and the domain ontology
	3.- The user gives a name to the new virtual schema. <uses load database schema> <uses load domain ontology>	
		4.- The user reports that the new virtual schema has been created, put the mapping options available
Exceptions:		

Use Case:	**6.2. OPEN VIRTUAL SCHEMA**	
Actors:	Mapping Manager	
Purpose:	To open an existing virtual schema for its edition	
Summary:	The user opens an existing virtual schema, that is maybe not complete, to modify it.	
Preconditions:		
Main Flow:	**Mapping Manager**	**System**
	1.- The user requests the system to open a virtual schema	
		2.- The system shows the list of available virtual schemas
	3.- The user selects the desired virtual schema	
		3.- The system loads the selected virtual schema and shows its current state to the user
Exceptions:		

Use Case:	**6.3. MODIFY VIRTUAL SCHEMA**	
Actors:	Mapping Manager	
Purpose:	To modify the current virtual schema	
Summary:	The user modifies an existing virtual schema, editing or deleting one or more mapping relations, or creating new ones	
Preconditions:		
Main Flow:	**Mapping Manager**	**System**
	1.- The user modifies or deletes existing mapping relations, or creates new ones	
		2.- The system reflects the changes performed by the user, incorporating them to the mapping model
Exceptions:		

Use Case:	**6.4. SAVE VIRTUAL SCHEMA**	
Actors:	Mapping Manager	
Purpose:	To save all changes in the working virtual schema	
Summary:	The user saves the virtual schema he is currently working with. All unsaved changes are stored in the disk	
Preconditions:		
Main Flow:	**Mapping Manager**	**System**
	1.- The user selects the save option	
		2.- The system saves all last modifications and stores them into de disk
Exceptions:		

Use Case:	**6.5. MAP ELEMENT**	
Actors:	Mapping Manager	
Purpose:	To establish a new mapping relation between elements in the database schema and the virtual schema	
Summary:	The user creates a new mapping relation, which relates one element of the current database schema with one element of the current virtual schema	
Preconditions:		
Main Flow:	**Mapping Manager**	**System**
	1.- The user selects an element of the virtual schema, and an element of the database schema, and indicates the system to establish a relation between them	
		2.- The system creates a new mapping relation between the selected elements
Exceptions:		

Use Case:	**6.6. ADD CLASS TO VIRTUAL SCHEMA**	
Actors:	Mapping Manager	
Purpose:	To include a new class into an existing virtual schema	
Summary:	The user asks the system to add a new class into the current virtual schema	
Preconditions:		
Main Flow:	**Mapping Manager**	**System**
	1.- The user selects a class from the domain ontology, and requests the system to add it to the virtual schema	
		2.- The system reports that the class has been added
Exceptions:		

Use Case:	**6.7. ADD NEW RELATION TO VIRTUAL SCHEMA**	
Actors:	Mapping Manager	
Purpose:	To include a new class into an existing virtual schema	
Summary:	The user asks the system to add a new class into the current virtual schema	
Preconditions:		
	1.- The user selects two elements from the virtual schema and request the system to create a relation between them	
		2.- The system asks the user for the name of the new relation
	3.- The user inputs the name for the new relation	
		4.- The system reports that the new relation has been created
Exceptions:		

Use Case:	**6.8. LOAD DATABASE SCHEMA**	
Actors:	Mapping Manager	
Purpose:	To load an existing database schema into to working environment	
Summary:	The user loads an existing database schema into the working environment. It will be used to create the virtual schema	
Preconditions:		
Main Flow:	**Mapping Manager**	**System**
	1.- The user selects to load a database schema	
		2.- The system shows a list with all available database schemas
	3.- The user selects the desired database schema	
		4.- The system loads the selected database schema, and incorporates it to the working environment
Exceptions:		

Use Case:	**6.9. LOAD DOMAIN ONTOLOGY**	
Actors:	Mapping Manager	
Purpose:	To load an existing database schema into the working environment	
Summary:	The user loads an existing database schema into the working environment. It will be used to create the virtual schema	
Preconditions:		
Main Flow:	**Mapping Manager**	**System**
	1.- The user selects to load a domain ontology	
		2.- The system shows a list with all available domain ontologies
	3.- The user selects the desired domain ontology	
		4.- The system loads the selected domain ontology, and incorporates it to the working environment
Exceptions:		

