A semantic and agent-based approach to support information retrieval, interoperability and multi-lateral viewpoints for heterogeneous environmental databases

Zuo, Landong

A SEMANTIC AND AGENT-BASED APPROACH TO SUPPORT INFORMATION RETRIEVAL, INTEROPERABILITY AND MULTI-LATERAL VIEWPOINTS FOR HETEROGENEOUS ENVIRONMENTAL DATABASES

Landong Zuo

A thesis submitted in partial fulfilment of
the requirements for the degree of

Doctor of Philosophy

The Department of Electronic Engineering
Queen Mary, University of London

2006

Declaration

The work presented in the thesis is the author's own.

DATE: _____

SIGNATURE: _____

:

QUEEN MARY,

UNIVERSITY OF LONDON

ABSTRACT

Data stored in individual autonomous databases often needs to be combined and interrelated. For example, in the Inland Water (IW) environment monitoring domain, the spatial and temporal variation of measurements of different water quality indicators stored in different databases are of interest. Data from multiple data sources is more complex to combine when there is a lack of metadata in a computation form and when the syntax and semantics of the stored data models are heterogeneous. The main types of information retrieval (IR) requirements are query transparency and data harmonisation for data interoperability and support for multiple user views. A combined Semantic Web based and Agent based distributed system framework has been developed to support the above IR requirements. It has been implemented using the Jena ontology and JADE agent toolkits. The semantic part supports the interoperability of autonomous data sources by merging their intensional data, using a Global-As-View or GAV approach, into a global semantic model, represented in DAML+OIL and in OWL. This is used to mediate between different local database views. The agent part provides the semantic services to import, align and parse semantic metadata instances, to support data mediation and to reason about data mappings during alignment. The framework has applied to support information retrieval, interoperability and multi-lateral viewpoints for four European environmental agency databases.

An extended GAV approach has been developed and applied to handle queries that can be reformulated over multiple user views of the stored data. This allows users to retrieve data in a conceptualisation that is better suited to them rather than to have to understand the entire detailed global view conceptualisation. User viewpoints are derived from the global ontology or existing viewpoints of it. This has the advantage that it reduces the number of potential conceptualisations and their associated

mappings to be more computationally manageable. Whereas an ad hoc framework based upon conventional distributed programming language and a rule framework could be used to support user views and adaptation to user views, a more formal framework has the benefit in that it can support reasoning about the consistency, equivalence, containment and conflict resolution when traversing data models. A preliminary formulation of the formal model has been undertaken and is based upon extending a Datalog type algebra with hierarchical, attribute and instance value operators. These operators can be applied to support compositional mapping and consistency checking of data views. The multiple viewpoint system was implemented as a Java-based application consisting of two sub-systems, one for viewpoint adaptation and management, the other for query processing and query result adjustment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The completion of this thesis is also attributed to several contributions. Firstly, I wish to express my sincere gratitude for all the support and help received from the department of Electronic Engineering, Queen Mary, University of London. Special thanks go to my supervisor Stefan Poslad for his continuous guidance and patient supervision throughout my Ph.D. study. This particular thesis would not have been possible without his help.

In addition, I would like to thank my colleagues from my department for their encouragement and motivation. I would like to thank John Bigham for his advice and support as my second supervisor and as my supervisor on the preceding MSc course. My appreciation also goes to other colleagues and friends, Karen Shoop, Juan Jim Tan, Leonid Titkov, Xuan Huang, Yong Zuo, Dejian Meng, Bin Li, Zekeng Liang, Iaonnis Barakos, and Bob Chew. Their company has made the journey easier.

The research work was carried in the EU FP5 EDEN-IW project. I am grateful to my colleagues in this project: Palle Haastrup, Jorgen Wuertz, Michael Stjernholm, Dominique Preux, Ole Sortkjaer, Athanasios Dimopoulos, Lisbet Sortkjaer, François-Xavier Prunayre and all the other people involved, particularly in the U.S. liaison. I have learnt so much from them.

I feel a deep sense of gratitude to my parents for their endless love that has guided all my visions and formed the most important part of growing-up. Thanks to my elder brother Weidong and his family for all their care and support. Thanks also to Jingrong who was always there when needed, helping me to face any difficulties.

Finally, I would like to thank for EU-IST EDEN-IW project (IST-2000-29317) and department of Electronic Engineering of Queen Mary, University of London for their support in funding for this research project. My research work has taken great advantage of the successful cooperation between the department of electronic engineering at Queen Mary and Beijing University of Post and Telecoms. I was a member of the first group of students under this international relationship in 2001.

GLOSSARY

ACL	Agent Communication Language
API	Application Programming Interface
COBRA	Common Object Request Broker Architecture
DAML	DARPA Agent Mark-up Language
DA	EDEN-IW Directory Agent (chapter 5)
DB	Database
DF	Directory Facilitator
DL	Description Logic
DSS	Decision Support System
DTD	Document Type Definition
EDEN	Environmental Data Exchange Network
EGV	EDEN-IW Global (data model) View
FIPA	The Foundation for Intelligent Physical Agents
GAV	Global As View
IR	Information Retrieval
IW	Inland Water
IOW	Information Office for Water
JADE	Java Agent Development Environment
JDBC	Java Database Connectivity
JSP	Java Server Pages
KR	Knowledge Representation
LAV	Local As View
LDV	Local Database View
MAS	Multi Agent System
NERI	National Environmental Research Institute
ODBC	Open DataBase Connectivity
OKBC	Open Knowledge Base Connectivity
OIL	Ontology Inference Layer
OWL	Ontology Web Language
RA	EDEN-IW Resource Agent

RAD	Rapid Application Development
RDBMS	Remote Database Management System
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SOAP	Simple Object Access Protocol
SQL	Standard Query Language
SWRL	Semantic Web Rule Language
UA	EDEN-IW User Agent
UDDI	Universal Description, Directory and Integration
WSDL	Web Service Description Language
XML	Extensible Mark-up Language

Chapter 1 Introduction

1.1 Motivation

Information Retrieval or IR is increasingly concerned with not only accessing data sources within a single or across multiple enterprise domains, but also with data interoperability and data integration between distributed, disparate data resources that were originally designed to be stand-alone. In addition it is concerned with the development of increasingly open information systems that can support multiple user types, applications and data sources. An open information system is advantageous so that new data sources can be added, unused ones can be removed, and the types of users and application can change dynamically with a degree of transparency. The process of data integration and data interoperability faces the following challenges:

- Data sources such as legacy databases have heterogeneous access interfaces that are oriented to stand-alone local use rather than to open system use. Transparent data access requires that data sources use a consistent vocabulary, syntactic structures and semantics. The documentation and on-line availability of such metadata (information about the data) in a machine understandable way to support automatic data access and data processing, are often omitted.
- User queries can often be processed more expediently by first querying the metadata information, i.e. the descriptions of information about the stored data, in addition to the normal data query. Metadata queries are often not supported in database systems.
- Evaluation of a general query may involve more than one data source. Sub-queries may need to be generated and directed to relevant data sources. This requires sufficient metadata description of the data contents for each data source that may not be available.
- The representation of content in data sources may vary between data models according to: structure, coding format, natural language and semantics. Integration and harmonisation of heterogeneous data is thus more complex.
- Different models of knowledge representation are used in applications and user groups. Information usage may vary with respect to: different levels of granularity,

different vocabularies, different scopes of a domain, different contexts of use and different perspectives.

- The differing representation expressivity in multiple types of data models may lead to information loss and restricted data operations between different data models, for example a relational database model is structured to be flat and data relations are constrained to support consistent data integrity whereas data in an Ontology data model is structured into class hierarchies and is constrained by class properties.
- The management of an open information system concerns data sources, users and applications that are autonomous and distributed. Information entities and data content can change dynamically. This may introduce new data inconsistencies, conflicts, and redundancies between different data models that were not present within the individual data models.

Heterogeneities among information entities need to be resolved to enable meaningful information exchange and to enable data interoperability [67]. In contrast, traditional database systems focus more on building individual homogeneous data models to satisfy specific data queries in a consistent manner, information heterogeneities are not well addressed. There is little support for online, accessible, metadata to enable data heterogeneities to be handled and for conceptual data structures and semantics to be presented and adapted, to be understandable to different users [63, 64].

If support for an explicit metadata model within a database to support transparent data access and data harmonisation for heterogeneous data is lacking, it could be supported in a model external to the database, yet linked to the data within the databases. The motivation for this is clear: not only can it be used to support data access transparency and data harmonisation but it could also promote data reuse and reduce the cost and complexity in developed integrated IR systems for different types of applications and users. There is an important design decision as to what conceptualisation and representation the metadata model should use, should it relate more to entities in the physical world versus those in the relational database.

There are a variety of approaches to model and interlink metadata to data [21]. A more expressive type of relational model could be used and interlinked to a separate knowledge based conceptual model of the real world or the relational model could be enhanced to support a more expressive knowledge conceptualisation of the world or a knowledge based model could be enhanced with relational database modelling support

[21]. Data consistencies, semantic consistencies, data constraints, and possible information loss must be carefully considered, when interlinking these two models or when combining them into a single data model.

Part of the database interoperability research in this PhD has been undertaken and applied as the author's contribution to EU-IST EDEN-IW or Environmental Data Exchange Network for Inland Water project. The model used for database integration in this project was developed by the author. The other main part of the PhD, to support multiple user views of data and to adapt the queried data to them, was undertaken outside the EDEN-IW project. The Inland Water or IW domain typically consists of distributed data source containing values of variety water quality indicators that are measured using different types of instruments, in different components of water and in different European geographical regions at a range of times.

Information systems for the Inland Water quality domain typically comprise a number of legacy databases that are developed independently and managed autonomously by national environmental institutes and agencies. These legacy database systems utilise different database management systems, data models, data structures and query mechanisms. Stored data can be represented in different scientific terminologies and even in different natural languages. Stored data representing physical, chemical and biological water quality measurements are correlated with other key concepts such as temporal and spatial relations in different ways. The information analysis that combines information from multiple sources can be used to discover and compare trends in the variation of environmental IW pollution indicators across the EU.

In addition, multiple user groups may have heterogeneous views over a domain conceptualisation. These user views can vary according to the scope of the domain conceptualisation modelled versus the conceptualisations that different types of users are interested in. Examples of different types of usage for IW data include their use by: policy-makers to compare water quality data across different national rivers, including those in cross-border areas; by scientists to test theories that explain the water quality variations and trends across space and time. Hence, different user views of the stored data need to be accommodated and terms and values need to be dealt with consistently across multiple user views.

1.2 PhD Focus and Objectives

The main PhD focus is to research and to develop a semantic approach to support heterogeneous information integration for the IW domain that involves machine-understandable metadata representations of data collected and stored in relational databases. A knowledge based semantic conceptualisation seems a good candidate model for this. The design of such a semantic model needs to be able to support, and to be reusable to reduce the development resources needed to support, multiple heterogeneous database resources, users and applications. A solution is needed that can deal with the complexity and data processing decisions in the mapping processes needed to handle queries about heterogeneous data and that may require heterogeneous data from multiple sources to be harmonised.

Objectives for this PhD have been specified with respect to the motivation given above as follows:

1. To survey, classify and model the information heterogeneities found when heterogeneous databases within a domain are integrated and to survey approaches to tackle these heterogeneities with particular focus on semantic based approaches.
2. To investigate and resolve the interoperability problems that may affect the use of a semantic mediation and data harmonization approach to combine heterogeneous database data. This objective can be further decomposed into:
    a. To identify the key effects of different types and combinations of information heterogeneities that hamper the interoperability amongst different information entities.
    b. To investigate the combination of semantic web with relational databases to improve the usability of the stored data.
    c. To resolve query transformation that involve different representations and expressivity of knowledge models in an information system.
    d. To investigate semantic-rich metadata services supporting query decomposition, data harmonisation and resource admission.
3. To investigate how to support information viewpoints and user queries that are oriented towards specific conceptualisations by users.

The focus of much computer science research is to develop ever more expressive Semantic models such as those based upon Ontologies, e.g., by adding support for

temporal constraints and more expressive logical inferencing. However, for information retrieval researchers and developers, it is more important that an Ontology representation is easy to maintain and integrate into conventional distributed information system infrastructures, so that it can be embedded into legacy information systems containing relational databases and interlinked and synchronised to legacy data – the use of the Ontology model is an enabler to enhance information retrieval.

1.3 Research Contributions

This research focuses on combining relational type of database information retrieval using Ontologies and multi-agent system techniques to resolve interoperability issues and forms a generic approach to information integration and representation by semantic means. The contributions are partitioned into two main parts with respect to: research and development of an Ontology-driven middleware service to mediate between information heterogeneities when integrating heterogeneous legacy databases; research and development of an Ontology based approach to support the projection, adaptation and validation of multiple user viewpoints over a common domain conceptualisation.

Regarding the integration part of work, the main contribution is to develop a more comprehensive solution to handle information heterogeneities and resolve the semantic mapping between different representations of domain knowledge. The novelty of such an approach is to hide the underlying details of information retrieval from legacy databases in a single domain and to project a semantic based single virtual information system to the user. It can support the reuse of stored data of relational database in compliance with different type of applications in a wider scope. The semantic meaning of terminology is analysed in terms of decomposition and processing of the user query. A core part of the database integration approach is the design of a partitioned multi-lateral Ontology model to support conceptual interoperability and information mediation. Information heterogeneities can be resolved at different levels using an Ontology-driven approach. A common Ontology model that reflects the common agreement of conceptualisation amongst domain experts is developed independently of, yet aligned to, the local data sources and applications. Database integration is achieved using both static and dynamic data transformations. Firstly, by using static transformations of a common or global semantic knowledge representation that maps related semantic correspondences of the conceptualisation. Secondly, by using a

dynamic query transformation and answering approach to answer query instances across different database models, using the global conceptual model as a mediator to support data transformations. Access transparency and data harmonisation are enhanced by an approach that supports semantic reasoning. The reasoning functions of a graph-based algorithm traverse through the interlinked ontologies to discover mismatched constraint relations.

The partitioned model of multi-lateral Ontology supports an open information system model, in the sense that there are well defined system processes for wrapping new heterogeneous database data, integrating them and supporting more abstract user representations that relate to the real physical world conceptualisation. Information mediation uses flexible semantic mappings when queries are expressed using a common Ontology that are passed to the distributed local Ontology models and then transformed into SQL commands. Information heterogeneities can be resolved in a comprehensive manner at multiple levels. Support for query transparency and data harmonisation has been achieved and demonstrated. The control and management of the metadata to support interoperability is decentralised to cope with the connection of new databases that use new database schema.

The second main contribution is to support flexible customisation of queries and the corresponding retrieved results that can be oriented towards specific user views, thus significantly improving the usability of IR systems. A specific process is defined to orientate the formation of user query with respect to the terminology, conceptualisation and preferences of a particular individual user or user group. This is again facilitated by a common Ontology model that has been extended to support user conceptualisations, terminologies and preferences. Concept customisation occurs with respect to both user group or user stereotypes and with respect to the individual user preferences. The user group viewpoint representation model uses an extended global-as-view approach, coupled with the use of logic inference, to validate data query consistency across conceptual views using the common conceptual model as a mediator. The semantic representation of user preferences is structured into a sub-Ontology that can also represent additional constraints associated with a particular group viewpoint.

The distribution and exchange of semantic and meaningful information is achieved using a Multi-Agent type distributed system infrastructure. A versatile information service has been built to enable sharing semantic messages concerning the use of the

multi-lateral Ontology model and to support semantic-based directory enquiries and task management. The semantic information is enclosed in an Agent Communication Language message as its payload.

The research methodology of this PhD was applied as part of the EDEN-IW project to integrate heterogeneous information in the Inland Water domain consisting of four national databases containing more than two million real water-quality records.

Arising out of this PhD research to date, there have been the following types of research publication, listed in Appendix I: one journal publication, four conference publications, three book chapters and three public project deliverables (available via the project web-site).

1.4 Thesis Outline

The remainder of this thesis is organised as follows. Chapter 2 introduces the background knowledge for an Information Retrieval systems project based upon methods and architectures for integrating multiple heterogeneous database sources. Section 3 surveys selected related work, it analyses the strengths and limitations of existing approaches and highlights the strengths of the Ontology-driven approach that is developed in this thesis. Chapter 4 describes an Ontology-driven integration method developed that consists of a partitioned multi-lateral Ontology model, semantic mapping services and a multi-agent infrastructure to enable the exchange to access and to manage different data sources. Chapter 5 extends the framework from chapter 4 to support information adaptation of the domain conceptualisation to facilitate multiple user viewpoints over an integrated information domain. A computational model is proposed to support this that can be underpinned with a formal logic framework. Finally, chapter 6 discuses the merits of the approach adopted, considers some important limitations of the approach leading to further work and presents the final conclusions.

Chapter 2 Background

This chapter gives a general review of relevant technologies and background knowledge concerning the integration of multiple heterogeneous users, applications and database sources for distributed IR systems.

2.1 System Architectures for Information Retrieval (IR)

2.1.1 General Architectures

Architecture models are a high-level model of the structure of a system in terms of computational nodes and the links that interconnect them. Garlan and Shaw [39] were two of the first researchers to generally classify system architectures into a set of main types according to the different types of nodes and links:

- Layered systems: organise nodes hierarchically with lower layers providing services to higher layers above it. A layered system model is often a good high-level model for partitioning the main functionality of the system.
- Object-oriented models: nodes are objects that encapsulate functions and offer these functions for invocation at well-known interfaces. In order to invoke a function in an object, a reference must be obtained to that object first.
- Event-based systems: events and messages can be exchanged once event receivers register their interest for events with event senders.
- Repositories: have two distinct styles of nodes: a central data store and a collection of independent components that operate on this store. There are two main sub-types of repository architecture: a (relational) database in which external applications make queries to data structured in tables and knowledge-base system in which knowledge based processors send, receive and process knowledge stored in a knowledge repository.

In practice, most architecture models, for database middleware are hybrid architectures. Database IR systems are generally partitioned into database resource management, application processing and presentation horizontal layers, see Figure 1. The database sources themselves are considered to be below the middleware. At a lower level of abstraction of the middleware model, the layers consist of service objects and agents that can interact using message-passing. A knowledge repository, based on an

Ontology model, forms an integrated meta-data model to interlink the database resources, the resource users (applications and human users) and resource processors.

2.1.2 Layered Information System Architectures

At the conceptual level, the design of information retrieval system categorises three layers: presentation layer, application logic layer and resource management layer, see Figure 1. The presentation layer interacts with the external entities to present the information to the clients. The application logic layer deals with the data processing to reflect the particular business objective and usage. The resource management layer deals with and interfaces to the different data sources of the information system, independently of the nature of these data sources such as databases, file systems or other information repositories [10].

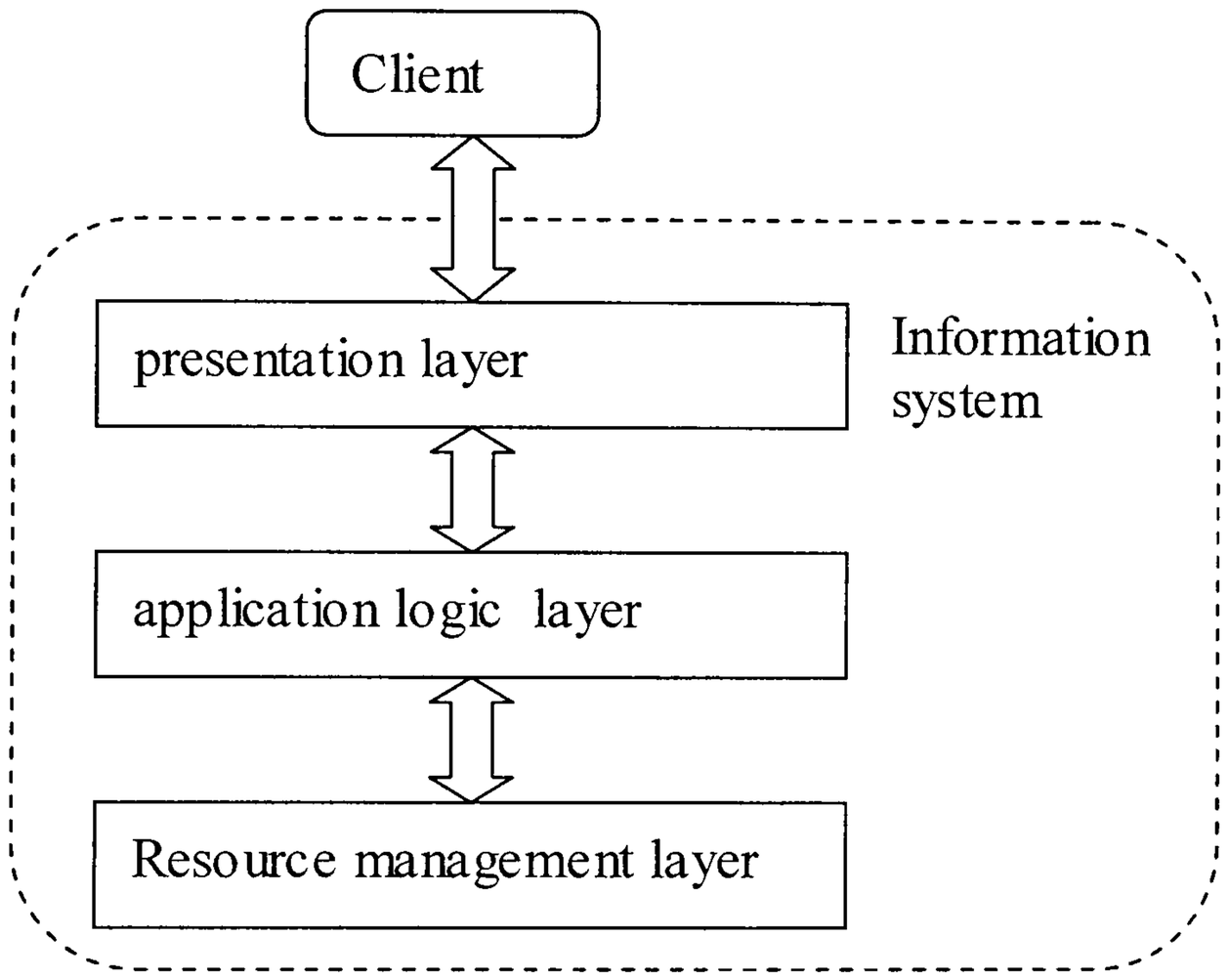

Figure 1 A layered information retrieval system model

Functionalities in these tiers can be combined, split further and distributed in deployed systems. In practice most complex distributed system are 3-tier or n-tier systems depending on how the tier abstractions are defined.