Use Case:	**6.10. MAP ATTRIBUTE**	
Actors:	Mapping Manager	
Purpose:	To establish a new mapping relation between attributes in the database schema and the virtual schema	
Summary:	The user creates a new mapping relation, which relates one attribute of the current database schema with one attribute of the current virtual schema	
Preconditions:		
Main Flow:	**Mapping Manager**	**System**
	1.- The user selects an attribute of the virtual schema, and an attribute of the database schema, and indicates the system to establish a relation between them	
		2.- The system creates a new mapping relation between the selected attributes
Exceptions:		

Use Case:	**6.11. MAP RELATION**	
Actors:	Mapping Manager	
Purpose:	To establish a new mapping relation between relations in the database schema and the virtual schema	
Summary:	The user creates a new mapping relation, which relates one relation of the current database schema with one relation of the current virtual schema	
Preconditions:		
Main Flow:	**Mapping Manager**	**System**
	1.- The user selects a relation of the virtual schema, and a relation of the database schema, and indicates the system to establish a mapping relation between them	
		2.- The system creates a new mapping relation between the selected relations
Exceptions:		

Feature 7: Unification of Virtual Schemas

Use Case:	**7.1. CREATE NEW UNIFICATION**	
Actors:	Unification Manager	
Purpose:	To create a new unification process, which will be able to include several virtual schemas for its future integration	
Summary:	The user asks the system to create a new unification, which will be empty at the beginning	
Preconditions:		
Main Flow:	**Unification Manager**	**System**
	1.- The user selects the option of creating a new unification	
		2.- The system asks the user for a name for the unification
	3.- The user inputs a new name for the unification	
		4.- The system creates an empty unification, with no associated virtual schemas
Exceptions:		

Use Case:	**7.2. UNIFY**	
Actors:	Unification Manager	
Purpose:	To add a new virtual schema to the unification, and perform de unification process	
Summary:	The users asks the system to include a new virtual schema into de current unification, and perform the unification process when it is done	
Preconditions:		
Main Flow:	**Unification Manager**	**System**
	1.- The User selects the *unify* option	
		2.- The system shows a list of existing virtual schemas
	3.- The user selects the virtual schema he wants to unify with the current unification	
		4.- The system performs the unification process with the current working unification. All virtual schemas previously included in this unification are unified
Exceptions:		

Use Case:	**7.3. DELETE UNIFICATION**	
Actors:	Unification Manager	
Purpose:	To erase an existing unification	
Summary:	The users asks the system to erase an existing unification	
Preconditions:		
Main Flow:	**Unification Manager**	**System**
	1.- The user selects to delete a unification	
		2.- The system shows a list with all available unifications
	3.- The user selects the unification he wants to delete	
		4.- The system deletes the selected unification
Exceptions:		

Use Case:	**7.4. SAVE UNIFICATION**	
Actors:	Unification Manager	
Purpose:	To save the current unification	
Summary:	The users asks the system to save the current unification	
Preconditions:		
Main Flow:	**Unification Manager**	**System**
	1.- The user requests the system to save the current unification	
		2.- The system reports that the unification has been saved
Exceptions:		

Use Case:	**7.5. LOAD UNIFICATION**	
Actors:	Unification Manager	
Purpose:	To load an existing unification	
Summary:	The users asks the system to load an existing unification	
Preconditions:		
Main Flow:	**Unification Manager**	**System**
	1.- The user selects to load a unification	
		2.- The system shows a list with all available unifications
	3.- The user selects the unification he wants to load	
		4.- The system loads the selected unification
Exceptions:		

Use Case:	**7.6. INCLUDE VIRTUAL SCHEMA IN UNIFICATION**	
Actors:	Unification Manager	
Purpose:	To add an existing virtual schema into a unification process	
Summary:	A virtual schema is included in the unification, from a list	
Preconditions:		
Main Flow:	**Unification Manager**	**System**
	1.- The users selects to include a virtual schema into the current unification	
		2.- The system shows the user a list of the existing virtual schemas
	3.- The user selects the virtual schema he wants to include	
		4.- The system includes the selected virtual schema into the unification
Exceptions:		

Use Case:	**7.7. REQUEST UNIFICATION**	
Actors:	Unification Manager	
Purpose:	The system performs the unification process	
Summary:	The user tells the system to perform the unification process with the current working unification	
Preconditions:		
Main Flow:	**Unification Manager**	**System**
	1.- The User selects the *request unification* option	
		2.- The system performs the unification process with the current working unification. All virtual schemas previously included in this unification are unified
Exceptions:		