2.1.3 Client-Server 2-Tier IR Systems

A 2-tier distributed system typically consists of clients and servers. The server merges the functionality of the resource management layer and application logic layer into one tier, while a client contains the other tier, the presentation layer, combining to form a so called thin-client server system. Alternatively, the application logic can be

accomplished at the client side; then the client program becomes a fat-client server system containing a wide range of complex functionality. Low level syntactic data communication between client and server based upon RPC or Remote Procedure Calls and socket programming used to be wide spread in this type of architecture. However, the client interaction is steadily becoming based upon Web services and XML (see below).

Database servers may need to support a heavy data processing load depending on the number of records and concurrent users it supports. The data traffic between the client and server may also be very heavy if client data queries return large data sets. Hence, an important part of the system design may be to handle the retrieval in large data sets in different ways such as batching them, filtering them and reducing them. The functional implementation or data query application is often designed to be tightly coupled to he stored data and to the business logic rules. The latter may often not be explicitly modelled and available for online computation, thus making data and their application processing logic to be reused or enhanced.

2.1.4 3-Tier IR Systems

Due to changing requirements in the problem domain, the client program may need to be able to connect to multiple applications thus data presentation may need to be designed to be application independent. Information applications may also need to access multiple data resources. IR system is expected to support these variations. A 3-tier the architecture clearly separates the presentation, application processing and resource management into three component tiers:

1. Presentation Tier: the front-end that is responsible for providing portable presentation logic;

2. Data Resource Tier: the back-end that provides access to dedicated data storage services, such as a database server.

3. Application Tier: the middle-tier component that allows users to share and control business logic by isolating it from actual data and users.

Communication between the presentation and application tiers used to be based on standard interfaces, such as the CORBA[76] or Common Object Request Broker Architecture from the OMG or Object Management Group and RMI or Remote Method Invocation type program interfaces, but these are also being replaced by Web services and XML. Applications in the middle-tier talks to the database back-end using the open database access interfaces such as ODBC or Open Database Connectivity and that can wrap SQL or Structured Query Language commands making use of the additional metadata support in OKBC[5] to allow processes to loop through data sets. Separation of business logic rules from the data storage and presentation makes the maintenance and developing much cheaper, as the access to the different application system is more flexible to cope with the requirements of reusability and compatibility. Some typical application systems using 3-tier architecture are federated databases, multiple databases and data warehouse systems. N-tier architecture is extension of the 3-tier system in order to fit the requirements of connectivity of different system through internet. The addition of new application systems can create more application tiers such as directory services and make the application logic more complex.

2.2 SQL-based Distributed Databases and Data Warehouses

2.2.1 SQL

SQL is the current standard for querying data from all major RDBMS or Relational Database Management Systems. In theory, distributed databases can transparently join data from different databases enabling queries to be applied across different databases. SQL is by definition a query language. Its power is as a data verification technique; it uses pre-determined queries and verifies the query in terms of whether results will be returned to answer that query or not. SQL uses simple textual search operators like NOT, LIKE or EQUALS, but these are syntactical operations. SQL and the relational model lack the inference capability and a semantic model in order to relate different data sets on-the-fly. In some cases, the user may not know the exact queries to retrieve the data, or which tables contain the relevant data, or even which databases contain the relevant data. The user may need to do a more general search to select data rather than to use prior knowledge to make specific queries. Searches are more efficient if they are made on metadata rather than the data itself. SQL supports meta-data and these can be stored as tables in the database. SQL queries can then be used to query the meta-data

tables in the same way that they can be used to query the data tables thus supporting rudimentary searches. However there are several limitations that restrict the use of SQL for searching databases rather than querying such as: a lack of a commonly used specification for metadata syntax and semantics; lack of provision of metadata in individual database instances and lack of a standard namespace to locate tables within a database and to locate tables across multiple databases.

2.2.2 Database Federation and Distributed Databases

The idea of a federated database is that databases could be loosely linked together so that data from them could be combined, but there is a lack of specific models to support this in any standard way.

A distributed database system enables multiple databases to exist at multiple locations but to be queried as if they were centrally located and without the need to export partial copies of data to a common data (warehouse) store. Distributed databases can transparently join multiple distributed data that is fragmented and replicated across multiple databases. But a major restriction for the fragmentation and rejoining to work is that data fragments need to have the same data schema (horizontal fragmentation) or for data schema to be a sub-set of another (vertical fragmentation). Hence this is not usable if data schemas in different databases are not compatible in this way. Distributed databases are supported as extensions to existing RDBMS.

2.2.3 Data Warehouses

A data warehouse follows the repository architecture style and is used to integrate related sub-sets of data extracted periodically from multiple databases and stores them centrally in the data warehouse. Data warehouses are primarily used for analysis in comparison to databases which are primarily used for on-line transaction processing and data queries. Data warehouses collect a subject-oriented, integrated, time-variant and non-volatile set of data, usually for further analysis, as input into management decision making processes [46]. Data Warehouses focus on pulling and processing huge amounts of data, periodically, according to specific logic rule and business objects in order to provide multi-view results for different user groups.

Three conceptual layers form part of the data warehouse design. In resource management, data is periodically imported from different data resources. The individual databases must be prepared to give up some of the autonomy to give up copies of large sub-sets of their data to data warehouses for processing under the control of the data warehouse. Whereas data in databases can be the result of up to date transactions, data in warehouses is typically refreshed daily and so the latter's data is less fresh. Data imported into a data warehouse needs to be cleaned and transformed so that data integrity is maintained across the data sets from the different databases. Data from the individual databases is integrated at the syntactical level according to a star or snow-flake schema pattern that forms the design of the stored data in the data warehouse. In the business logic part, business rules and application logic are used to post process and analyse the data along specific application dimensions such as time, region and type of water quality indicator. Another key difference between databases and data warehouses is that a warehouse processes and views data along more than two dimensions, such as along three and six dimensions. In the presentation part, data can be transformed and presented to support different user views of the stored and analysed data.

Metadata, described before as data about data, needs to be explicitly defined and presented in an on-line computation form in data warehouses. It is needed to define the data sets to be imported from the individual databases. The metadata needs to contain the information to describe how to transform and relate the individual databases data into a whole, according to data warehouse schema. In the 2000s, standards are emerging for managing the warehouse metadata such as CWM or Common Warehouse Meta model from the OMG group. This is based upon XML, for the on-line data representation, UML or Unified Modelling Language, for the data design and CORBA [76]. However, interoperability is still complex to achieve and uses proprietary and manual processes to create and manage the data in practice, especially when multiple databases from heterogeneous vendors within the same application domain use different terms including multi-lingual terms and use multiple different schema to represent the same sub-sets of data. Further the underlying OMG CORBA architecture and the use of an abstract definitional language to specify services appear to have lost ground to XML and SOAP Web Service models. A competing approach to CWM is

OIM, the Open Information Model, from the Meta Data Coalition (MDC) led by Microsoft.

2.3 Web based Portals

A Web-based portal consisting of a Web browser front-end to offer query forms and results, a Web Server to execute the database applications and middleware to connect to database back-ends is now becoming a common IR system architecture. The portal connects to the Web server by sending data structures, over HTTP. The Web server connects to the database server by using an OKBC interface to embed SQL commands and send them over a TCP/IP connection. The application logic and presentation logic are embedded into the web server and form the middle tier. The user query is interpreted into the SQL statement at the web application and then be sent to the back-end database server for processing. Thus the user can have easy access to the multiple stand-alone databases via web-pages. User queries are usually formulised according to the predefined query templates.

2.3.1 XML

Although HTML, Hyper Text Markup Language, is by far still the most common representation language for content made available by the Web, HTML lacks any ability to define user-define data structures for its content and is less able to separate the data structure in the content from presentation forms to provide more flexibility for presentation the same data according to different user views. The ability to support structured data and flexible presentation are key requirements for IR systems and these have driven the development of the XML or eXtensible Markup Language standard from the W3C group.

XML is a mark-up-language that supports the definition, transmission, validation and interpretation of data. XML is one of the components required to exchange information in a universal format but is not the ultimate solution for integrating heterogeneous databases. Agreeing a common syntax for structured data exchange, can be argued, is the easy part. Agreeing a common domain model of terms and their relationships is the hard part. Frequently there are multiple XML specifications for a given application domain. XML itself supports linearised hierarchical data structures. but its simplicity leads to ambiguities in interpreting terms and it lacks the expressivity to support inference, to explore and match data structures to support interoperability.

XML based extensions, such as RDF and DAML, see below, support richer inference, but lack maturity and are still not widely used in practice. Explicit communication protocols are still emerging. Most XML data exchanges use an implicit simple message template that includes both the request and reply in the same message. Richer interaction patterns and communication protocols are needed to adaptively match user requests to service capabilities, to support service push as well as service pull and to support multi-party interactions and negotiation.

XML is used to provide the syntax to encode the exchanged agent messages. XML alone is insufficient to act as a metadata model to be used to search and integrate heterogeneous IW databases because of its lack of expressivity to describe the semantics of the data and to support reasoning about the data.

2.4 Web Services and the Grid

2.4.1 Web Services

IR systems need more than a data exchange model such as XML, they need services and communication protocols to describe data resources, to advertise and search for particular data resources and to support more complex processes that can use multiple data queries and post-processing operations to combine data from multiple databases. There are a wealth of Web service models and specifications proposed by the W3C standards consortium and others to define additional message-passing protocols based on XML that can be used to provide additional services to support IR. These include: the Simple Object Access Protocol or SOAP for XML-based message exchange, the Web Service Description Language or WSDL, directory services based upon Universal Description, Discovery and Integration or UDDI and declarative models for specifying sequential patterns of XML documents that relate to business processes such as the Business Process Execution Language or BPEL [87]. Both open-source and commercial implementations of Web services are available. The main support for data integrity in Web services and the XML community is to use encryption type techniques and data signatures to support data exchange confidentiality and integrity checks.

2.4.2 The Grid

Data Grids [37] are emerging as an important middleware model for managing data in a range of scientific and engineering disciplines that require computationally intensive analysis of large quantities of subject-specific data. The term "Grid" refers to technologies and infrastructure that enable coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organisations. This sharing relates primarily to direct access to computers, software, data, networks, storage and other resources, as is required by a range of collaborative computational problem-solving and resource-brokering strategies emerging in industry, science, and engineering.

A Data Grid system consists of a set of basic Grid protocols used for data movement, name resolution, authentication, authorisation, resource discovery, resource management, and the like. A Data Grid provides transparency in how data-handling and processing capabilities are integrated to deliver data products to end-user applications, so that requests for such products are easily mapped into computation and or data retrieval at multiple locations. The focus of the Grid software community is defining APIs at the Grid level to access databases. More recently the Grid community have based their architecture upon XML Web-service models to access and process data.

2.5 Semantic Web and Ontology Models

2.5.1 Semantic Web

The Semantic Web is a Web of actionable information - information derived from data through a semantic theory for interpreting the symbols. The semantic theory provides an account of "meaning" in which the logical connection of terms establishes interoperability between systems [84].

The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. The aims of Semantic Web are to structure the information in all kinds of data resource and applications and to promote more automatic machine-readable data and processing and hence improve IR efficiency. "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [20]. XML-based Ontology languages have been also

proposed as Web based knowledge description languages [42]. Figure 2, taken from [84] shows the proposed layers of the Semantic Web, with the higher level languages using the syntax (and semantics) of the lower level languages. This thesis focuses primarily on the Ontology language level, and the sort of agent-based computing that they enable. Higher levels (with complex logics and the exchange of proofs to establish trust relationships) will enable even more interesting functionality.

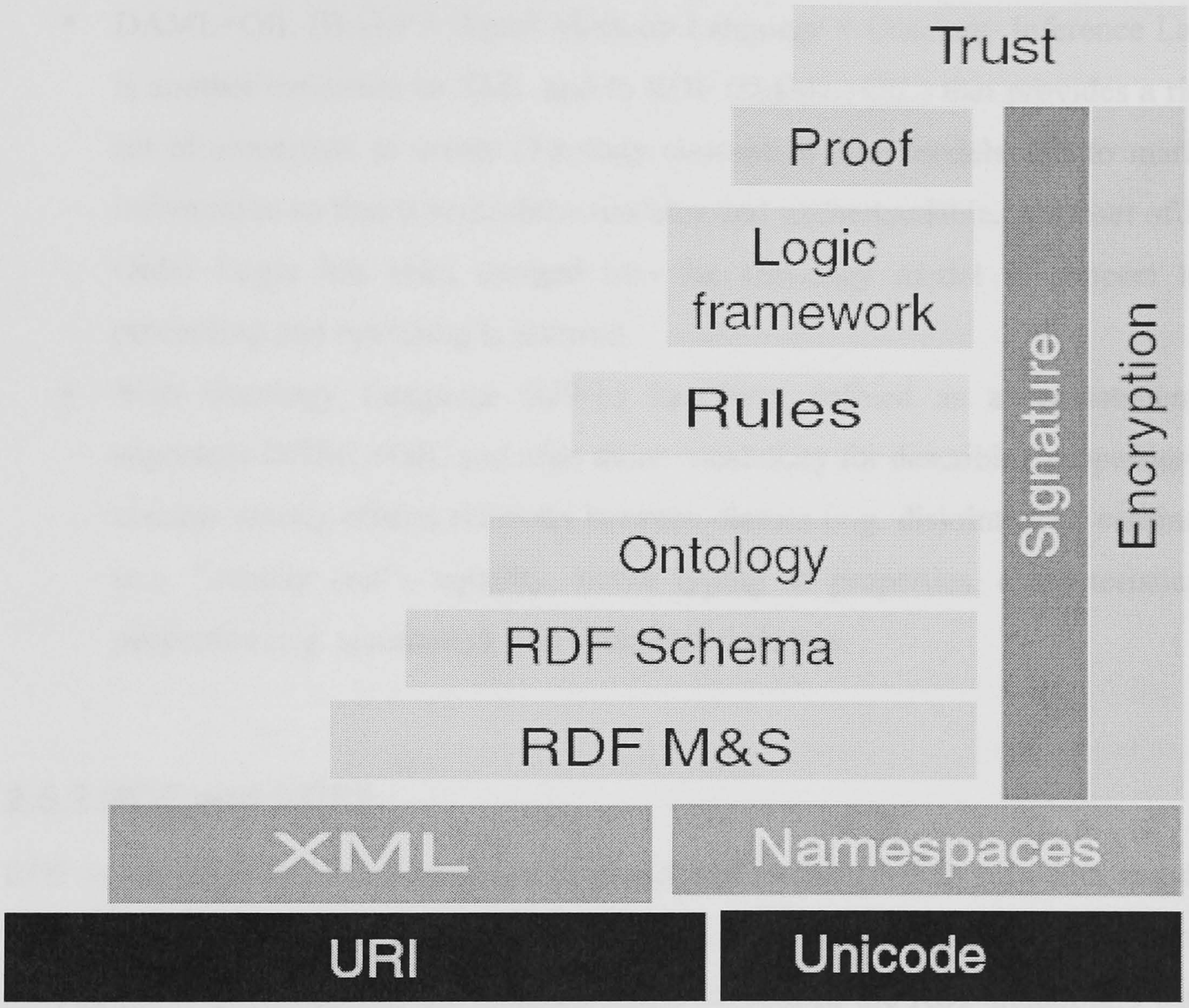

Figure 2 The Semantic Web layered mode as presented by Tim Berners-Lee in 2003, taken from [19]

Some of these levels in more detail are:

- Extensible Markup Language (XML) provides the syntax for structured documents, but imposes no semantic constraints on the meaning of these documents.
- XML Schema is a language for restricting the structure of XML documents.

- Resource Description Framework (RDF) is a metadata model for defining data structures called resources and relations between them and provides a simple semantics for a data model whose syntax is XML.
- RDF Schema or RDFS is a vocabulary for describing properties and classes of RDF resources that supports a more expressive semantics for generalisation-hierarchies of such properties and classes.
- DAML+OIL (DARPA Agent Mark-up Language + Ontology Inference Layer) is another extension to XML and to RDF (DAML+OIL) that provides a richer set of constructs to create Ontology conceptual data models and to mark-up information so that it is machine readable and understandable. A subset of First Order Logic has been merged into the Ontology model to support logic processing and operating is allowed.
- Web Ontology Language (OWL) has been defined as a replacement to supersede DAML+OIL and adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes.

2.5.2 RDF and RDFS

RDF is the W3C proposed language to model and exchange both metadata and data. More specifically the metadata is modelled as a resource, a concept that is universally addressable. Statements are the main metadata concept in the RDF model and can be used to link two resources together. Hence statements specify triples of a verb (or predicate or property) that links a subject resource to an object or value. The verb may also be specified as a resource. Hence triple statements specify subject-verb-object or subject-predicate-value relationships. Each RDF statement can be stored as a relational database table whose name is the predicate and whose subject-value instances form the rows in the table. The advantage of using RDF rather than a relational data model to model and store metadata include:

- RDF is a standard to exchange metadata – there is a standard XML syntax for RDF.

- RDF can be used to combine data A with other data B that doesn't fit the model of data A, e.g., add an alias name.
- RDF can easily link to data and to add metadata stored elsewhere, e.g., other databases
- RDF can serve as base for higher-level languages that can describe vocabularies and establish the usage of terms within the context of the specified vocabulary (ontologies).

RDFS (RDF Schema) is a language for describing ontologies. RDFS defines basic classes for resources, properties, literals, containers, container member properties and classes of properties such as sub-classes, domains, ranges and labels. RDFS supports many of the above properties and can be considered an Ontology language. However, RDFS was never issued as a final recommendation by the W3C. A reworking of RDFS called the RDF Vocabulary Description Languages has in 2004 developed as a proposed specification but this wasn't available in time.

2.5.3 Ontologies

Ontologies are conceptual models that can be used for knowledge sharing. An Ontology is characterised by the explicitness of the conceptual model and richness of the structures used, to represent and manage knowledge, information and services. The model and the structures will also influence the degree of flexibility of the computation or inference that applications can derive from it. Sowa [88] defines an Ontology in the following way: "The subject of Ontology is the study of the categories of things that exist or may exist in some domain. The product of such a study, called an Ontology, is a catalogue of the types of things that are assumed to exist in a domain of interest D from the perspective of a person who uses a language L for the purpose of talking about D. The combination of logic with an Ontology provides a language that can express relationships about the entities in the domain of interest".

Unlike data models, ontologies are usually formed to be relatively independent of and reusable across particular applications, i.e. the Ontology consists of generic knowledge that can be used by the different kinds of applications and tasks [70].

There are many proposed Ontology models. Regardless of the properties of the specific Ontology, ontologies in general include the following elements:

- Taxonomic relations between *classes*
- *Datatype properties*, descriptions of attributes of elements of classes
- *Object properties*, descriptions of relations between elements of classes
- *Instances* of classes and properties.

Data type properties and object properties are collectively referred to as the *properties* of a class. A set of assertions about the loaded into a reasoning system is called a *knowledge base* (KB). These assertions may include facts about individuals that are members of classes, as well as various *derived* facts, facts not literally present in the original textual representation of the Ontology, but *entailed* (logically implied) by the semantics of the particular Ontology language. These assertions may be based on a single Ontology or multiple distributed ontologies that have been combined using defined mechanisms. Semantics is the set of formulised concept and relations that have been defined to describe the logic representation with the given restriction so that the logic application can read, understand, process and deduce the logic relations from the defined the knowledge base in order to answer information queries in a more intelligent way. Most of applications are designed to handle the case in the particular domain and application. The logic inference, reuse and reasoning in such application are quite limited [81] .

There are many Ontology representations that can be chosen. Ontologies started to gain widespread interest and support as part of an initiative called the Semantic Web. The Semantic Web covers a range of XML-based approaches such as RDFS, as it supports the above Ontology features, DAML+OIL and OWL. At the start of the PhD in late 2002, DAML+OIL was the most widely used and supported Ontology Model.

2.5.4 Description Logic

Description Logics or DL have several key features that make them attractive as Ontology languages [59]:

- ***Expressivity*** DLs are highly expressive, enabling rich and complex descriptions of domain concepts. Concepts can be defined in terms of their properties and their relationships to other concepts. It is not necessary to use all of the expressive power of the DL, some or all of the Ontology can be represented as a simple taxonomy.
- ***Automated Reasoning*** DLs are logics so that there is a clear understanding of the language's formal properties. This enables the development of reasoners, i.e. software that is capable of checking ontologies for consistency and inferring that one concept is a kind of another concept. This latter characteristic means that the concept hierarchy can be inferred based on the content of the Ontology instead of being handcrafted by the ontologist.
- ***Compositionality*** The previous two properties enable the building of ontologies in a compositional way, i.e. by making new concepts from combining previously defined concepts and properties. This means that it is unnecessary to predetermine and enumerate all the concepts of the Ontology beforehand, making the process of building large ontologies more manageable and flexible.

OWL is developed as a vocabulary extension of RDF (the Resource Description Framework) and is replacement for the earlier DAML+OIL Web Ontology Language. The proposed OWL language actually consists of three subsets of language: OWL-Lite, OWL-DL (Description Logic) and OWL-Full. OWL-Lite and OWL-DL provide the basic DL constructs combined with RDF syntax, whereas OWL-full is more expressive and complicated with less restriction to support RDF syntax with logic operator. The difference between OWL-Lite and OWL-DL is that OWL-Lite only provides a basic subset of constructs for representation use of OWL syntax, while OWL-DL provides a language subset that has desirable computational properties for reasoning systems[15]. The OWL-Full allows free mixing-use of OWL and RDF syntax, which makes the formal inference more complicated.

From the perspective of effective representation reasoning, this thesis mainly uses OWL-DL as Ontology representation language, whereas some parts of Ontology was implemented in its precedence DAML+OIL. The Ontology entailment of OWL-DL can be reduced to Description Logic Satisfisbility problem using a subset of Description Logic. SHIOQ[45].

Description Logics (DLs) are a decidable subset of First Order Logic. It is the most recent name for a family of knowledge representation (KR) formalisms that represent the knowledge of an application domain (the "world") by first defining the relevant concepts of the domain (its terminology), and then using these concepts to specify properties of objects and individuals occurring in the domain (the world description)[11]. Semantics of DL represents the subsumption relations in a four-tuple consisting of abstract domain, concept names, property names and individual names. A knowledge base of description logic consists of two components: TBox and ABox. TBox indicates the extensional data, i.e. all terminologies in the abstract domain. ABox asserts all named individual in terms TBox vocabularies. The reasoning service based upon DL knowledge base can inference implicit knowledge from explicit representation of logic axiom and facts in the knowledge base.

The primary building blocks of DL are the atomic concept (unary predicate), atomic role (binary predicate) and individuals. The formal semantic of atomic concept and atomic role can be defined as an interpretation I consists of a non-empty domain Δ^I and interpretation function, which assign to each atomic concept C a subset $C^I \subseteq \Delta^I$, and assigns to each atomic role R a binary relation $R^I \subseteq \Delta^I \times \Delta^I$. The compositional concept and role can be represented in a combined form of atomic concepts and atomic roles using logic operator such as negative, interaction, union, existential restriction, and universal restriction. There is much debate about whether or not further operators are needed. Other feature operators that may be introduced into DL to form a subset of representation language include cardinality restriction, transitive relations and inverse relations.

An OWL-DL model with non-cycle RDF syntax can be successfully mapped to description logic for inference and reasoning where decidable computation can be guaranteed under NP-Complete time. An OWL DL Ontology is translated into a SHIOQ knowledge base by taking each axiom and fact in the Ontology and translating it into one or more axioms in the knowledge base [44] such that the optimal algorithm of formal logic reasoning can be implemented in practice.

2.6 Multi-Agent Systems

An agent is a software abstraction that supports the properties of reactivity, proactivity, deliberation, social interaction and autonomy between other agent-based computation peers that may not necessarily be organised hierarchically as in a client-server distributed system architecture. Agents can autonomously monitor their own environment and takes action as they deem appropriate. These characteristics of agents make them suitable for applications that can be decomposed into independent processes. They are capable of doing useful things without continuous direction by other processes or users. The autonomous ability coupled with an intelligent behaviour is further enhanced in a Multi-Agent System or MAS.

A MAS is a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity. More recently, the term multi-agent system has been given a more general meaning, and it is now used for all types of systems composed of multiple autonomous components showing the following characteristics [47]:

- An individual agent has incomplete capabilities to solve a problem
- There is no global system control
- Data is decentralised
- Computation is asynchronous
- Agents socialise with each other either to cooperate or to compete.

An information agent is an agent that has access to at least one and potentially many information sources, and is able to collate and manipulate information obtained from these sources in order to answer queries posed by users and other information agents. A Cooperative Information System (CIS) is considered as a cooperative multi-agent system integrated by a set of agents, data, and procedures working, in a cooperative way, to support daily activities in the organisation. They have a common goal, exchange information, and work together in order to achieve their objective.

Agents can socialise using a rich set of standard interaction patterns. Communication enables the agents to coordinate their actions and behaviour, resulting in systems that are more coherent. Coordination involves cooperation, planning (centralised and distributed) [95]. Agent communication also involves knowledge exchange using a higher-level semantic model that is often based on ontologies. A multi-agent system is

a good potential architecture for integrating heterogeneous databases in that agents are naturally distributed and autonomous; they can use rich explicit communication protocols to interoperate and they can naturally link to semantic models to help resolve interoperability problems. Multi-agent systems have been and are the subject of a very active research community.

The first types of MAS where closed distributed systems in the sense that agents in one type of MAS were unable to understand or interact with agents from another type of MAS. Examples of these include:

- *InfoSleuth* [70-72] provided middleware in terms of an agent shell that includes a white-page directory service (library), an autonomous composite component, called the conversation layer, which provides routing, message-forwarding and basic dialog management, and a broker agent component. The agent system was implemented in a Prolog like language called LDL++[99]. Infosleuth was the MAS used by the forerunner project EDEN.
- *JATLite* (Java Agent Template Lite) system [3] provides Java middleware libraries, called layers, for basic communication service, a combined routing and message forwarding autonomous component or 'active library' and an agent communications library. The libraries can be substituted with alternatives. For example, the default basic communication library supported only TCP/IP transport not UDP/IP nor CORBA but it can be substituted by an alternative which supports these alternatives. Similarly, the agent communication library supported KQML by default but other alternatives can be supported.
- *KAoS* (Knowledgeable Agent-oriented System) system [25] was designed to be independent of a particular communication service. Several types of communication service "have been investigated" such as OMG's CORBA, IBM's SOM, Microsoft's COM and Java socket model. All KAoS agents are derived from a generic agent class (template-library type of middleware), which provides basic communication mechanism. Several important agents may play a persistent role but it is not clear whether this is implemented as middleware. Specialised middleware agents carry out other generic services such as a matchmaker (yellow-pages), domain manager (keeps track of ownership issues, white-page service), proxy and mediation agents act as external interfaces to the agent platform.

- *OAA* (Open Agent Architecture)[61] middleware system consists of an agent component called a facilitator, which provides yellow-page directory, persistence and co-ordination services. OAA also provides an agent library, implemented in several languages such as Prolog, C, Java, Lisp, Visual Basic and Delphi, which is linked to each agent and offers the agent communication service, via the facilitator. The communication language is proprietary called ICL and has a Prolog like syntax.

There are however interoperability problems, none of these proprietary MAS is able to interoperate with each other. Further, few of these proprietary MASs, if any, are open source. The highly interactive nature of multi-agent systems points to the need for consensus on agent interfaces in order to support interoperability between different agent systems in order for MAS applications to become pervasive. Whilst it is challenging to develop MAS applications for a closed vertical architecture and market, it is even more challenging and necessary to develop MAS for horizontal MAS markets and open services.

In the late 1990s and early 2000s, FIPA, the Foundation for Intelligent Physical Agents, led a community effort to develop the first standard specifications for agent communication languages or ACL based on speech acts. FIPA focused on specifying external communication between agents rather than the (internal) processing of the communication at the receiver. Several open source implementations of the core FIPA specification have developed and these include JADE, FIPA-OS, ZEUS and a Java Community Process or JCP specification, JSR00087, for agents called JAS, Java Agent Services with subsequent implementations [77].

2.7 Database Integration Models

One core focus of this research project is to support IR from heterogeneous databases within the IW domain. Designs are needed to make the integration of heterogeneous databases transparent to the user. There are several different types of metadata systems for integrating databases, classified according to whether they are syntactical versus Semantic or logical.

Syntactical:

1. SQL: based Global schema, federated schema based models.
2. XML / Web based models.

Semantic:

3. XML and RDF Semantic Web or Ontology based models.

The use of a specified data model is not in itself enough to integrate data. Communication protocols and services are needed to manage the life-cycle of meta-data in general from creation, to operation to data becoming obsolete and to support the more specific data management tasks for exchange, mediation and browsing needed to support heterogeneous data integration.

2.7.1 Database schema based Integration

A database schema is another example of meta-data, e.g., a database schema is meta-data about the database structure. There are two main approaches to database schema based integration: federated schema and global schema [85]. In the federated schema approach, each database supplies an export schema, a proportion of its schema that is willing to share, for others to import. Whilst in the global schema, each local database's schema is combined into a single integrated schema. There are questions about the scalability of schema-based approaches, including data warehouses, because of the number of possible heterogeneous schemas and the difficulty in normalising numerous syntactical mappings between heterogeneous database schemas. As a result interoperability based upon models of the semantics of the underlying databases has been proposed [53]. Thus the problem of resolving differences in structure is reduced to the problem in understanding the differences in the semantic models of the different databases and then integrating the individual semantic models into a common semantic model such as an ontological model.

A further problem with syntactical approaches is the lack of computable on-line representations of the meta-data schema. Generally the database design models are in a graphical format such as E-R or Entity Relationship type diagrams and not in a form for computation and automated processing.

In addition, as there is a lack of a global namespace or even a database wide namespace to address the individual database, there is no standard service or method to for browsing to locate data within in a database or to locate a database whose location is unknown. Users are required to master use of SQL to make queries. Some SQL queries are fairly complex, e.g., to find common elements between tables (the equivalent of the relational algebra divide operator).

2.7.2 XML based Integration

XML is more of an extensible language for syntax and representation of data rather than being a meta-data model in itself. XML can be used to define a syntax for SQL queries and for the tables that result from the queries. At this level, the XML syntax suffers from the same limitations as using non XML syntactical approaches. One proposed standard for database metadata that is the OMG Common Warehouse.

Other limitations to the database schema and XML syntactic approach is that they do not define semantics of the data collected.

2.7.3 Semantic based Integration

In an information retrieval (IR) application, ontologies are used to guide the search so that the system may return more relevant results. The assumption in this class of application is that the Ontology will allow the IR system a better representation ("understanding") of the concepts being searched and thus make possible an improvement of its performance from what is presently the case [56].

The problems of IR are well known to the research and user communities. Amongst the most widely recognised ones are the so-called missed positives and false positives [56]. In the first case the system fails to retrieve relevant answers to the query whereas in the second case the system retrieves answers that are irrelevant to the query. However, the benefits of using ontologies for information retrieval outweigh the potential problems and include:

Query augmentation: the use of the Ontology for the expansion of a user query so as to better understand the context, e.g., taking into account the search mode employed in order to return more relevant results.

Content harmonisation: that is sought when internal (proprietary) and external (non-proprietary) information sources differ. Generally Ontology alignment or merging process are used whereby multiple proprietary internal information sources are mapped to a single external information source.

Content Aggregation/presentation: the presentation of content to the user. It covers both the collection and integration of content from various sources, increasingly made possible by the Web, and the creation of intuitive user interfaces. The Ontology can enable the results to be filtered, ranked and presented according the data semantics. Contradictions and the inter-linking of related information, e.g., a different possible

answer to the same query, or an answer to a different but related query, can be handled using the Ontology.

Content Management: the categorisation, (re)structuring and indexing of information in the form of documents and other source data can be enhanced using the Ontology. This makes in addition the domain conceptualisations assumptions explicit, which in turn makes it easier to change domain assumptions and to understand and update legacy data.

Domain knowledge / operational knowledge separation: an Ontology enables the operation, in terms of the application specific business rules, used to formulate the queries, to be represented independently of the stored information. The advantage of this separation is that we can more easily reuse the domain knowledge with different sets of application specific operational knowledge. For example, a Core Ontology for the Inland Water (IW) domain can be reused in conjunction with different commitments from applications, and from different users, such as the European Water Framework Directive policy-maker and the European citizen at large.

2.7.4 Integrating Rule-based and Semantic Logic Systems

Traditionally many IR systems are passive, queries, data updates and transactions are only executed on request. Many applications require IR systems to be active, e.g., to monitor and take actions when the underlying data changes. There are several ways to express rules such as the ECA or Event, Condition Action paradigm when an event is received, it is evaluated and if it passes a guard condition an associated action is triggered. Another common way is to express a rule is as a production rule using a logical implication. When the conditions in an antecedent clause A are evaluated to be true, then the consequent clause B is implied to be true. This is equivalent to a rule "if A Then B".

Rules could be embedded as part of the stored data, so called stored procedures, or contained in special applications or middleware that interacts with the data, the latter design leads to more reusable rules and has the advantage that applications can define and use their own specific rule-sets. There are several processes associated with rules such as detecting events and evaluating the guard conditions and executing the actions, how rules trigger other rules and resolving conflicting rules when several are active. Hence, generally rule systems are specified differently compared to the more passive relational or semantic stored data models.

It is important to note an important effect of two different types of semantics on facts and the rules for deriving new facts: Open World Assumption or OWA versus Closed World Assumption or CWA. The closed world assumption is often implicit in database models where every record not explicitly present in a table is implicitly assumed to represent a fact that is false rather than unknown. OWA is implicit in the Semantic Web that statements or resources not presenting in RDF based is assumed to represent a fact that is unknown rather than false.

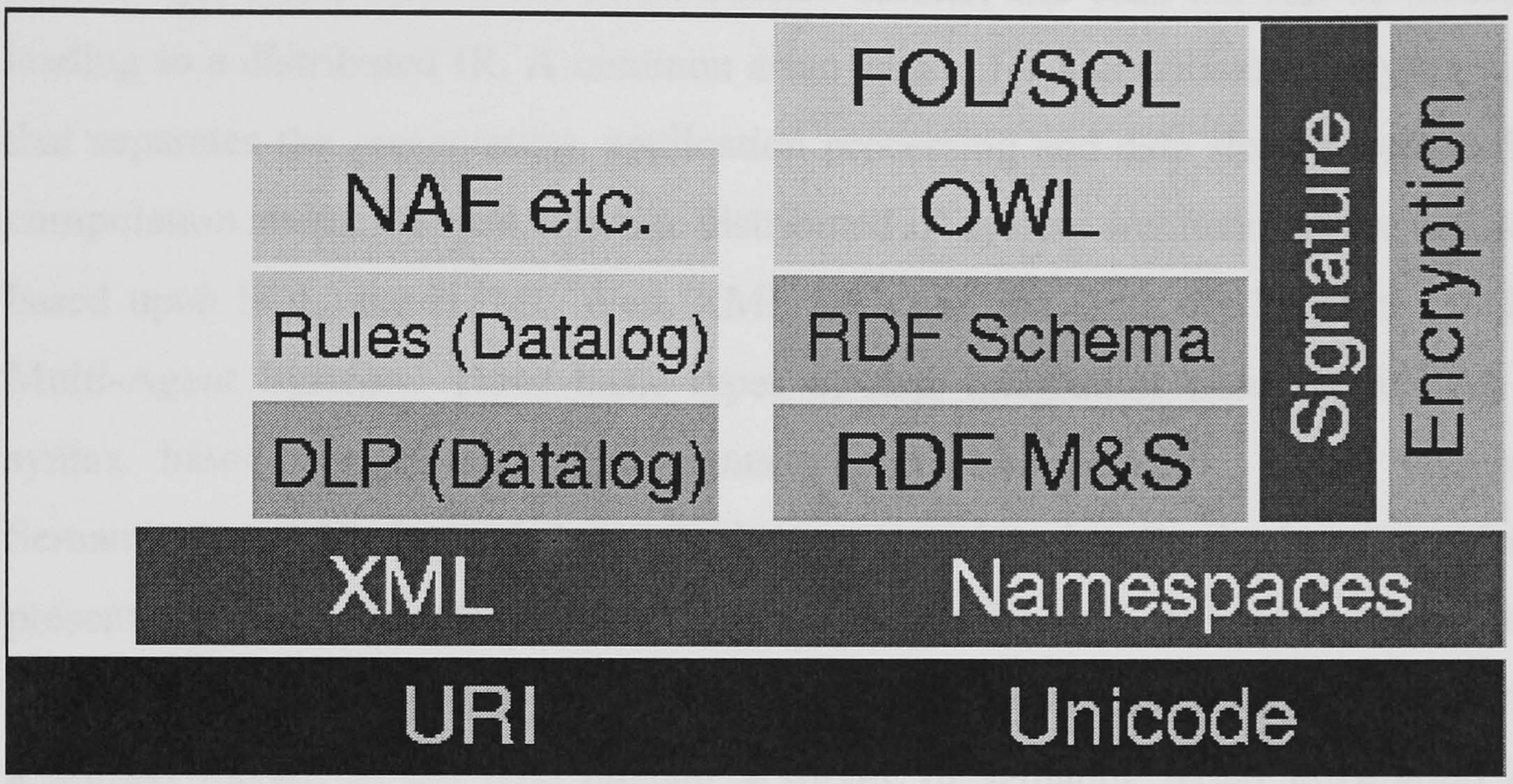

Figure 3 Semantic Web with Datalog rules, taken from [43]

Figure 2 shows a Semantic Web vision for combining syntax, semantics in the form of ontologies, logic and rules. It is assumed that these functions are defined in a hierarchy of languages with each one in one layer dependent on the one below. However there are many different types of rule systems and it is not clear what expressivity is needed for the rule system and its relation to the Ontology layer and the expressivity in the Ontology model needed to support rules. As a result several alternative layered models are available. In [43] three alternative layered models of the Semantic Web to Figure 2 are presented to deal with issue of how to combine rules, semantics and logic in a single model. For example, rules and OWL can be considered as being elements sitting side by side in the same layer in one version. In another version of a layered Semantic Web architecture taken from [43], the base layer split into two stacks or towers at higher layers rather than being a single layer in order than one of the towers the Datalog can deal with the CWA and CWA rules whereas the other tower deals with OWA rules, see Figure 3. There is as yet no clear winner or optimal framework to combine rules, logic semantics and closed world assumption data models.

2.8 Summary

High-level system architecture models for distributed Information Retrieval (IR) systems consist of three basic tiers of functions: data resource management, application logic and presentation. Modelling a system in this way gives systems the flexibility to add new data resources without requiring changes to the application processing or to the presentation, providing the interfaces with the data resources do not change. These tiers could be partitioned further, and each tier can be distributed, leading to a distributed IR. A common arrangement for a distributed IR system is one that separates the presentation, application processing and data storage onto different computation nodes. Several concrete distributed IR system architectures are considered based upon SQL, the HTML Web, XML Web and the Grid, the Semantic Web and Multi-Agent Systems. Three basic types of data integration were considered: SQL syntax based integration, XML syntax based and semantic based integration. Semantic-based IR systems have the best potential to handle the heterogeneities at present in some distributed IR systems. However, such a system faces design challenges when integrating different kinds of behaviour such as rules and semantics in a unified model. In the next chapter a survey of semantic based integration of IR systems is given.

Chapter 3 Literature Survey

3.1 Introduction

Today, often referred to as the age of the information technology society, access to available information that is often heterogeneous and distributed, is required [94]. Information sources, services, applications and users within a domain also require some form of interoperability between these in order to share and combine information across these. This can be greatly facilitated by the sharing of a domain conceptualisation amongst different information entities such as applications, user groups, and data sources. Interoperation between different entities is challenged by the existence of heterogeneous representations and interpretations of the domain knowledge that can result in interoperability problems within a domain.

Much research recently has focussed on the use of Ontology driven or semantic approaches to support interoperability by providing a formalised representation of conceptual structures in an explicit manner. Ontologies have been used in a wide range of information systems and these are surveyed in this chapter. The role of an Ontology in IR systems varies. It may be used to support the wrapping of and be used mediate and translate between, related information entities. The aim of this chapter is to survey and classify the use of ontologies in some key areas of information retrieval and in particular for relational database type information sources.

3.1.1 Motivation

In traditional IR systems, the accessibility and usability of information is often limited because of: insufficient expressivity of the data model to adequately reflect the complexity of the real word; because of the information heterogeneities, lack of data integrity and data redundancy that arise when data is distributed and because of the poor productivity in developing and managing data application. As a result, in the 1970s Codd [29] proposed the relational model as the basis for a new data model that organises data into tables, linked via key relations to form a flat or single layer data space. Subsequently, a data retrieval interface, SQL or Structured Query Language, to relational type databases has been standardised by ANSI, the American National Standards Institute in 1986 that has been subsequently extended several times. This still remains as the dominant data storage model in the 2000s. Its key strengths are its

ability to maintain data quality via data integrity constraints and concurrency control for a data model that may be distributed and that has been designed to adhere to a single data schema. However, the relational data model lacks the expressivity to model complex rich data structures and hierarchies that are found in the physical world, to describe how physical world relational data structures map to flat relational data structures and lacks support to relate different but yet related data schema models and associated data instances.

Much recent research, for example has investigated if Semantic Web or Ontology data type models can provide a complementary data model that can interlink with the relational data model to overcome the limitations of the relational data model mentioned above. Ontology data models are recognised as an important means to express semantic knowledge using an explicit representation of the domain conceptualisation. The reason for ontologies being considered so useful is largely their potential to support a shared and common understanding of some domain that can be communicated across people and computers. Ontologies can be used for data and metadata representation, metadata directories, information interoperability and information integration. Ontologies can help to resolve the potential information heterogeneity and information interoperability problems found in the application domain.

Ontology alignment and Ontology merging, or integration, are the two major approaches to solve interoperability problems for distributed and heterogeneous data within an application domain. Ontology mappings can provide a common layer to interlink several related ontologies for the exchange of information in a semantically sound manner[51]. Ontology mappings can be set up at different levels of abstraction including vocabulary, syntax and semantics, depending on the nature of the interoperability problem to be solved.

This chapter analyses the use of Ontology-driven approaches for integrated and interoperable information retrieval (IR) from multiple heterogeneous data sources. IR systems are clustered into two types of architecture alignment systems and integration systems. The use of semantic mappings is a crucial component in dealing with Ontology alignment and Ontology integration. It supports the transformation of knowledge representations amongst entities in a large information society.

3.1.2 Information Heterogeneities

Sheth has classified information heterogeneities into types, mainly focusing on the technical differences with respect to system, syntactic and structural and semantics heterogeneity [86]. S*ystem heterogeneity* refers to the utilisation of different software and hardware platforms including deployment of different DBMS and operation systems, different file systems and access operations, command interfaces, transaction control and recovery capabilities. S*yntactic and structural heterogeneity* refer to the different terminologies, data models, logical structures and corresponding operations used. S*emantic heterogeneity* indicates the meaningful representation of knowledge and its interpretation by different information entities.

The Knowledge Web project classifies information heterogeneities according to another type of classification at the level of syntactic, terminology, semantic and pragmatic heterogeneity[22]. *Syntactic heterogeneity* encounters all forms of heterogeneity that depend on the choice of the representation format. *Terminology heterogeneity* encounters all forms of mismatches that are related to the process of naming the entities (e.g. individuals, classes, properties, relations) that occur in the domain Ontology. *Semantic heterogeneity* encounters mismatches to do with the content of Ontology. *Semiotic or pragmatic heterogeneity* encounters the discrepancies that have to do with the fact that different individuals and communities may interpret the same Ontology in different ways in different contexts.

Regarding support for universal retrieval to legacy databases in EDEN-IW, therefore we have further developed a heterogeneity classification to cover the specific types of information reflecting the heterogeneities in the inland-water domain.

- *System heterogeneity* query interfaces varies among different RDBMS such as SQL Server, Oracle 9, Oracle RDB and Microsoft Access. The system needs to provide transparent query access to all these types of data repositories.
- *Syntactic heterogeneity*: different language representations and logical structures for information storage, retrieval and exchange are used. For example, query expression varies from language structure, query syntax and corresponding constraint relations., e.g., SQL-1 tables vs. SQL-3 user-defined data structures, RDF query vs. SQL query.

- *Conceptual heterogeneity*: deals with the mismatched classifications, modelling, and structuring of the domain knowledge. The conceptual heterogeneity can be divided into sub-types:
    - *Structural heterogeneity* indicates all different property relations in a conceptual domain, especially for is-part and is-a relations. The understanding variation of knowledge domain can lead to disparate hierarchy structures in conceptual representations.
    - *Classification heterogeneity* indicates different categorisation relations between data instance and relevant classes according to different intended usage.
    - *Modelling Heterogeneity* refers to the nature of general features for conceptual modelling, e.g. object-oriented model vs. relational model.
- *Terminology heterogeneity,* covers all the naming differences according to linguistic representation such as synonym and homonym that indicates the choice of entity naming according to natural language conventions: the same named concept may have different meanings and be related differently to other concepts (homonyms) and different named concepts may have the same meaning (synonym). Terminology heterogeneity also concerns other linguistic problem such as different abbreviations, spelling and multi-lingual support.
- *Convention heterogeneity,* envisages the knowledge presentation variation in respect to different referential knowledge, assessment systems and coding conventions. For example, values for chemical concentration may be represented in different units that varies according to the spatial locations of the monitoring stations and expressed in terms of different coordinates systems.
- *Semiotic heterogeneity*: focuses on the meaningful interpretation of domain conceptualisation regarding the understanding of semantic expression in contexts of different individuals or communities. Semiotic heterogeneity mainly reflects the process of human understanding of knowledge conceptualisation of a certain information domain. The variation of representation mostly relies on a developer's intended usage of information. The semiotic heterogeneity can be further subdivided to support user view customisation along the dimensions of coverage, granularity and perspective [22]. Coverage identifies user interest in subset of knowledge conceptualisation.

Granularity describes the general level of terms for users to represent their understanding about domain knowledge. Perspective is a unique viewpoint about how a user evaluates the domain knowledge. The viewpoint may be derived from conceptual representation of knowledge domain reflecting the intended goal and utility functions for particular user groups or application. For example, environmental concerns of inland water information can be expressed as general interest in the water quality grade or as specific determinand observations (granularity), chemical or nutrient quality assessment (coverage). Water quality can be assessed using general criteria or in relation to its chemistry (perspective).

3.1.3 Database Schema Models

3.1.3.1 Multi-lateral Database Schema Models

SQL views or virtual tables are an established way of projecting a more abstract or application-oriented view of a table or combination of tables in relational databases. They can provide data customisation and can adapt content to meet the demands of specific applications and users [27]. A view can be seen as an arbitrary query stored upon database schema in order to provide customised information retrieval to satisfy different user demands. The ANSI/X3/SPARC Study Group on Database Systems has outlined a three-level data description architecture [91]. The schemas in its three layers are the conceptual schema, internal schema and external schema. A conceptual schema describes the logical structures and relations amongst these structures for a database system. An internal schema describes the physical storage and access characteristics for those logic structures in conceptual schema. An external schema supports a customised viewpoint to access a subset of conceptual schema. The aim of such layered model is to maintain independency of data representation with respect to different applications and users, so that a change in on layer will not necessarily require a change in other layers if the interface between the layers behaves the same. This means, for example, that a new database can be added without necessarily requiring the logical schema or external schema to change.

SQL views have been thoroughly studied in the context of database integration, query optimisation and other relevant areas[27] [78] [57]. The formal semantics of database

views in database integration systems is described in [57] according to a context of relational database integration. Database integration is defined as the problem of combining data residing in different sources and providing the user with a unified view of these data. The semantic of a database integration system *I* is represented as a triple <*G,S,M*>, where G is the global schema, S is set of source schema and M is set of assertion mapping query over G to corresponding query over S. The well-established approaches, such as local-as-view (LAV)[78] and global-as-view (GAV) [28] [97] have been developed to support query reformulation in the context of database integration. Data mapping makes a distinction between LAV and GAV, where constructs of local schema are represented as views over global schema in LAV and constructs of global schema are represented as views over local schema in GAV.

Database schemas provide a description of the stored data structures in the form of tables and the query interface to access them; relations between data structures are defined using key relations and restricted using entity and referential integrity rules and other data integrity rules. Views over relational databases focus much on information retrieval in a closed information system, where centralised management is carried out throughout whole knowledge domain. Development of such an integrated system is oriented to expert users who are supposed have a sufficient knowledge understanding of the knowledge domain and its logical structures. It is important for the database view to maintain the query consistency amongst different views to guarantee the processing correctness of expression transformation and result answering for execution of a global query in local data sources.

3.1.3.2 Limitations of Database Schema based Integration

Relational database schema integration faces the following potential challenges:

1. Semantics of relations and attributes are not formally defined
2. Query reformulation focuses on transformation of syntactic representation, whereas the semantic meaning and possible interpretation of data instances is not covered within its knowledge domain.
3. Expressivity is limited because of the rigid structure of RDBMs that relates to their contained tuple set relations.
4. Lack of support for information operations upon hierarchy structures that are naturally found in the physical world data models

5. Database model is oriented to a closed application domain that is often under centralised management.
6. Related underlying knowledge such as context and constraints are not captured in relational data model.
7. The management of the information model is normally de-centralised: update of the intensional and extensional information is designed to occur concurrently and to maintain data integrity.
8. There is no widely supported standard to support standardised querying and reasoning about relational models.
9. Database schemas and catalogues often do not provide explicit semantics for their data. Either the semantics has never been specified, or the semantics were specified explicitly at database-design time, but the specification has not become part of database specification and is not available anymore [73].

3.1.4 Overview of Survey

This chapter surveys the research for Ontology based information sharing and retrieval along the following main themes:

1. Architectures for information retrieval system
2. Types of Ontology mapping to finding semantic correspondences across related information models.
3. Types of information adaptation to support multiple viewpoints of information.
4. Methods to combine heterogeneous data schema such as logical data schema and external user defined schema.

The remainder of this chapter is organised as follows: it start with a general classification of data interoperability and data integration systems. Ontology alignment and Ontology integration are regarded to be main interoperability solutions.

Then the conceptual roles of semantic mapping and during the process of Ontology integration and alignment are examined. This is followed by the part of the survey that examines information tailoring to support multiple user viewpoints. Then solutions that combine semantic user data models with logical data models are examined. Finally, a summary is given.

3.2 Semantic Integration of Database Resources

The integration of multiple data models during the modelling process of the conceptual world can be roughly classified into two types, merging or integration and alignment. Noy and Musen [68] defined view *merging* as the creation of *a single coherent Ontology that includes the information from all the sources* and *alignment* as *a process in which the sources must be made consistent and coherent with one another but kept separately*. This may entail maintaining local Ontology wrappers for each data source leading to a multi-lateral Ontology model. The merging approach often leads to the creation of a global knowledge model where individual local Ontologies can be mapped to each other. The alignment approach avoids the process of creating a global knowledge model, instead it maps specific semantic content between the local ontologies, directly.

3.2.1 Architectures for Semantic based Data Integration System

An Ontology model can act as a metadata model in a distributed IR domain. Metadata is usually defined as data about data. Metadata often involves more than simply being information about data. Metadata needs to be stored and managed. It can reveal partial semantics such as the intended use of data [86]. Metadata can be represented in various formats and expressivity, from database schema to semantic model. In addition, to classifying data integration approaches into either alignment or merging to map between two or more disparate data models, Ontology or semantic based integration approaches can also be classified as to whether they use a single semantic model, multiple semantic models or hybrid semantic models.

3.2.1.1 Single Ontology system

A single Ontology system is characterised as the sharing of a single harmonised vocabulary set at the global level that is mapped to local data sources for information retrieval. Query access to local data sources must be formed in the global vocabulary and using its syntax structure. The Ontology works like a common dictionary base [30, 31] to identify resource locations [13, 31, 38] and terminology mappings [31, 54]. A similar conceptual structure is enforced in all local data sources so that the harmonisation of mapping relations between global Ontology and local data resources can be conducted in a straightforward way, i.e. no structural mediation is defined. Extensions to IR systems can be quick and cheap as any plug-in of new data sources

with a similar conceptual model is relatively easy. However, the system flexibility is restricted when local resource may have a different conceptual structure that is not covered by the model at the global level. Similarly, the conceptualisation changes of local data sources may result in re-development of the whole global model and all relevant mappings to data sources. It is inherently easier to develop and integrate data sources with a similar conceptualisations within a single knowledge domain. When the management of such integrated IR systems are conducted at the global level, some control and autonomy by the local data- source owner is lost.

3.2.1.2 Multiple Ontology System

A multiple Ontology system consists of multiple ontologies representing separate conceptualisations of each data source. The conceptualisations for each local data sources may be too disparate to be integrated into a common global Ontology. An ad-hoc mapping is established between each peer's local ontologies and another's [64]. The information retrieval is performed in terms of peer-to-peer knowledge translation between different ontologies. No global or harmonised conceptualisation is available in multiple Ontology systems. Remote information access is undertaken by the mediation or mapping service which could be defined or generated dynamically in the resource wrapper in order to achieve peer to peer translation. The advantage of a multiple Ontology solution is to keep the local logic view at a maximum level without any common or minimum commitment or vocabulary set to a global view. The ad-hoc mapping relations allow for flexible knowledge transformation between different conceptualisation. A flexible process for the addition of new data source can be developed. The control autonomy is left to the local data-source owners. However without the common logic view, the maintenance of the local logic translation can be difficult because more mapping relations have to be maintained to cope with interoperability between many peers.

3.2.1.3 Hybrid Ontology

A hybrid architecture [31]comprises both single and multiple Ontology systems. The conceptualisation of data sources is expressed in local ontologies but a common conceptualisation can also be developed at a global level independently of the local conceptualisations. Semantic mapping are deployed to mediate between the global and local models. Only a partial mapping is required between local model and global

model and the local data-source owner can choose the part of information for exportation. The global model is an independent representation of common conceptualisation in the knowledge domain such that the global model can be shared and reused in different applications. The common query syntax and semantics can be defined to give a global interpretation of user queries throughout the system. The representation transformation can be set up at different levels depending on the reasoning processes used with the global to local semantic mappings. The addition of new data source connections is achieved via development of a new local Ontology and its mapping to the global Ontology. The management of hybrid system is conducted at two levels: at the local data-source layer where the data-source owner can change the local conceptualisation and data content; at the global level where a system administration ensures the correctness of global conceptualisation. The content change at global level may involve an update of semantic mappings throughout the breadth of the system data model.

3.2.2 Ontology Mappings for Data Integration

Ontology mappings are needed to overcome the interoperability issues through the information transformation across different Ontology models. The mapping from one metadata set to another and from metadata to a real data set. The current approach for Ontology mapping covers a number of computer science disciplines ranging from machine learning, concept lattices and formal theories to heuristics, database schema and linguistics [51].

Ontology mapping plays a crucial technical role during the integration of distributed IR applications. Ontology mapping could provide a mediation layer from which multiple ontologies could be accessed and hence could exchange information in a semantically sound manner, i.e. Ontology mappings map a term T1 of Ontology O1 to another term T2 of Ontology O2, such that the axiom if T1=T2 for any axiom in O1 with T1, its substitution axiom with T2 also holds. The mapping relation gives a morphism for a terminology interpretation over a specified knowledge domain.

Vocabulary and semantic expressions are mapped across different conceptualisations to resolve representation transformations with different focuses. The survey has grouped relevant projects into three classes regarding their usage of Ontology mapping to solve data integration problems:

- Syntactic mappings to support schematic integration of relational databases.

- Vocabulary mappings to support terminology integration.
- Semantic mappings to support the integration of different meanings.

Each of these is discussed in turn.

3.2.2.1 Syntactic Mapping: Schematic Integration of Relational Databases

Conventional integration for relational databases e.g. multi-databases and federated database establishes a syntactic based approach for the integration of database schemas by introducing mapping relations between schematic constructs. This approach focuses on determining the corresponding relation and schematic structure via relational operations [98] [30, 31], in order to reformulate global access of the integrated schema to the distributed local data sources. There are two main approaches: federated schema and global schema [85]. In the federated approach, each database supplies an export schema, a proportion of its schema that is willing to share, for others to import. Whilst in the global approach, each local database's schema is combined into a single integrated schema. There are questions about the scalability of schema-based approaches, including data warehouses, because of the number of possible heterogeneous schema possible and the difficulty in normalising numerous syntactical mappings between heterogeneous database schemas. The E-R or Entity Relationship diagram was used to as the concept representation for relational and object-oriented data models. This model is not online, machine-readable and processable by applications. Based upon the conceptual model of the E-R diagram, the Data dictionary was used mainly for the integration of structured data resources; however it is simple and non-standardised. The meta-data is used at the schematic level. Syntactic mappings between schemas mainly target resolving SQL syntax issues, i.e. to generate appropriate SQL expression for target data sources, and on schema derivation using relational operators. However, this approach is limited by the existence of similar conceptual structure for synonym relations. The reuse of an IR system becomes difficult due to the tight coupling between the metadata for the data schema and the application queries and transaction processing that is designed to use a particular data schema.

More flexible solutions have been proposed for a generic database access system, for example using SQL2, CORBA or Common Object Request Broker Architecture and

VTI or Virtual Table Interface [66]. SQL2 supports user defined data type and function. A complete UDF or User Defined Function facility will allow data-intensive functions to execute in the same address space as the query processor, so that the enterprise database methods may achieve the same performance levels as built-in aggregation functions. VTI allows the user to extend the "back end" of the ORDBMS, to define tables with storage managed by user code. The query processor and other parts of the ORDBMS "front end" are unaware of the virtual table's special status.

The syntactic approach creates schematic mappings between database schema based on relational operations, such that one schema element can be derived from the other element formally. To achieve that, knowledge about the database structure and domain knowledge is needed. Changes and updates to the system architecture and schema content needs to involve contributions from both database administrators and domain experts.

3.2.2.2 Vocabulary Mapping for Terminology Integration

Vocabulary approaches are heuristic rather than being a formal method applied in syntactic system. It focuses on solving terminology heterogeneity amongst application systems. Terminology heterogeneity is due to the design and development autonomy of the local database source and different contexts being used. A common problem is the use of synonym where the same term stands for the different concepts and homonyms where different terms represent the same concept. Similarities measured between terminologies use different criteria w.r.t. machine learning [58, 60], concept lattices[50], linguistic structure [54, 83], instance classification and instance representation[67] .

A vocabulary based mapping system can be applied to a wider scope including RDBMS, structured file, plain text storage, and multimedia resources. The standard metadata mark-up language for example XML and RDF links the metadata model with the heterogeneous data resource. The uniform access and integration of heterogeneous data resources has been achieved. Metadata defined at the terminology level can be structured in terms of a data dictionary and keyword-based Ontology.

The vocabulary system provides a common solution to derive a semantic matching using the Ontology content or external linguistic thesaurus without the aid of domain background and underlying knowledge. The mapping relation can be established automatically at the level of a shared vocabulary.

3.2.2.3 Semantic Mappings

There are questions about the scalability of syntactic approaches because of the number of possible heterogeneous schema and the difficulty in normalising numerous syntactical mappings between heterogeneous database schemas. As a result interoperability based upon models of the semantics of the underlying databases has been proposed [52]. Thus the problem of resolving differences in conceptual structure is reduced to the problem in understanding the differences between different semantic models corresponding to the different databases.

Heavy-weight Ontology-based knowledge representation languages, so called because they support an expressive conceptualisation with an associated logical model, such as CLASSIC, LOOM, DAML+OIL, OWL can be used to build Ontology models to express real-world conceptualisations according to semantic relations. Such languages include some common features such as an embedded logic framework and frame-based or class-based hierarchical structures. Inference can be deployed in an expressive logic-based framework to enhance data access and data integration. Semantic mappings are expressed in terms of subsumption relations between conceptual terminologies and instance sets in the same knowledge domain.

Information processing applications can use knowledge inferences and rule-based reasoning techniques to generate new information derived using the metadata. The forms of concepts and their relations used in Ontology representation languages are much more expressive and complex in comparison to syntactic approaches. In addition, logic processing is available to provide knowledge processing and intelligent services to underpin decision making, strategy analysis, problem solving, relaxation of information query constraints and customised user queries.

3.2.3 Systems, Projects and Applications

3.2.3.1 Information Retrieval systems

Carnot

The Carnot project [30] extends a conventional composite database integration approach, by enhancing it with a global semantic knowledge layer to accommodate syntactic heterogeneity in database schemas. A concept dictionary in the global

schema gives the vocabulary mappings from a user query in the form of topic hierarchy tree to the global Ontology and to the local database schema. The Ontology is expressed in CYC and Carnot's own knowledge representation tools called KRBL or Knowledge Representation Based Language. The mapping relations between information resources and the global schema are represented in the terms of a set of articulation axioms: statements of equivalence between the components of the two theories. The schematic mapping between local and global view is constructed at the synonym level.

Carnot focuses on the schema integration of heterogeneous databases in the same knowledge domain, where an exact semantic equivalence is maintained in order to build the synonym mapping axioms between global and local schemas. The axioms define a set of substitute rules for global terms and values in the local schema. The expansion of additional heterogeneous data sources in Carnot can lead to the modification of global semantic models, making the mapping relations difficult to maintain. Queries may not be able to be mapped to all local data sources because direct synonym relations may not exist.

Rule-based articulation axioms define the semantic mappings between two view expressions. The semantic equivalence is described as two entities with an equivalent meaning under given semantic relations and constraints. The processing for query translation involves replacing the semantically equivalent entities in the source sentence with entity expression in the appropriate view. The translation is conducted syntactically here.

Carnot supports the development of applications that can be tightly integrated with closed information systems. Carnot doesn't solve the problem of value mapping and the scope of the relevant global schema. The system is closed in the sense that it lacks the use of standard semantic representations and knowledge exchange protocols.

Dome

DOME (Domain Ontology Management Environment) [31] is an Ontology-based corporate information system for the integration of heterogeneous databases within an open Business-to-Business eCommerce (B2B) environment. Independent data sources that share a similar data model are supported. A shared Ontology represents the vocabulary commitment across the knowledge domain that can be mapped to application and resource ontologies. A resource Ontology is the description of the data

model and terminology of a local data source; it can be automatically extracted from a database source using the specific tools.

Ontologies in DOME are implemented using CLASSIC [24] – a type of Description Logic and the Open Knowledge Base Connectivity (OKBC) Ontology service model [5]. A common Ontology representation is derived and can be mapped to different database schema to support query transparency. The content-based data source directory is maintained in XML/DTD format. The mapping between global and local ontologies is defined in the terms of a rule-based declarative syntax. The mapping rules are created manually providing the mapping relations between the common Ontology and the local data sources. A resource dictionary facility is used to record the location of information sources within application domain. A resource wrapper is designed for each database type. The terminology matching between global and local view ontolgies is solved using exact concept or attribute mapping – these are derived manually. A rule-based inference application is deployed to perform the query translation between the shared and resource ontologies. The terminology mappings are described in terms of synonym relations. DOME is developed for a static application domain where distributed data sources have similar structure. The domain knowledge is partitioned into application, shared and resource ontologies and supports different presentation view to user groups. The introduction of new data source or application service may involve the modification of global Ontology and mapping relations. No value mapping process is explicitly specified.

The system contains independent data sources with similar data models. To solve vocabulary mismatches, i.e. the same terms having different meaning or the different terms having same meaning in local source domain, exact mappings between ontologies on the level of concept and attribute are maintained. A rule-based inference application is deployed to perform query translations between shared and resource ontologies. A top-down approach is used to build the shared Ontology. The resource Ontology is built using a bottom-up approach.

InfoSlueth

InfoSleuth [38],[13] is comprised of a network of cooperating agents that uses an agent-based communication protocol, KQML (Knowledge Query Meta Language) and KIF (Knowledge Interchange Format) [40] content language to gather data queries and to process them. The agents also use the OKBC service [5] model to manage and

maintain a common Ontology model that interlinks the different data resources. A service broker (software agent) employs an internal Ontology representation of deductive database language LDL++ [99] to reason about information content and hence to identify a relevant data repository. InfoSleuth deals with the information transformations between user queries and local database access. Mappings between the common Ontology and the local database schema are developed manually.

Information integration operations in InfoSleuth use a set of software agents and asemantic Ontology model. Each agent performs a designated role:

- User Agent: interacts with the user interface to provide an intelligent information gateway for agent system. It retrieves the system's common domain ontologies to assist the user in formulating queries and in displaying their results.
- Ontology Agent: provides general access to ontologies and answers queries about ontologies.
- Broker Agent: a match-making agent that receives and stores advertisements from all InfoSleuth agents about their respective capabilities. It accepts and answers queries from other agents. It can direct queries to specific data sources according to the agent directory information.
- Resource Agent: wraps information sources and provides a uniform query interface to agent systems. It handles the semantic mappings between local data schema and the common Ontology representations.
- Data Analysis Agent: corresponds to resource agents specialised for data analysis and data mining.
- Task Execution Agent: coordinates the execution of high-level information-gathering subtasks (scenarios) that are necessary to fulfil queries.
- Monitor Agent: tracks the agent interactions and the task execution steps. It also provides a visual interface to display an agent's execution

The resource agent is now discussed in more detail. The semantic mapping in a resource agent comprises both schematic mappings and value domain mappings. The schematic mapping deals with the synonymy mapping among the database schemas. The value domain mapping is the value instance mapping of object representations between local database and the common Ontology. The resource agent uses a syntactical approach to map concepts in one domain to another. A value mapping

agent does reasoning of the mapping with reference to the ISO/IEC1117 meta-data registry standard.

Information sources include data repositories such as relational database, object databases and plain text storage. One or more common ontologies are modelled as the knowledge reference to support communication for a multi-agent system design. The common Ontology is modelled in OKBC. A Java based backend application is embedded in resource agent to provide a local data access interface akin to but at a higher level of abstraction to JDBC to local database repositories. Each local data source contains the distinct parts of the domain knowledge and no local data source overlaps. The query translation occurs in the resource agent, focuses at the schematic level and deals with the synonym mapping. Value mappings in a domain may involve more complex processing and use rule-based reasoning.

Observer

Observer [64] is a query processing application designed for global information systems that comprise several types of data sources for example web page, pre-existing Ontology, files and relational databases. The local data repository is wrapped by a query processing component that is responsible for external query processing and translation for local data access. An Ontology server residing in each local query processing component provides information about how to access ontologies and any data repositories. Solving information heterogeneity is limited at terms and data structure level via logic-based inference.

Distributed multiple Ontology models are defined for data sources to handle the information heterogeneity and translate queries for local data repository access. The Ontology is described in CLASSIC using a description logic (DL) notation. The access to the local data repositories is conducted as the intermediate mapping between DL expressions and queries to the local data repository. A separate mapping relation repository is defined to capture the concept and role alignment relationship between ontologies. The OBSERVER system assumes the number of relationship between terms across ontologies is less than the number of terms relevant to system, hence the mapping is formed in alignment style, i.e. no global conceptual model is developed. The mapping relations are classified into synonym, hyponym, hypernym, overlap, disjoint and covering. The query processing module browses the mapping relations for the target Ontology for terms to substitute. For the case when no synonym can be

found, relevant terms will be considered instead, thus information loss will happen. Observer is capable of estimating the intensional information loss in the terms of vocabulary subsumption relationships and external loss in terms of recall and precision. However this may be imprecise, as the measurement of metadata terms relation may vary in comparison to that of the real data repository depending on the particular collection of data. The semantic integration is conducted upon the premise of shared vocabulary sets and hierarchy relations that may not be satisfied in the environment containing several independent vocabulary sets. In order to introduce the new data source, modifications to the mapping repository may be required. The effort can be extensive if the introduction of new vocabulary set is quite disparate.

A multiple Ontology model was used in Observer. The key objective of the multiple Ontology approach was to solve the problem of homonym and synonym relationships between terms across ontologies. Mapping between one user Ontology and more components ontologies, the mappings were maintained based on the synonym relations.

TSIMMIS

TSIMMIS [28] implemented a Global-As-View approach to data integration, in which a lightweight object model called OEM (object exchange model) is applied to integrate heterogeneous data sources. OEM is an object-oriented, declarative-syntax model that is independent from the data source model and schema. The simple and general data model represents has-a relation with semantic ID naming and set object values with object references and object type information.

A mediator is generated automatically using a predefined template and rule descriptions for the result fusion of query evaluation upon a data source. MSL or Mediation Specified Language is used. MSL is an object-oriented, logic query language targeted at OEM data models and functions and heterogeneous information integration. Wrappers are written in WSL, an extension to MSL that supports additional query capabilities and content descriptions for data sources.

The mapping rules in MSL specify the OEM (global) object and relations as a view or as data source relations using Global-as-view loose-coupled relation mappings. OEM is flexible enough to cover various data structures and models. No explicit global data schema is specified. A constraints manager specifies the rules to ensure the semantic consistency over the stored information.

The project focuses on the integration of various types of semi-structured or non-structured data sources, such as plain text, excel file and command-based query system. The data query and information retrieval for such systems are not well-structured. A lightweight object-oriented model is used for the global conceptual representation. Embedded simple semantic relations make the system flexible enough to cover more diverse data sources. Syntactic and modelling heterogeneity is resolved by using GAV view unfolding and rule mapping between the global query and local access interface. The semantic reasoning and inference is not a focus, although the system mentions object ID paths that may contain corresponding semantic meanings for an object value in a corresponding context.

Knowledge shifter

The knowledge shifter [54] is an agent-based system that supports access to heterogeneous web information sources for the specific knowledge domain. The knowledge models are partitioned into three layers consisting of a user layer, knowledge management layer and data source layer. A collection of cooperating agents reside at the various layers and performs specified function. User can specify queries via a given interface. The user query is refined by an Ontology agent in two phases: structural extension with defined conceptual models for the knowledge domain and synonym and hyponym terms extensions through querying vocabularies such as WordNet and the USGS Geographic Names Information System. A refined user query is decomposed and sent to a data source server using a corresponding interactive protocol. Results are combined and ranked.

BYU-Global-Local-as-View

Xu and Embley [97] proposes a hybrid database integration approach, BYU-Global-Local-as-View, to integrate RDBMS. The aim is to solve the vocabulary, structure and schema heterogeneities among different database schemas via a virtual view mapping approach. The approach combines the advances of both GAV and LAV by provision of scalable source evolution for LAV and reduces the query reformulation complexity for GAV.

A conceptual global schema is created independently from all source schemes. A semi-automatic approach can create a virtual source schema thus the schema elements

with semantics corresponding to source and target schemas can be mapped. The mapping process can be derived semi-automatically from source schema through predefined data operation (data algebra) in the design phase such as selection, projection, join, union, decomposition, composition, Boolean, de-Boolean, rename and skolemisation. More algebra operators are defined to extend the standard operators. The global schema element can be mapped to the source schema as a view with inclusion dependency.

The query is reformulated using mapping rules that substitute the corresponding schematic views with derived rules (GAV). Evaluation of a global query can be decomposed into many sub-queries with global elements that are substituted by semantic correspondences in the source schema with inclusion dependencies.

3.2.3.2 Ontology Mapping Systems

ONION

ONION (Ontology compositION) [67] is an information interoperation system providing ad hoc Ontology transformation, based on semantic alignment. The system supports a precise composition of information from multiple diverse sources by not relying on simple lexical matches, but on human-validated articulation rules among such sources. An articulation generator semi-automatically derives semantic matches among concepts in a pair of ontologies when strict-typed relationships with pre-defined semantic exist.

The Ontology mapping process includes non-iterative and iterative algorithms. Non-iterative matching is generated based on similarity measurements of relevant concepts. Iterative algorithms require multiple iterations over source ontologies in order to generate semantic matches between them.

Ontologies are modelled in a graph structure. These algorithms look for structural isomorphism within sub-graphs of a shared lexical hierarchy, or use the available Ontology rules and any seed rules provided by an expert to generate matches between the ontologies. Iterative algorithms are typically used after non-iterative algorithms have already generated some semantic matches between the ontologies and then use those generated matches as their base. Domain experts validate the semantic matching rules after non-iterative and iterative mapping generation has occurred to modify or remove any generated error links.

The ONION approach is useful for semi-automatic generation of semantic matching. The approach also seems useful for an open environment that supports the addition and removal of data sources where no requirements for global information retrieval exist and where Ontology alignment matching relations can be easily maintained. The mapping analysis is conducted based on limited semantics and known relations and application dependent rules. The introduction of new semantic relations in an application domain model is difficult.

INFO-MAP

The IF-MAP [50] project presents a theory and method for automated Ontology mappings that is based upon channel theory, a mathematical theory of semantic information flow proposed by Barwise and Seligman [48]. The theory is based on a formal concept analysis of the knowledge domain and the type and instance inference utilised to deduce the equivalent concepts across the source and reference ontologies. The approach formalises the notion of an Ontology, Ontology morphism and Ontology mapping and links them to formal notions of local logic and logic info-morphism stemming from Information Flow theory.

The IF-MAP approach requires a thorough information specification of the type and instance description in order to conduct the concept analysis. The semantic mapping between equivalent concepts can be generated automatically, but the quality of the mapping is not ensured sometimes, e.g., when concepts share the same type and instance descriptions but use different semantics. In this case, further manual validation may be needed. A semantic mapping is established at the level of conceptual mapping based upon the prerequisite of sharing common attributes, type and instance descriptions.

The Ontology morphism generation can automate the process for finding concept-to-concept and relation-to-relation mappings between source and reference database schemas. A formal concept analysis requires a shared lexical structure for the knowledge domain.

Semantic learning of Ontology mappings

Wiesman and Roos [96] proposed a learning-based approach to establish conceptual mappings between two ontologies. The learning method is based on exchanging instances of concepts in the Ontology contents. This approach aims at resolving the

main issues of structural and semantic heterogeneities using an agent infrastructure. Structural heterogeneity refers to the different representation of same data. Semantic heterogeneity concerns the intended meaning of the described information.

Agents exchange flattened instance utterances to establish a joined attention. This approach identifies a corresponding concept in a target Ontology through calculation of the appearance probability of the particular words in the utterance from a source Ontology. The conceptual similarity is measured and assigned a probability value. The concept with a maximum probability value is considered as the correspondence concept. The estimation is calculated based on conditional probability theory. This approach presumes two ontologies describe the same set of instances with different representations. The approach measures the similarity for all instances to find the identical pair of instances in two ontologies. Hence the value transform rule can be derived as a combination of set of predefined functions upon the plain string. Thus, the mapping between two concepts can be marked. The information representation can be transformed between ontologies via established mapping functions.

This approach can establish mappings automatically without the necessity for domain knowledge. But there are a few constraints in order to do this: Firstly, two ontologies must be represented in same language. Secondly, the same string fragment has to appear in the other Ontology describing the identical instance. Thirdly, ontolgies must have at least one identical instance. Finally, this approach only solves the heterogeneity problem at limited level, i.e. plain text matching. It is not suitable for complex semantic heterogeneity situations, for example with database model heterogeneity that involves unit conversion and context translation.

BUSTER

Buster [93] is a hybrid RDF-based Ontology system. It is developed at a global level for content-based retrieval, it supports location reasoning. Additional features can also be defined using formal semantics of a Description Logic. A "concept@location" query is supported for finding of information sources. Terminology and special information integration is achieved according to content classification using TBox reasoning. A terminology query is conducted in terms of simple terminology queries, i.e. reasoning about the user query terminology in relation to registered terms in the Ontology i.e. a user can select and define their own concepts with Ontology support.

This has resulted in a number of systems that provide user interfaces and intelligent reasoning services, to access and integrate information sources. A metadata repository, called a Comprehensive Source Description or CSD has been developed at a global level to provide information source descriptions to facilitate additional services such as data integration, data translation and the addition of new features.

Direct and indirect Matching of schema elements

This approach [98] considers semantic correspondence between different database schematic views as a set of direct and indirect element matches, each of which binds a virtual source schema element to a target schema element through appropriate manipulation operations over the source schema. Direct mapping indicates a semantic correspondence between source and target schemas using synonym relations. Indirect indicates the binding of semantic correspondence between source and target schema involves an appropriate matching algorithms operations. A matching algorithm includes a different approach to set up schema mappings w.r.t. schema element and data values. Characteristics of both intensional and extensional data, e.g. synonym relationship, data value characteristics, expected data values and structure comparison, have been considered as key factors of algorithm input. A confidence value is calculated using combined output of matching algorithms representing the similarity of possible correspondence pairs.

3.2.3.3 Classification of Semantic Data Integration Approaches

An explicit conceptualisation of computer-processable knowledge is useful to support information integration of heterogeneous data resources. An Ontology is recognised as a powerful approach to wrap data sources and to specify the underlying knowledge in a computer-processable format. Ontology merging or alignment can solve the problem by providing semantic mappings to bridge between different Ontology models. If data sources are structured radically differently, are semantically difficult to equate, if only a few specific relations between local ontologies need to be maintained, then alignment seems more expedient. In contrast, if data sources are structured similarly, are semantically similar and more relations between local ontologies need to be maintained, then the merging approach seems more expedient. A comparison of the key approaches in the survey is summarised in Table 1.

Table 1 Comparison of related work with respect to the type of Ontology approach they use for data integration.

	Focus	***Domain data model type***	***Use of Ontology***	***Ontology creation process***	***Ontology model***	***Semantic integration***
Carnot	*DB integration of independently developed data sources.*	*Single domain no partitioned model*	*Selective CYC model containing relevant info. for local data source schema*	*Data- driven*	*CYC and KRBL*	*Merging*
ONION	*Ad-hoc dynamic Ontologies with different structure and representation language*	*Single domain with no partitioned model*	*common conceptual model referenced to a lexicon*	*Data-driven*	*Horn Clauses and RDF*	*Alignment*
Info-Sleuth	*DB integration with heterogeneous, distributed information sources*	*Single domain with no partitioned model*	*Database schema, conceptual model, agent description and reference to standard lexicon*	*Data-driven and process-driven*	*OKBC*	*Merging*
Dome	*Open corporate B2B domain with different service role views*	*Single domain with partitioned models*	*Content-based resource location and conceptual knowledge of integrated data sources*	*Data-drive / Service driven*	*CLASSIC*	*Merging*
IF-MAP	*Multiple Ontology alignment*	*Single domain with no partitioned model*	*Conceptual mediator shares common understanding between different sources.*	*N/A*	*Horn Logic and Prolog*	*Alignment*
OBSERVER	*Global info. system*	*Single domain with no partitioned model*	*Conceptual wrapper of data source, Ontology mapping*	*Data-driven*	*CLASSIC*	*Alignment*

			repository consists of both vocabulary and conceptual relations.			

The process of information integration from heterogeneous resource consists of the creation and maintenance of explicit descriptions of metadata, mapping processes between metadata models, and mapping processes between metadata to data models. In Table 1, the Ontology methods used in different projects are categorised with respect to their focuses and the Ontology modelling and integration process. The Ontology model can be maintained to accommodate different types of metadata instances for a domain.

Table 2 Comparison of related work with respect to Ontology mapping and query translation.

	Type of Ontology mapping	***Mapping Process***	***Mapping representation***	***Query Translation Process***	***Info. Query language***
Carnot	*Attribute mapping and simple value mapping*	*Manual*	*Logic articulation axiom*	*Mapping rules and proofs using articulation axioms*	*SQL-like*
ONION	*Conceptual mapping with given semantic relations*	*Semi-automatic articulation rules based on reasoning about common relationships*	*Binary relations and Horn Clauses*	*Not specified*	*N/A*
Info-Sleuth	*Attribute and value mapping*	*Manual*	*Template-based Query Mark-up Language (TQML)*	*Rule-based reasoning*	*SQL/KIF*
Dome	*Attribute mapping*	*Manual*	*XSLT-like rules with pre and*	*Terminology substitute with*	*SQL-like XML*

			post-conditions	*Rule-based reasoning*	
IF-MAP	*Conceptual mapping*	*Automatic, Channel theory and ,formal concept analysis*	*RDF*	*Not specified*	*NA*
OBSERVER	*Conceptual mapping*	*Automatic*	*Not specified*	*Terminology substitute and query plan mapping and decomposition*	*Descriptio n logic*

Semantic mapping between Ontology models is regarded as an essential element when dealing with the semantic interoperability amongst individual knowledge models. In Table 2, the mapping approach is analyzed and compared further during the process of query transformation between Ontology models.

Table 3 Comparison of related work with respect to query accuracy, query transparency and data source integration

Info Integration System	***Query Accuracy***	***Query transparency***			***Data source integration***
			High-level data query language	***Use the meta data repository***	
Carnot	*Schematic integration with selective info.*	*Yes*	*No*	*No*	*Structure and semantic*
ONION	*Yes*	*No*	*No*	*No*	*Syntactic and semantic*
InfoSleuth	*Yes*	*Yes*	*KIF*	*Multiple criteria(content and service based)*	*Structure, syntactic and, semantic*
Dome	*Yes*	*Yes*	*XML*	*Content-based*	*Structure and syntactic*
IF-MAP	*Lexicon structure analysis*	*No*	*No*	*No*	*Syntactic and semantic*
OBSERVER	*Controlled query expansion in other Ontology.*	*Yes*	*Description Logic*	*No*	*Structure, syntactic and semantic*

An attribute mapping, see Table 3, searches for the exact string matching between the attributes of corresponding conceptual entities that have synonym relations. A conceptual mapping goes further. It browses different conceptual structures across multiple Ontology models to discover the corresponding entities with equivalent semantic meanings. The equivalent semantic meaning can be identified by discovering a common set of attributes in the lexicon structure or through sharing a common set of instances in a close information world.

3.3 Multiple User Views of Data

3.3.1 Logical Data views Versus User Views

Thus far, this survey has focused on query management of an IR system and more specifically on the use of relational model or SQL type approaches and semantic based approaches to support an interoperability and integration of multiple heterogeneous autonomous database sources within the same application domain. Each heterogeneous data source in the integrated IR system has its own data model and potentially the user could see multiple views, one for each heterogeneous data source although such a IR system usually offers a global as view approach to mask the differences of the heterogeneous data schema for each data source.

In earlier IR systems, users of the data were required to understand the logical or database designer's schema for each local database or to understand some common or global database schema that harmonises the different local schema into the same view in order to query a database. Later, additional abstractions were added e.g., the ANSI/SPARC architecture allows users to have more abstract view of the data than the logical schema of the shared data. There can also exist multiple user views for a set of data sources in an integrated IR system. Next, this survey focuses on support for the presentation layer of an IR system and more specifically on techniques to support multiple viewpoints representations and result adaptation.

3.3.2 Projects and Applications

Sheth and Larson [85] have proposed a five-layer architecture for federated database systems as a modification of a conventional three-layered model of a centralised database system in order to support knowledge distribution, information heterogeneity

and conceptual anatomy amongst database stakeholders. The five layers include the local schema, component schema, export schema, federated schema and external schema. Local schema is a local data model representation of database components. Component schema is derived by translating a local schema to a common data model. Export schema presents a subset of common data model for integration use. A federated schema is an integration of multiple export schemas. An external schema defines a schema for user or application use. Two types of mapping approach have been identified to conduct schema translation between layers: explicit mapping and constrains rule mapping. The former gives exact mapping relations between corresponding entities. The latter specifies rules that how schema constraint is translated during mapping. Component schema is a derived view over local schema, whereas external schema is a derived view over federated schema.

Layered view adaptation [9], [49], is a common approach to solve multiple representation of information system on the basis of a specific user and application perspective. The representation adaptation is decomposed into layers so that a specific change of data schema and objects can be limited into certain scopes and the reusability of information system can be maximised.

In Adnani et al. [9], a multi-layered functional data model is presented to support multiple representations and information sharing among different application views in GIS domain. Layered model has separated primary concept and composite concept to enable dynamic representation of object and classes. Identified layers include geometric layer, functional layer and domain layer that provide the corresponding representations with respect to the basic geometric types, common function based on geomantic types, and specific functions in the domain. The cross layer schema derivation is achieved via inheritance and class composition. The distributed representations of these types were mapped using equivalent and aggregation relations across layers.

Multiple representation of domain knowledge was classified into two dimensions of schema change and object change. Schema view adopts a traditional database view that is a derived relation from the integrated schema model, and object-oriented hierarchy structures. Object view indicates the multiple classification problems. i.e. one single instance may belong to multiple information classes, its property may change during the life-cycle. Multiple representation of an object can be achieved via a role mechanism. A role is an object like structure with set of properties, behaviour and

semantics. An object can belong to different classes corresponding to its roles. Dynamic object association, one object can change from one class to another during its evolution life-cycle. It results in an introduction of a role. A role is an alternative classification of an object, such that an object may become a member of several role classes, remain a member for some time and then release its membership [89].

Ribière and Dieng-Kuntz [80] have proposed a multiple viewpoint solution to reconcile diverse developer interpretation processes upon the domain knowledge. The viewpoint here is defined as different terminologies and instance category relations within the domain knowledge: "an interface allowing the indexation and the interpretation of a view composed of knowledge elements". A viewpoint is characterised by its consensual and non-consensual interpretation of is-a relations and the use of a terminology. Each individual viewpoint defines an instantiation of general viewpoint template for certain type of Ontology experts. A common basic concept is instantiated via different is-a relation in different viewpoints to reach different instance object in the final representation.

The DIF (Design Information Framework) [49] knowledge system supports a translated, collaborated and integrated multiple user viewpoints via a consistent and explicit representation of metadata and data type information . The metadata of a data instance is organised into two layers including DIP(Design Information Primitive) and DIL(Design Information Elements). Primary and basic types such as attribute, entity, time and act are defined as basic units in DIL that can not be further decomposed. The basic units are used to build high level concepts of function, goals and profile in DIP. PDIF (Project Design Information) is composed of multiple sets of DIF elements representing the different interest's intension and acts of project groups. The metadata are structured in hierarchy tree with instance table for each project DIF. A DIL element is a composite set consisting of DIP basic units.

Benchuka and Boufaida [18] proposed an dynamic extension approach for the object-oriented database model. The single integrated database schema is extended at multiple levels: role, view and viewpoint in order to improve the representation flexibility and access interoperability amongst different application and users. A viewpoint is constructed on the basis of partial knowledge of the referential model. A view reflects an extracted external schema of a database with a generalisation hierarchy change. A role defines a dynamic schema with type and attributes change from viewpoint to cope with user viewpoint. A viewpoint schema is obtained in two steps: at first, a projection

operation is carried out on the referential schema to select the part of it, which will be described according to the considered viewpoint. Then, an extension operation of the resulting schema customises the entities description according to the viewpoint. Dynamic evolution of views can be achieved via this adaptive model through different levels reflecting upon complicated real-world representation.

Regarding the information heterogeneity discussed in the previous chapter, knowledge representation and interpretation difference are classified into sub-types including system, syntactic, conceptual, terminology, convention and semantic heterogeneities. The multiple viewpoint representation and access to domain knowledge indicates adaptation of diverse user interests to a common agreement of the knowledge representation. The viewpoint adaptation mainly concerns a dynamic representation in terms of conceptual, terminology, convention and semantic heterogeneities during user interest evolution, in dimensions of coverage, granularity and perspective. The variation of classification representation mostly relies on a developer's intended usage of the domain information. ONTOWeb [56] has suggested that it analyse conceptual related problems at three abstract levels, coverage, granularity and perspective. Coverage actually identifies user interests as a portion of the domain knowledge. Granularity gives the level of a hierarchy for a user's understanding of the knowledge representation. Perspective indicates the beliefs or notions to convey the hypotheses, facts, and assumptions that form the contents of a viewpoint independent of how the beliefs are expressed [12]. Table 4classifies related work that supports multiple viewpoints depending on the basic form of the viewpoint model in terms of conceptual, terminological, convention or semantic w.r.t. information heterogeneities as defined in section 3.1.2.

Table 4 Comparison of multiple viewpoint systems with respect to the type of information heterogeneties

Surveyed System	***Conceptual***	***Terminology***	***Convention***	***Semantic***	***Derivation approach***
Sheth and Larson[85]		√		√	*SQL view, terms mapping*
Adnani [9]	√				*Instance category*
Ribière and Dieng-	√	√			*Terms mapping, instance category*

Kuntz [80]					
Jung [49]	√	√	√		*Instance category, concept composition*
Benchikha and Boufaida, [18]	√		√	√	*SQL view, instance category, role*
Calvanese [26]	√	√		√	*Instance category, SQL view,*

Additionally, the surveyed approaches are analysed in terms of their supports to viewpoint adaptation at different abstract levels of representations w.r.t. coverage, granularity and perspective, see Table 5.

Table 5 Comparison of multiple viewpoint system w.r.t. coverage, granularity and perspective

Surveyed System	***Coverage***	***Granularity***	***Perspective***
Sheth and Larson[85]			√
Adnani [9]		√	√
Ribière and Dieng-Kuntz [80]	√	√	
Jung [49]	√		√
Benchikha and Boufaida, [18]	√	√	
Calvanese [26]	√		√

The surveyed approaches in Table 4 and Table 5 have shown that a user viewpoint can be derived from a primary schema or a common knowledge representation via a transformation operation resolving specific types of heterogeneities. Thus the consistency amongst viewpoint representation can be satisfied. However it may be rarely the case regarding user's IR demands in a real physical domain, where viewpoint conceptualisation may be generated from independent knowledge

representation containing coexistent heterogeneities. Some common drawbacks of surveyed systems are summarised in terms of:

- A user's view in terms of their understanding and preferences is often not considered when retrieving information.
- There is a lack of overall support for flexible types of adaptation in the viewpoint representation, i.e., to combine coverage, granularity and coverage.
- Viewpoint representation and conceptual adaptation are less supported with formal standard framework. It makes reuse of such model by different applications is difficult.
- No explicit well-defined process has been defined to adapt information retrieval to the user view and to support evolving or changing views and domain models.

3.4 Integrating Semantics, Rules, Logic and Databases

Thus far, the surveyed work has focussed on using a semantic approach to :

- Mediate between, and to reason about, different semantic and syntactical data models that are maintained and related to the relational schema of the data sources;
- Mediate between different user views of the relational schema of the data sources using a semantic model.

The benefits of using a semantic model for database integration have been highlighted. Using an Ontology representation in information system has significantly improved the ability to solve the problems 1,2,3,6 and 8 in Section 3.1.3.2, however some of other issues need further research. There are also some fundamental issues when dealing with Semantic Web and Database Integration that have not been explicitly raised. This is mainly because the approaches discussed so far haven't used the semantic model to attempt to reason about the relational model schema such as what can be said about queries that return no results but rather reason about derived semantic conceptualisations of the relational model schema. The main challenge here is that the database relational models operate under a closed world assumption whereas the Semantic Web operates under a closed world assumption. Reasoning under an open world assumption can infer information about a closed world model that conflicts with it or causes the data integrity of the closed world model to be reduced. Reasoning

using Semantic Web models that involves rules are constraints, are often needed in practice, but there is still a lack of agreement about whether any single way to interlink rule-based models, logic models and conceptual models is more beneficial than any other way. As a result there is as yet still no standard way to interlink these models in a system, see chapter 2. This challenge and some projects that have attempted to address this issue are now discussed in more detail.

Ontology models developed on the basis of description logic have been described in chapter 2 but this is briefly reviewed here again in order to lead to problems of combining open world and closed world semantic models. A DL-based information system comprises two components, the TBox and the ABox. The Tbox introduces the terminology, i.e. the vocabulary of an application domain, while ABox contains assertions about named individuals in terms of this vocabulary. The ABox of Ontology model can be seen as a relational database with only unary and binary relations. The semantics of relations amongst concept, property and individual are imposed in TBox, which does not exist in the relational data model.

An important semantic distinction between Ontology and database is so-called "open-world" and "close-world" assumption, i.e. ABox of Ontology indicates one of subset of information model satisfying the TBox, it may be incomplete as more assertions can be inserted at any time, whereas a database is a completed data model. As a consequence, absence of information in a database is interpreted as negative information, while absence of information in an ABox only indicates lack of knowledge [11]. Inconsistencies can arise when system conducting information reasoning within a knowledge model.

A relational view over a database indicates a designated query to retrieve a data instance according to the schema, whereas an ontological viewpoint contains more content involving different representations of conceptual structures and relations upon the domain knowledge. Since each view over database can be derived from original database schema via relational operations of projection, selection, join and rename in a straightforward way, see virtual table [66], this ensures the consistency between two data models during the process of derivation. However, an ontological viewpoint may contain open information about domain knowledge, where representation confliction may exists in terms of different types of information heterogeneities.

Instances data retrieval to Ontology model via a conceptual viewpoint can be reduced to SQL queries over relational view if no further information inference is involved. By

that means, tuple-set database is considered as a closed subset of ABox assertions in the knowledge domain. Thereafter well-established relational view approaches for database can be adopted here to support data queries posed on different viewpoints.

Reasoning is an important feature in a description logic framework and is used to support information inference. Logical relational schema data integration assumes that each source is basically a database, i.e. a logical theory with a single model, such an assumption is not made in Ontology integration, where a local Ontology is an arbitrary logical theory, and hence can have multiple models [26].

Damasio et al. [32] consider closed-world reasoning in which negation-as-failure is the only negation mechanism supported. They then propose two major extensions to the semantics to better support open world reasoning: answer set semantics and well-founded semantics with explicit negation. These can be used to support two forms of negation, weak and strong. Weak negation is similar to the mechanism of non-monotonic negation-as-failure, and strong negation allows the user to express negative knowledge and is monotonic. The combination of these two forms of negation allow the distinction between open and closed predicates, is illustrated in the paper but practical computational versions of their model are not given. Pan and Heflin [74] present, DLDB, a knowledge base system that extends a relational database management system with additional capabilities to store and query DAML+OIL inference. The most significant aspect of theory approach is the use of a description logic FaCT reasoner to pre-compute the subsumption hierarchy in order to flatten it to be stored in relational database issues. However, they do not consider closed world vs. open world semantic issues.

In addition, since the early 1990s there has been much work that preceded the uptake of the semantic web and description logic based approaches that have looked at extending database models to support logic based reasoning about the database data, so called deductive databases [33]. Perhaps the most well-known based upon the Datalog but there are many others [21]. Datalog aims to separate out facts that relate to a closed world in an extensional database part from inference rules that can derive other data from facts in an open world in an intensional database part. It extends relational models but without negation and recursion support in the inference. Patel-Schneider and Horrocks [75] consider Datalog in relation to classical logics such as First-Order Logic and Description Logics, and their use as underlying formalisms for the Semantic Web. They argue however, that although they are similar, they have important

differences at more expressive language levels and that after considering some of these differences, they argue that, although some of the characteristics of Datalog have their utility, the open environment of the Semantic Web is better served by standard logics. De Bruijn et al. [34]have undertaken a recent survey of the attempts by the Semantic Web community to combine classical, first-order logic and various description logics, with rule languages rooted in logic programming such as SWRL (a Semantic Web Rule Language Combining OWL and RuleML), dl-programs, and DL+log and highlight that they differ significantly in the way ontologies combine with (nonmonotonic) rules bases. However, each of these approaches overcomes the differences between the first-order and rules paradigms (open vs. closed domain, non-unique vs. unique names, open vs. closed world) in different ways and vary with respect to the ease of implementation and availability of reasoning techniques. There is as yet to clear recommendation for combining logic and rules. Ng [69] also considers the issues of combined Open and Closed world and Rules and Queries in a common model using two use cases from Industry. They have outlined the necessity of a notion of negation-as-failure within these use cases and propose an extension of OWL that supports two additional operators to support this and have provided an implementation approach using only open-world query answering services.

3.5 Summary

Semantic, Ontology, models offer powerful benefits for use to mediate between and to reason about heterogeneous data sources, when data from multiple sources needs to be combined. A critical survey of related work has been conducted and has classified these with respect to: the type of Ontology approach they use for data integration; the types of Ontology mapping and query translation they use and the types of query accuracy, query transparency and data source integration they use. The surveyed integration system has been summarised regarding their supports the whole range of these characteristics, see Table 6. This illustrates the important point that all the surveyed approaches are at best only a partial solution to fulfill the application domain requirements given in Section 4.2.4.

Table 6 Summary of surveyed project limitations in relation to the domain application requirements

Integration System	*Interoper-ability process*	*Heterogeneities (see Table 6) resolved*	*Best Practice*	*Limitations (other than certain types of heterogeneity from Tab)*
Carnot	*Merging*	*Syntactic and Terminology*	*Common semantic reference to Cyc ontology, articulation axioms proofing for the synonym translation. Query translation*	*Metadata model and logic reasoning model. Does not support data harmonization.*
ONION	*alignment*	*Syntactic and conceptual*	*Semi-automatic articulation rules based reasoning for given common relationship, lexicon analysis for synonym matching*	*query processing model, query transparency, data harmonization are not supported*
InfoSleuth	*merging*	*Syntactic and, conceptual*	*Rich metadata set for conceptual model, database schema and agent service. Query transparency and metadata provenance.*	*Data harmonization, query augmentation, data quality control*
Dome	*merging*	*Structure and conceptual*	*Query transparency, application and physical storage independency*	*Data harmonization, query augmentation, data quality control*
IF-MAP	*alignment*	*Conceptual and Terminology*	*Formal conceptual analysis, automatic semantic mapping extraction*	*Query transparency, Data harmonization, query augmentation, data quality control*
OBSERVER	*alignment*	*Terminology, syntactic and conceptual*	*Query transparency and data quality control*	*Data harmonization, query augmentation*
PROMPT	*merging*	*Syntactic and conceptual*	*Data quality control*	*Data harmonization and query augmentation*

Secondly, semantic models can be used to project and mediate between different user views of the relational schema of the data sources. A critical survey of related work has been carried out and classified with respect to different types of views such as conceptual, terminological, convention or semantic and on dimensions of views such as coverage, granularity and perspective. No surveyed system enables user views to be generated based on all three dimensions of coverage, granularity and perspective.

The third part of the survey concerns more fundamental issues of how logic based, semantic approaches, database systems and rule based systems can be combined. Semantic models can be used to reason about indirect, derived, semantic conceptualisations of the relational model schema but there are additional challenges when semantic models are used to directly reason about relational model schema such as when considering what can be said about queries that return no result. The main

challenge here is that the database relational models operate under a closed world assumption whereas the Semantic Web operates under an open world assumption. Reasoning under an open world assumption can infer information about a closed world model that conflicts with it or causes the data integrity of the closed world model to be reduced. Reasoning using Semantic Web models that involves rules and constraints is often useful in practice but there is still a lack of agreement about whether any single way to interlink rule-based models, logic models and conceptual models is better than any other way. As a result there is not a standard way to interlink these models yet.

In the next chapter, a comprehensive agent-based semantic framework is developed and applied to support queries of multiple heterogeneous database sources in the inland water domain. The framework is designed to support a range of characteristics that were used to classify the surveyed systems.

Chapter 4 A Method for the Semantic Integration of Inland Water Information

4.1 Introduction to the Inland Water Domain

The Inland Water or IW quality domain concerns water quality data queries and analysis and comparisons of chemical and biological measurements of water quality indicators, over space and time. Raw data in database repositories that were distributed physically in different countries, autonomously developed, managed and processed in accordance with disparate national and international environment monitoring programmes, have been integrated. The semantic data integration application for the IW domain was researched and developed as part of the EU funded EDEN-IW Environmental Data Exchange Network for Inland Water, project (IST-2000-29317).

The EDEN-IW project aimed to develop a service integrating disparate, heterogeneous, government databases on inland water at a European level. It aimed to make existing distributed environmental data available to researchers, policy users and citizens through an intelligent interface acting as a one-stop shop for them. Users, who may also be public authorities, e.g., environmental regulatory agencies, and the public, will be able to address their needs for Inland Water data through one common intelligent interface, independent of physical or logical location of the databases providing information. The user should not need to know the database query languages used, or the specific nomenclature used in a specific database, or indeed know which database or databases contain the relevant information. The prototype operated on a limited number of databases and in a limited number of languages.

The remainder of this chapter is structured as follows. Section 2 gives the motivation and requirements. Section 3 introduces the method developed for IR and reported using two main information system models. An Ontology based framework is presented in section 4, then a multi-agent system framework is presented in section 5. The combined system implementation and application is described in Section 6. A summary of this chapter is given in section 7.

4.2 Motivation and Requirements

4.2.1 Information Retrieval

The major requirements for Information Retrieval (IR) from distributed heterogeneous databases are to support: query transparency, data quality, data source aggregation, harmonisation of heterogeneous data sources and metadata management.

By *query transparency*, it is meant that users need not be concerned with the access details of the data source to answer the query such as the location of the database and the data within the database, the schema used to store the data in the database or a particular vendor's relational database management system (RDBMS). Query transparency is difficult to support using a pure standard RDBMS model as metadata to locate data structures as tables within a database is poorly standardised and is hindered by the flat autonomous table organisation within the RDBMS. In addition, there is no inherent standard mechanism within the RDBMS model itself to interlink databases and to locate and identify which database holds specific data.

Data aggregation requires a data model that can reach across multiple heterogeneous databases - a metadata model. Metadata is data about data that describes indexes and characterises of the stored data. When multiple databases need to be queried, the metadata managed as a metadata repository or directory, is first typically queried to identify candidate data sources, else queries would need to be sent to each individual data source, leading to a poor information retrieval performance.

The problems of poor data quality are well known. Amongst the most widely recognised ones are the so-called missed positives, false positives and data anomalies. In the first case the system fails to retrieve relevant answers to the query whereas in the second case the system retrieves answers that are irrelevant to the query. A system should seek to minimise both of these. Within each individual database, database transaction management supports the so called ACID (atomicity, consistency, isolation and durability) concepts and good data model design can reduce data redundancy and the existence of insertion, update and deletion anomalies. However, quality individual database design and management can still lead to variable data quality across autonomous databases because of information heterogeneity and information redundancy.

The EDEN-IW system can be described as a virtually integrated IR system. Data integration is the process of combining several data sources such that they

may be queried and updated via some common interface[62]. A common data model is defined at the global level, to which access to local data sources can be mapped. The design of data integration system can follow two different approaches with respect to explicitly managed data by the system: virtual or materialised integration [27]. In the virtual approach, data residing at sources are accessed during query processing, but they are not replicated in the integrated system. In the materialised approach, the system computes the extension of concepts in global schema by replicating data at the sources. [27] A virtually-integrated information retrieval system separates a canonical information representation from the processing logic, for example water quality data in different data sources can be compared and analysed in different ways. Materialised integration has major difficulties such as the need to refresh data to keep it "up-to-date" and to maintain consistency between data at global replication and sources. A major challenge of the virtual approach concerns complexity because of the need to align and merge heterogeneous distributed representations of metadata and data. EDEN-IW used an extended virtual approach including a partial data representation in global and local conceptual schema.

Monitoring and data processing systems for water quality require information retrieval and access to national-based inland-water databases that were physically distributed in different research institutes. The knowledge representation and logic structure of databases are very heterogeneous in part because the classification and structuring of domain knowledge is conducted at a local, application level without referring to any canonical standard. The knowledge representation and conceptualisation are the choice of the local database developer and administrator regarding the particular purpose of a particular information application and processing programme.

The data content in inland-water databases although physically distributed may hold environmental information about the same or closely related water bodies. For example, a river may flow through several countries; the quality comparison of upstream and downstream of the same river may request information gathering and analysis involving different databases. The measurements of water samples are expressed in different conceptual structures and coding formats according to the particular purpose and focus of local research institute, which may be expressed as observations of different parameters and analyse programmes. The data content in separated databases may overlap and need to be correlated.

Databases of inland-water information have been developed, used and maintained over decades. The data structure and data representation in these 'legacy' databases reflects the processing intension of organisations that maintain them. A majority of these databases have been established long before distributed services such as "public access", "Web services" and "e-government" were envisaged. Data retrieval is commonly organised using relational database systems and normalised tables but the underlying meta-data, other than the primitive data types used for table columns, is often not available on-line or standardized.

Inter-regional quality measurement and trends monitoring can be investigated by establishing a virtually integrated information consortium for the water domain. Hence global retrieval and access can be achieved with all local details such as physical location and logical structure remaining hidden. However, the target is hard to achieve for several reasons: information heterogeneities at different levels can be interleaved, information entities have different conceptual perspectives for knowledge perception and representation, inaccessible underlying knowledge that is not in a computational form for global exchange and the maintenance problems for distributed data and metadata that result when supporting the evolving use and extensions of databases.

4.2.2 Information Heterogeneity in Inland Water Domain

Regarding the heterogeneity types given in section 3.1.2, the Classification of information heterogeneity for IW is summarised in Table 7.

Table 7 Classification of information heterogeneity

	Problem	***Solution candidates***	***EDEN-IW Examples***
System	*Interoperability between different platforms*	*JDBC adaptation, COBRA, wrapper service for data sources, and general query syntax*	*Different legacy RDB systems: Oracle, Access, SQL Server*
Syntactic	*Structure and representation formats*	*Logic translator programming*	*Language translation between RDF and SQL*
Conceptual	*Classification of domain knowledge*	*Conceptual mapping*	*Different database schema and Ontology structure*
Terminology	*Linguistic problems*	*Canonical glossary and thesaurus, data dictionary*	*Multi-lingual support, schema element abbreviation,*

Convention	*Expression of underlying knowledge regarding usage conventions*	*Procedure-oriented conversion*	*Different coding formats for time, unit, coordinates.*
Semiotic (usage and perspective change)	*User may have different levels of understanding with respect to different coverage, granularity and expression*	*Separate user ontology and knowledge representation from common understanding.*	*Different user preferences and viewpoint conceptualisation.*

Mediation techniques have been developed to overcome particular types of information heterogeneities of the query syntaxes and the underlying data schemas. However, in practice, more than one type of heterogeneity may be interwoven with that of another, introducing overlapping heterogeneities. A composite approach is needed to solve this problem. In the inland water domain, the integration of heterogeneous information mainly aims to focus on syntactic, conceptual, terminology and convention heterogeneities, such as anomalistic naming and abbreviations, disparate data value representations, and multi-lingual terms, that emerge during the integration of mismatched database schema and the underlying data modelling. For example, the Danish definition of parameter A corresponds to the French observation relation in the context of "parameter X observed in medium Y analysed in fraction Z and expressed in unit U". The context contains multiple heterogeneities such as non normalised relational, mismatched database schema, non-canonical naming conventions and multi-lingual terms.

The coexistence of multiple types of heterogeneities is a major challenge in an IR system that spans multiple distributed databases, because of the difficulties in classifying and managing domain knowledge using a common single approach. The overlapped heterogeneities introduce an extra interoperability problem under such circumstance in terms of their metadata representation and data reconciliation that may entail information loss.

4.2.3 Heterogeneous Databases in the Inland Water Domain

The following institutions[1] provided data sources for water quality measurement for the use in the EDEN-IW system, see Table 8.

- National Environmental Research Institute (NERI), Denmark
- International Office for Water (IOW), France
- European Topic Centre and European Environmental Agency on Water (ETC/EEA), United Kingdom
- Environment Agency for England and Wales, United Kingdom (UKEA)

Table 8 Heterogeneous databases in IW domain

Database Name	***Physical Location***	***Language***	***Database Type***	***Measurement records***	***Observed Determinand*** [2]	***Stations***
NERI	*Denmark*	*Danish*	*Oracle RDB/ SQL Server*	*348788*	*39*	*553*
IOW	*France*	*French*	*Oracle 9i*	*92278*	*87*	*29*
EEA	*Italy*	*English*	*MS Access*	*189253*	*18*	*3438 stations from 27 countries*
UKEA	*UK*	*English*	*MS Access*	*565225*	*116*	*277*

The NERI inland water database system is partitioned into a number of observation 'programs', where each program has its own set of tables. The observation programs cover both research projects with public access and monitoring programs without public access. The stored data in the IOW database comes from national thematic databanks and the river basin data banks. The technical architecture is based on an Oracle data server and ARC/INFO server for map processing. The differences between NERI and IOW were not restricted to the structure of the databases but also involve the understanding of simple expressions such as water medium, hence producing model and semantic heterogeneities, see Table 9 . A water sample from a lake or river includes small organic or inorganic particles and even fish that can be filtered and

---

[1] The rest of chapter will focus on two major candidates, NERI and IOW to outline the Ontology-driven approach for metadata modelling and virtual database integration, the other data sources are connected to the system using the same approach.

[2] There are about more than 200 determinands have been identified in different databases, whereas 8 out of them overlap regarding their meaningful definitions.

divided into a water and particle phases. A determinand like *Nitrogen* can be found in the water fraction as well as in the particle fraction, or it can be analysed as *Total Nitrogen* in a water sample. It is important to define every determinand to ensure that at least the main concepts are commonly accepted, as the observation may not represent the same meaning in different databases. The issue of how the water sample is treated before analysing is also important.

Although the main concepts may commonly be accepted, local implementations can vary substantially. Similar observations may be handled differently in different database implementations, see Table 9.

Table 9 Different implementations of observations in a French (IOW) and a Danish (NERI) database.

Database 1 (IOW)	Database 2 (NERI)
• Each Observation value is linked to a Determinand and an Analytical fraction (local codes). • Each combination of Determinand and Analytical fraction is linked to a specific Unit defined in a Data dictionary (text document). • The Analytical fraction is implicitly linked to a Medium	• Each Observation value is linked to a Determinand (local code). • The local Determinand name (in Danish) implies the Medium and Analytical fraction. • Each local Determinand is linked to a specific Unit (local code).

The above heterogeneity issues were considered as issues of the underlying domain knowledge that were not explicitly modelled in the databases. In order to conduct an analysis and comparison of water quality across different database systems, semantic correspondences need to be discovered through an understanding of local knowledge classification, during analysis by domain experts.

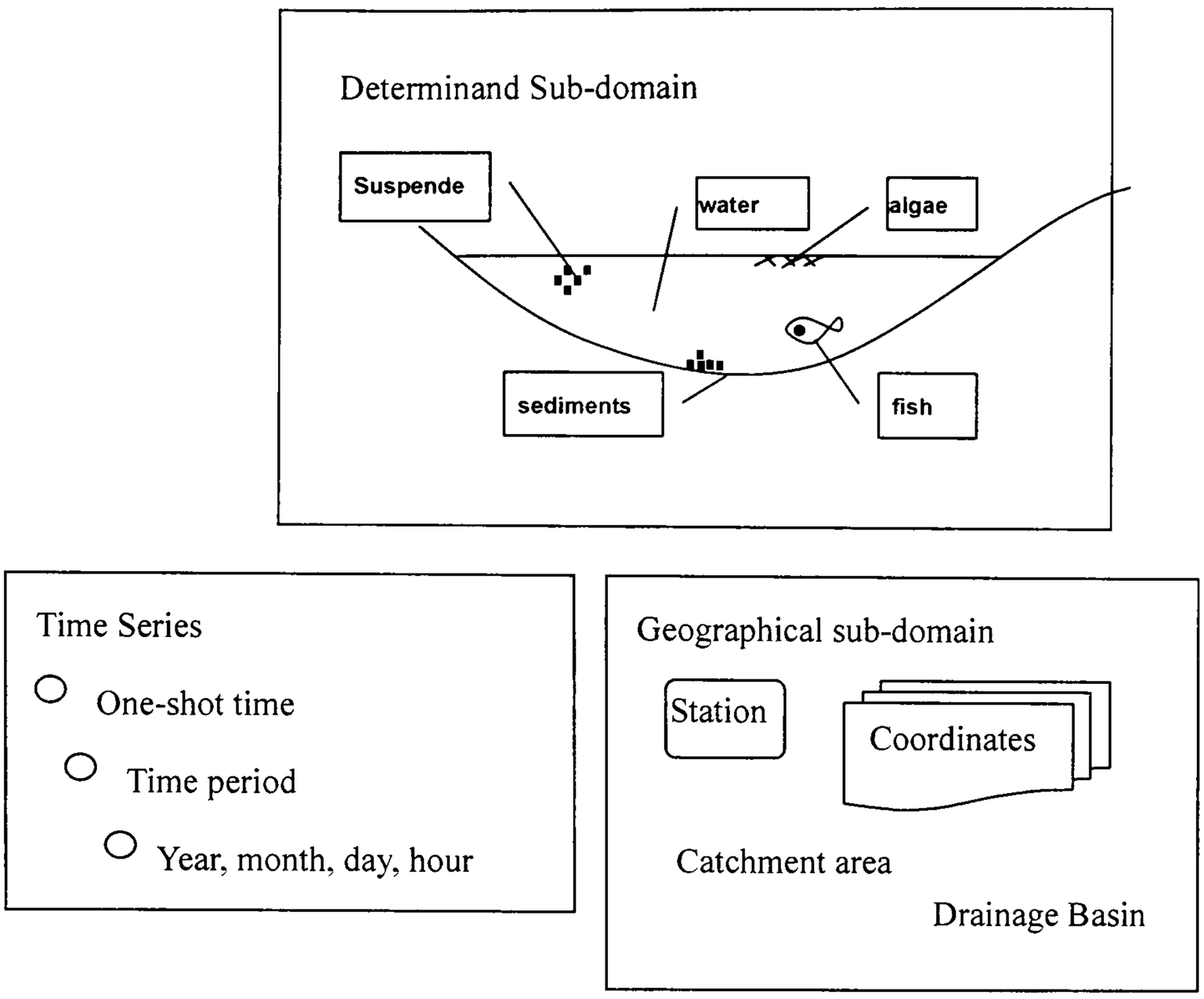

Figure 4. Key concepts in Inland-Water domain

The basic concepts were illustrated in the key scenarios of IR query to give basic examples of how heterogeneous information can be integrated within the IW domain. Some basic concepts are given in Figure 4. For example, Observation is a measurement of a Determinand, e.g., Mercury, in a fraction of a Medium, taken at a Station at a Time and expressed with a Unit. Medium can be classified in some basic categories like water, algae, sediment and fish and suspended particles as shown in the figure. Inland water data are heterogeneous because terms, meaning for example observation and related concepts description, may vary according to scenarios and views in the local database domain. The above concepts may need to be precisely interpreted within a local database domain within the context of the particular query, for example:

- *Query use case 1: What is the concentration of determinand X in medium Y at monitoring station S during the period T.*

- *Query use case 9: At which stations has determinand X been observed above a threshold value Y during period Z?*

Determinand is a dominant parameter in an observation of a water quality sample. Different interpretations of determinand are mainly expressed as non-standard naming, definition, coding-formats and compound groups such as *Heavy Metal* and *Nutrients*

elements. The conceptual indexing of determinand may vary according to the query context of the particular program such as one that queries information about pollution versus one that queries information about hazards elements. Some queries about determinand concentration may refer to certain parameter or compound group that may not be available in a local institute's database. In such case, the request may be semantically related to other relevant determinands that can be substituted.

Medium has a more specialised meaning in IOW than being used in NERI. It indicates the certain medium being analysed in particular fraction. The combination of such information is regarded as background knowledge for local information applications, whereas the same knowledge is expressed as separate concepts with given semantic relations in NERI.

Analytical Fraction: indicates the special part of substance of observed medium, e.g. organicBound and inorganicBound.

Station is a generic concept not only for a geographical sub-domain but also involves some underlying knowledge. The concept includes stations of a varying nature – some representing surface water stations with sub-types Lake Station and River Station, others representing Ground water. The *Station* concept can span or be composed of different types of stations such as an observation point with intermittent observations or monitoring stations with continuous observations, e.g. of water flow.

Concentration is expressed as a numeric value of a certain unit. The unit representation varies. The numeric value may have different meanings according to different scenarios and contexts, e.g. single measurement value, aggregated value or average value. In a special case, the stored concentration value can also indicate the observed threshold when the actual value is too small to be measured.

Time: this varies upon different expression formats according to different context.

The instances of such concepts are realised in local databases as tuple sets according to the different database schema in multiple natural languages using different coding-formats. For example, to illustrate the statement above, the determinand of dissolved O_2, which is defined literally as the "quantity of gaseous oxygen dissolved in water, at the temperature and the atmospherical pressure of the instant of the sampling.", is coded in NERI database with PARAM=400, which semantically corresponds to the description of " determinand Oxygen observed in water with dissolved fraction in unit mg/l" Oxygen is represented in the IOW database with a data fraction of

"code_parameter=1312, code_support=3 unite=6". The direct syntax mapping between terms does not solve the problem of harmonising semantic heterogeneities.

4.2.4 Requirements for Environmental Information Retrieval

The information retrieval requirements for the EDEN-IW system are derived from previous parts in section 2.

Query requirements:

1. *Query transparency*: users need not be concerned with the access details of the data source, such as the location of the database and the data within the database, the schema used to store the data in the database and the idiosyncrasies of a particular vendor's relational database management system (RDBMS).

2. *Query internationalisation*: information access should be multilingual to support information retrieval in an international setting.

3. *Query augmentation*: the user query can be expanded, generalised or specialised to better understand the context.

Data source requirements:

4. *Data aggregation and presentation*: the effects of collected and integrating content from various sources need to be handled. Post-processing of the results such as filtering, ranking and presenting is needed.

5. *Data harmonisation*: harmonisation is needed when internal (proprietary) and external (non-proprietary) information sources differ. This is needed to resolve different possible answer to the same query, e.g., they are measured in different units, or different queries can be analysed to show they are equivalent.

Metadata requirements: generally metadata is needed to facilitate the above data interoperability requirements and this introduces additional metadata requirements.

6. *Application and storage independence*: a metadata model should represent data processing, in terms of the application specific business rules, used to formulate the queries, independent from the stored data. The advantage of this separation is that the domain knowledge can be more easily reused with different sets of application specific operational knowledge.

7. *Metadata provenance*: the metadata used to describe the data should have provenance, i.e., be grounded using concepts from a group such as an International standards group.
8. *Metadata restructuring*: the categorisation, (re)structuring and indexing of the source data by adding metadata that is machine-readable should be supported. This makes the domain assumptions explicit, which in turn makes it easier to change domain assumptions and to understand and update the legacy data.

4.3 An Ontology based Approach for Information Retrieval: EDEN-IW

EDEN-IW can broadly be characterised as a semantic based information retrieval system. In an information retrieval (IR) system, Ontologies are used to guide the search so that the system may return more relevant results and query transformation and post-processing can be conduced automatically without human participation.

Ontologies are conceptual models that can aid knowledge sharing within an application domain. An Ontology is characterised by an explicit semantic model for the conceptualisation of the structures used to represent and manage information, that is machine-readable, and by the consensual nature in agreeing and sharing this model. An Ontology aims to provide a formal model and structure for the domain knowledge on the basis of a common agreement of conceptual domain so that Ontology may be reused and shared across applications and user groups. This involves an explicit description of the assumptions and assertions regarding both the domain structure and terminology.

4.3.1 Ontology-driven Information Retrieval and Interoperability

EDEN-IW is virtually integrated with a global conceptual schema. The use of an Ontology makes explicit the information content in a manner independent of the underlying data structures that may be used to store the information in a data repository[64]. In an information retrieval (IR) application, Ontologies can be used to guide the search so that the system may return more relevant results. The assumption in this class of applications is that the Ontology will allow the IR system a better representation of the concepts being searched and thus make possible an improvement of its performance [56].

The potential advantages of using a semantic integration approach to information integration are as follows.

- It can support *query augmentation* (expansion of a user query using the metadata as a context). If the data domain is originally modelled and represented in a form, e.g., relational database tables, that is not expressive to represent rich organisation structures, the creation and introduction of a more expressive metadata representation such as Ontology can overcome this limitation.
- An Ontology model can support *content re-structuring*, it can be used to classify, (re)structure and index information. There are questions about the scalability of approaches that seek to harmonise schema-based and syntax across heterogeneous databases because of the number of possible heterogeneous schema and the difficulty in normalising numerous syntactical mappings between heterogeneous database schemas. As a result integration based upon models of the semantics of the underlying databases has been proposed as being more scalable.
- An Ontology model can also be used to support the general information retrieval requirements of *data harmonisation*, when information sources differ and to support content aggregation.
- An Ontology model can be organised to support *application and presentation independence*, to support reuse across multiple applications and presentation viewpoints.

However, there are also challenges in using a semantic metadata approach – the chief-one being that heterogeneous local data sources rarely have a common metadata model and even less so a semantic one. Hence, in practice, either local data sources would have to be re-engineered to support this (usually impossible in practice) or mappings must be created and maintained either to link a common semantic metadata model to local data instances (e.g., the local database schema, and in this in turn requires local metadata models to be created to interface to the local data) or to link different local metadata models to each other without using a common metadata model. Metadata conceptual models such as Ontological models often do not have to define explicit data types such as Integer or String, however, computation software and databases require explicit data types, type metadata that must be incorporated into the Ontological model.

Whereas, there exist mature and robust models, processes and tools for maintaining the quality of stored data in RDBMSs, these are not so robust and available for use to maintain the quality of the metadata.

Finally, an important challenge during Ontology creation is that whilst a consensus regarding the concepts, structure and scope of a model can be achieved within a community, many different communities can promote their local Ontology model to a global community as being "the" domain model for a particular domain - this raises the risk of a lack of interoperability between different Ontologies within the same domain and the risk that a badly formed and defined Ontology for that domain could take hold. One way out of this conundrum is firstly to ground or reference parts of a domain Ontology in terms that have international provenance.

4.3.2 Aims of the EDEN-IW Ontology

The targeted aims of the EDEN-IW Ontology can be derived as:

- A consistent representation of knowledge in the EDEN-IW application to enable a common understanding among different components in the system (*Content management);*
  - A common view of heterogeneous resource files regarding the EDEN-IW knowledge support (*Content harmonisation, Content management*);
  - A unified knowledge representation over different language domains (*Content harmonisation, Content management*);
  - Knowledge Mediation between different user views, e.g. database owner, information retrieving and Decision Support System (DSS) (*Query augmentation, Content harmonisation, Content aggregation/presentation*);
  - The information retrieving system is independent of the domain knowledge (*Domain knowledge / operational knowledge separation*).
  - A unified representation for the local underlying knowledge to enable metadata and data transformation over local data sources.(*Content harmonisation, Metadata restructuring: the categorisation*).

4.3.3 Multi-lateral Ontology Architecture

The overall design of EDEN-IW system follows a conventional 3-tiered information architecture design (Figure 5) consisting of a resource management layer, an

application logic layer and a presentation layer. In a heterogeneous distributed system, such as EDEN-IW, components in each of these layers can be distributed and heterogeneous. In the EDEN-IW system, functions in each of these layers are integrated using a semantic metadata model. This is shared using a multi-agent infrastructure.

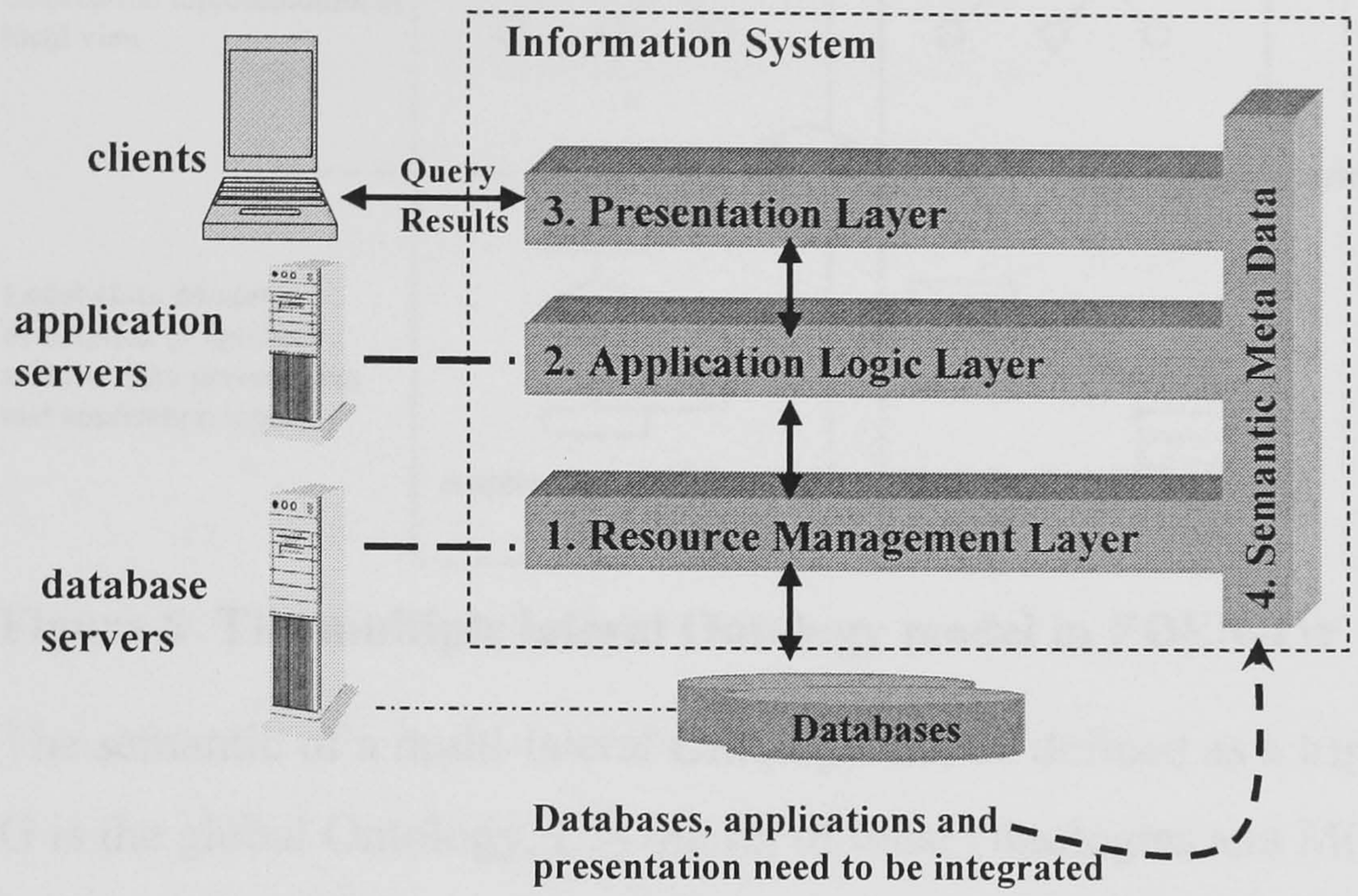

Figure 5 Standard model of an information system.

The semantic metadata model (Ontology) is partitioned into layers with respect to presentation, application logic and resource management - a multi-lateral Ontology. The architecture of the EDEN-IW data model, Figure 6, follows the three schema ANSI/SPARC architecture [91]. It has a lower layer reflecting the local physical representation in the database, a middle conceptual schema and an upper external schema that provides different views of the conceptual schema from the perspective of an application. Note many network and information system models further refine the upper layer application layer into a processing layer and a presentation layer. For example, the same processed information may be presented in French and English. The main advantage of the basic 3 layer partitioning is that it supports semantic autonomy and physical distribution of metadata and local data sources, i.e. it allows additional database models to be added without changing the other layers, providing they do not require changes into the conceptual model. New application uses of the conceptual data model can be added with minimal disruption if they do not introduce new concepts into the global Ontology.

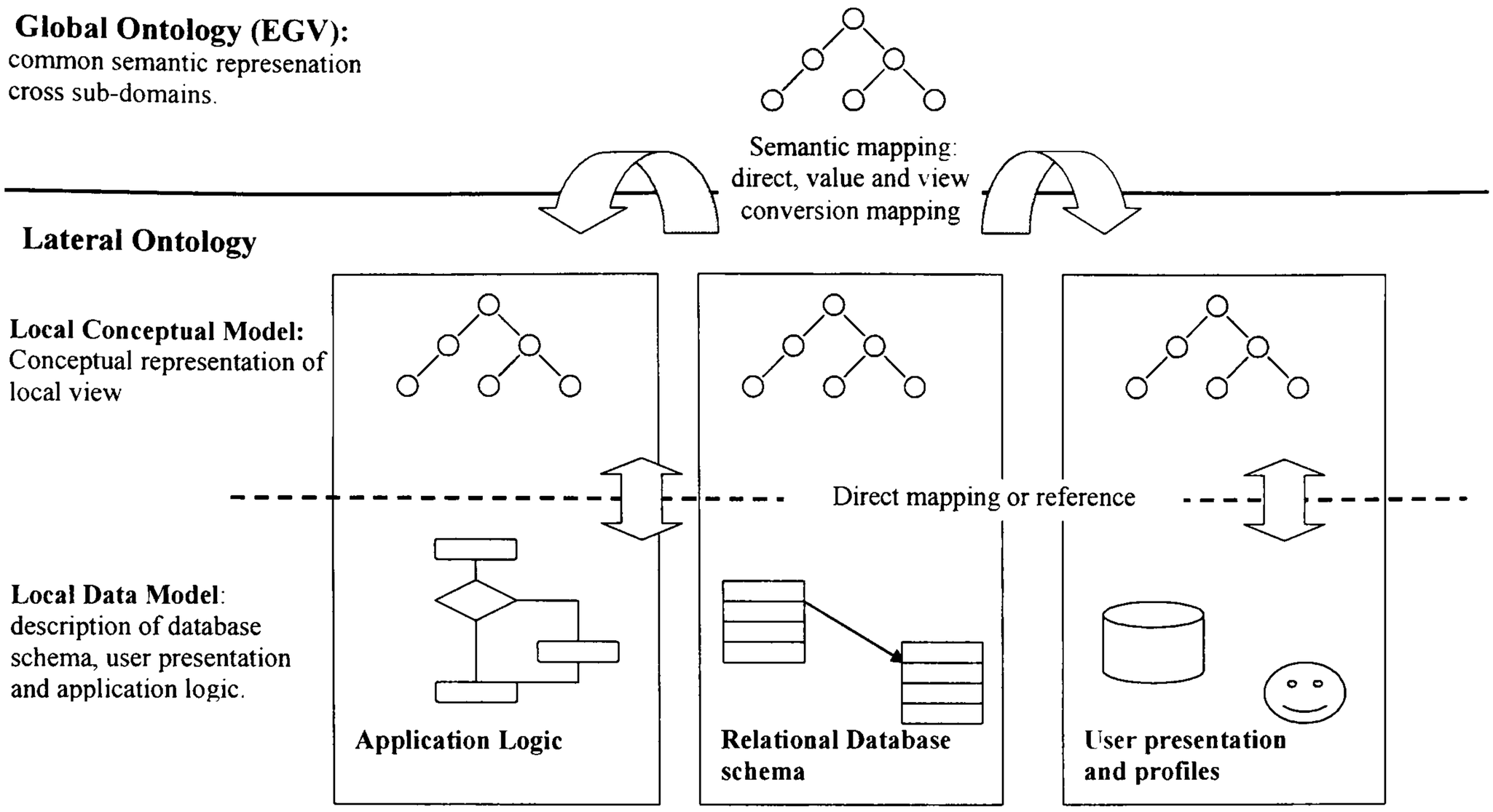

Figure 6 The multiple lateral Ontology model in EDEN-IW

The semantic of a multi-lateral Ontology can be defined as a triple of, <G, L, M(G,L)>. G is the global Ontology, L is the set of local Ontologies and M(G,L) is the mapping between G and L.

- *Global Ontology G*, represents the common conceptual representation of knowledge in inland water domain
- *Local lateral Ontology L,* expresses the conceptual structure of local data model or application logic. The local Ontology set of L consists of local Ontologies {L1, L2,…Ln}
- *Mapping between G and L, M(G,L),* the mapping allows the knowledge transformation in two dimensions: the conversion of metadata and their corresponding data extents.

An Ontology mapping specifies inclusive relation and functional dependencies between global and local Ontology conceptualisations. Regarding information retrieval over integrated systems, mapping of data and metadata plays a crucial role to resolve interoperability problems between global and local conceptualisations. The queries and results formed in either representation can be transformed and processed into the other one via Ontology mappings. The mapping is defined as a set of enumerated rules identifying semantic equivalences throughout multi-lateral Ontologies. The mapping includes metadata mapping for Ontology terminologies and value mapping for the corresponding data instances. The mapping is defined in formalised syntax of Ontology representation that can be reused by different applications.

4.3.4 Global View Ontology

The global conceptual model, called the EDEN-IW global view Ontology or EGV model represents the common understanding of domain conceptualisation that is independent from any local database and other application. The EGV Ontology model serves several purposes:

- It provides a common data dictionary - definitions, concept names and enumeration's of e.g. determinands and units.
- It provides the basic classes for conceptualising the intensional data in local databases.
- It provides a schema of the required information for each concept, e.g. an "observation" requires more than just a value and a unit to describe the type and context of the observation.
- It provides an organisation for the common knowledge including class relationships and other relationships.
- It provides a virtual integrated data schema that supports information queries to all local data sources.
- It provides a semantic-based transformation path between different types of data and metadata categories.

In order to encompass a variety of local database implementations exemplified in table 1, the EGV is to a large extent, made up of "primitive" classes. The EGV include classes that are specific to the Inland water domain, as well as more universal classes suited for describing database schemas and elements like Time and Units. The classes are organised in hierarchies, with EdenGlobalConcept as a super-class. The EGV also contains relevant instances of the defined classes.

For example in use case 1, the Inland water databases contain information of the type "the VALUE of DETERMINAND observed at a STATION at a TIME". A deeper analysis of the concept of "the VALUE of DETERMINAND" in a couple of databases has identified that the value of a determinand may actually express different types of information:

- Instantaneous values vs. time-aggregated values.
- The same determinand observed in different media and fractions.
- Values may be expressed with different units e.g. milligram/litre or nanogram/litre.

- Values may be expressed in different chemical compounds, concentration of Nitrate may e.g. be expressed either in milligram N per litre or in milligram NO_3 per litre.

Hence, this has led to a model of global class relations for determinands that supports these design requirements, see Figure 7 .

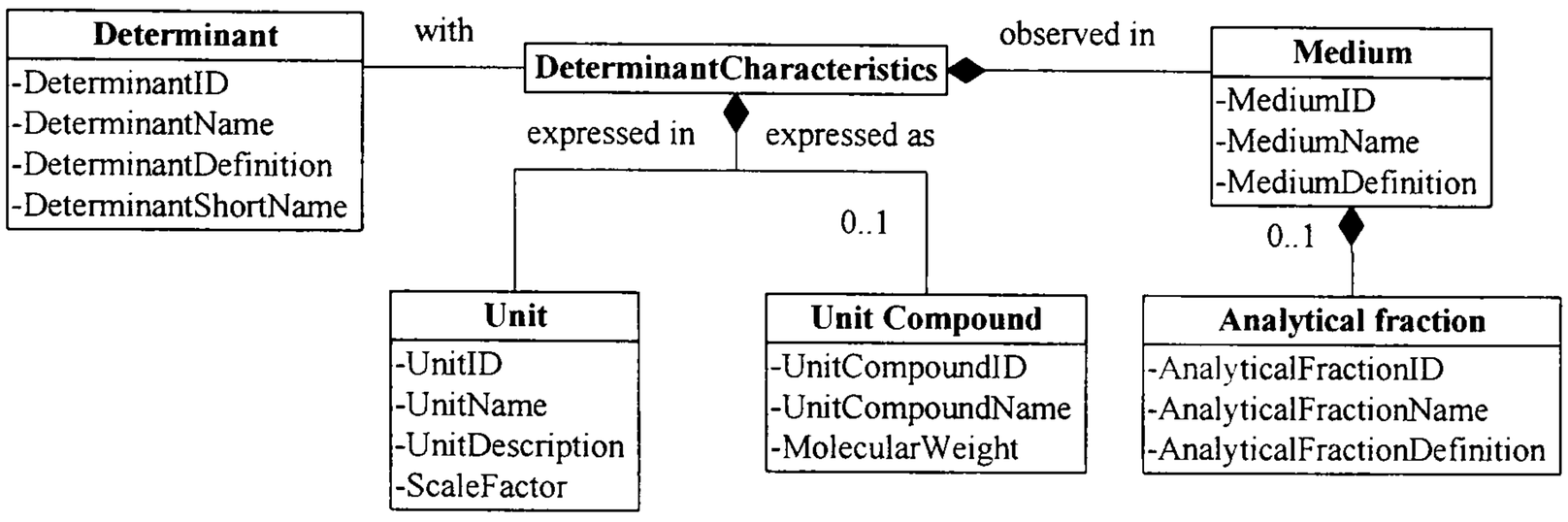

Figure 7 EGV representation of determinands and associated classes

4.3.4.1 Class vs. Instance Modelling Issues

The use of any representation of a data model necessitates conforming to restrictions of expressivity of that particular data representation. For example, the Web Ontology Language (OWL), representation only supports limited relationship expression between instances, for example owl:differentFrom and owl:sameAs. User-defined instance relationships are not allowed in OWL syntax, which makes the expression of instance relations difficult in practice. SQL supports different ways (redundancy) to express a join between relational data tables. Another type of modelling choice is which type of domain class relationships to represent and whether or not to represent concepts into set or has-a relationships or to represent the same concepts instead in class inheritance or is-a relationships.

When application users and application domain experts start to develop a domain model, this is often approached by examining instances of classes and relationships between instances, i.e., the concrete data rather than the abstract data. There may be a desire to capture relationships between instances rather than to view this more abstractly as relationships and constraints on classes. We can capture some instance constraints in terms of specifying classes whose properties have certain values. e.g., the determinand "Discharge" can only be observed in medium "Water".

```
<owl:Class rdf:ID="Nitrate">
 <rdfs:label xml:lang="en">nitrate</rdfs:label>
 <owl:disjointWith rdf:resource="#Nitrite"/>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#Nitrogens_oxided"/>
 </rdfs:subClassOf>
 <owl:equivalentClass>
  <owl:Restriction>
   <owl:hasValue>
    <Determinand rdf:ID="nitrate">
     <DeterminandID>19</DeterminandID>
     <DeterminandDef>Nitrogen in the form of NO3- </DeterminandDef>
     <EDENTermID>1014</EDENTermID>
     <ChemicalFormula>NO3-</ChemicalFormula>
     <DeterminandName>nitrate</DeterminandName>
    </Determinand>
   </owl:hasValue>
   <owl:onProperty rdf:resource="#hasIdeterminand"/>
  </owl:Restriction>
 </owl:equivalentClass>
</owl:Class>
```

Figure 8 Determinand list modelling in inheritence relation

The statement above can be expressed in an Ontology with two distinct understandings: inherence or subset, according to the specific design purpose of domain application. In the inheritance case, e.g., “Nitrate” and “Nitrite” can be abstracted as the disjoined subclasses of “Nitrogens_Oxided”. Semantically, the inheritance hierarchy implies that a class can inherit all properties from its super-class, i.e. “nitrate” is a “determinandList”, although it leads confusion because “nitrite” and “nitrate” are instances of determinand. The redundancy definition can benefit from a further definition of determinand collection at a lower granularity level, e.g. nitrite can be defined as a collection of varied compounds. An example of OWL representation is for a fragment of the IW concept model is shown in Figure 8.

In Figure 8, the class “Determinands” has an instance “nitrate” with a set of properties (formula, definition etc). “Nitrate” is a subclass of “Nitrogens_Oxided” and is defined by the property “hasIdeterminand” having exactly the value of the in-stance “nitrate”. In the alternative understanding of the subset case, the “nitrite” and “nitrate” can be simply defined as an instance of “Determinand”, while “Nitrogens_Oxided” is de-fined

exactly as an enumeration value class consisting of "Nitrite" and "Nitrate", which is shown in Figure 9 below.

```
<owl:Class rdf:ID="Nitrogens_oxided">
<rdfs:subClassOf rdf:resource="#DeterminandList"/>
<owl:equivalentClass>
<owl:Class>
<owl:oneOf rdf:parseType="Collection">
<Determinand rdf:ID="nitrate">
<ChemicalFormula rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>NO3</ChemicalFormula>
<DeterminandName xml:lang="en">nitrate</DeterminandName>
<owl:differentFrom>
<Determinand rdf:ID="nitrite">
<DeterminandName xml:lang="en">nitrite</DeterminandName>
<ChemicalFormula xml:lang="en">NO2</ChemicalFormula>
<DeterminandID rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>18</DeterminandID>
<owl:differentFrom rdf:resource="#nitrate"/>
</Determinand>
</owl:differentFrom>
<DeterminandID rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>19</DeterminandID>
</Determinand>
<Determinand rdf:about="#nitrite"/>
</owl:oneOf>
</owl:Class>
</owl:equivalentClass>
```

Figure 9 Determinand list modelling using the subset relation

Both models in Figure 8 and Figure 9 are correct in the sense of OWL syntax. They represent variations in the interpretation of the domain knowledge from different view-points. The modelling of the domain Ontology is not a straightforward process leading to a single monopoly result. The representation of a domain Ontology model may vary depending on several factors including the expressivity of the Ontology language, the scope of the domain, requirements, the application commitments and the Ontology development process.

4.3.4.2 Ontology Harmonisation: Unit Ontology

Data from multiple data sources often cannot easily be compared because the data represents different values, for example, differences in whether or not the measurement

system, has been calibrated recently or the data has been averaged differently. Additionally, metadata to record the provenance of the data from the measurement source, the characteristics of the measurement technique and tags to indicate any post-processing of the measurement data, are needed. These are needed in order to do a true comparison, e.g. a unit conversion that may be needed to equate measurement data in different units. Generally, many semantic models are not expressive enough to support general data transformation rules and rule-based processing of the semantic data.

One problem with units conversion is that it is cumbersome to define conversion factors for all the possible combinations. The solution is to define a set of basic unit classes (weight, length, time etc.) with instances in the EGV model. For each instance, the scaling factors (offset and scale) has been defined relative to the basic unit. More core complex units are defined using the basic unit classes. A "FluidConcentration" unit is a subclass of "ConcentrationUnits" and is defined by having a numerator from the "WeightUnits" and a divisor from the "VolumeUnits". Different unit instances may now be compared according to the class types. "ConcentrationUnits" is a subclass of "FractionUnits" that are specified to have both a numerator and a divisor.

A Comparison of different instances of "ConcentrationUnits" may now be applied using a general rule applicable for all "FractionUnits" and using the scaling factors for both numerator and divisor.

4.3.5 Local Database View Ontology

The local database view Ontology (LDV) wraps the local database content. The aim of the LDV is to reformulate database schema to fit conceptual representation and semantic relationships in an integrated Ontology model and hence corresponding elements between EGV and LDV that can bind successfully. The LDV model consists of database schema and local conceptual models that contain the created concepts from primary EGV concepts. The conceptual model contains a semantic representation of the underlying knowledge in explicit descriptions. The intermediate semantic relations between EGV and local database schema are classified into types including syntactic, model and semantic relations, see section 4.3.7. LDV is defined in generic rules to ensure the reusability of the LDV model and Ontology service. In the prototype system, the relationships of "inheritance", "equivalence", "aggregation" and "functional dependency" have been specified. Each LDV element is defined as a view over EGV

primary concepts. The database schema is represented in LDV model with all remained key concepts and constraints, for example "Relation", "Attribute ", "PrimaryKey" and "ForeignKey". Each relation in a local database is described in terms of a subclass of the common super-class concept "Table". Each attribute is described as a sub-property of the common super-property "field". A primary key and foreign key could be defined as the particular object property in the table class, whereas each key relation may contain one or multiple properties in the table that it belongs to.

4.3.6 Application Ontology

Ontologies are central to the semantic function of EDEN-IW because they allow applications to agree on the terms that they need to interoperate. These terms cover the logic concepts and relations. The combination of concepts and relations indicates the precise semantic meaning of the application communication.

Ontology services allow applications to load and parse the EDEN-IW Ontology models, in order to support querying and retrieving of the local database data. The Ontology services are implemented as Java applications that were developed using Jena [4], a Java framework for building Semantic Web applications developed by HP. The version of Jena was used (in 2005) provides a programmatic environment for RDF, RDFS and OWL, including a rule-based inference engine. At the start of the project, the focus was on DAML+OIL, supported in an older version of Jena, as this was the most mature semantic model. As the project progressed, support for OWL became more mature.

4.3.6.1 Query transparency

The main application described in this chapter is to make and answer core user queries to local IW data with respect to determinand, station (location) and time constraints. Other applications have been investigated as part of the EDEN-IW project such as DSS (Decision Support System) queries but these are not covered here. User queries are specified at a Web based user interface using terms from the EGV Ontology. They are then translated into service action invocations expressed using RDF, the Resource Description Framework. There is one RDF service invocation defined for each type of core user query. RDF was chosen as the query representation rather than DAML+OIL because at the time, Jena recommended the use of RDF to represent instance data. The

service invocation then gets mapped into local data resource concepts and from these, SQL statements get generated to retrieve the data from the data sources. This process is described in more detail in later parts of this section. The user does not need to know the location of the database nor need to be able to express a data query in SQL.

4.3.7 Semantic Mapping of Metadata to Data

The semantic mapping is represented in a form of views interlinking semantic correspondences for query request and query result represented in different representations according to their semantic interpretation over knowledge domain.

- Ontology mapping can be used to enable Ontology alignment, Ontology integration, information retrieval, and to support Web service, and e-commerce application interoperability. Ontology mapping could provide a mediation layer from which multiple Ontologies could be accessed and hence, exchange information in a semantically sound manner[51]. The development of semantic mappings over multi-lateral Ontologies involves syntactical and semantic transformations. A concept or constraint relation in one Ontology may correspond to a view (i.e. a query) over the other Ontology. The multi-lateral Ontology model can be developed in two approaches as defined in [27], denoted as a global-centric approach and a local-centric approach.
- A *Local-centric approach* is where concepts from the local Ontologies in L are mapped to queries over the global Ontology G.
- A *Global-centric approach* is where the concepts of global Ontology G are mapped into queries over the local Ontologies in L. The local-centric approach has a scalability advantage over the global-centric approach, i.e. the local data and metadata update can be conducted via modifications of local view mapping into global concepts and relations.

The local-centric approach has been adopted in EDEN-IW to keep data autonomy in the local data sources. The details of the method of development are discussed in section 4.3.8.4. EDEN-IW multi-lateral Ontology has adopted the *Local-centric approach*, i.e. each concept in local view is considered as a view over EGV[26]. A single LDV concept can be represented as a sequence of one or many EGV entities having an equivalent semantic meaning. The semantic mapping here mainly focuses on solving semantic heterogeneity and representation heterogeneity issues in legacy databases.

The mapping is specified as the enumerated functions to map the corresponding entities and relations, describing the equivalent information sets across global and local Ontologies. The semantic equivalence between queries is validated by domain experts. Mapping relationships need to be constructed between the EGV and LDV views. The semantic mapping falls into three categories:

- *Direct mapping from database column*: The direct mapping is applied for the condition that an EGV property has a direct synonym column in the database schema, no additional logic or value conversion is needed.
- *Value mapping*: Value mapping is applied when an EGV property has the same semantic meaning as a LDV property, but the terms mapping could not be established due to the problem of different coding format and value representation between EGV and LDV terms. In this case, an interim concept is introduced to map the EGV concept and provides a value mapping or conversion. For example, due to the name coding difference between EGV determinand and IOW determinand, an interim concept "IOWDeterminand" is created in IOW LDV and mapped to "Determinand" in EGV.
- *View conversion*: The water quality data in local data source is formed as the product of specific processing programs in the local domain. It can be represented as a logic view over EGV consisting of a designated sequence of EGV concepts and relations. For example, in the NERI LDV, a local logic concept NERIObservationCharacteristic is created to represent the observation meaning in EGV as "*Determinand* X measured with *Analytical Fraction Y* in *Medium* Z , expressed in *Unit* U".

From the Cardinality view, the direct mapping and value mapping are marked as one-to-one mapping between two entities, and then view conversion mapping involves more complex relations of one-to-many mappings.

Although, automatic generation of the local view Ontology derived from the database system file, is seen as a beneficial, such a goal is difficult to achieve. The process of performing the first mapping a database to the EGV will always have to include knowledge experts who know about the database structure and the concepts behind it. A simple element like text field string labels does not necessarily contain a term from a natural language, and even if it did the interpretation of the concept would still have to be verified. The development of semantic mappings is conducted using the process described below.

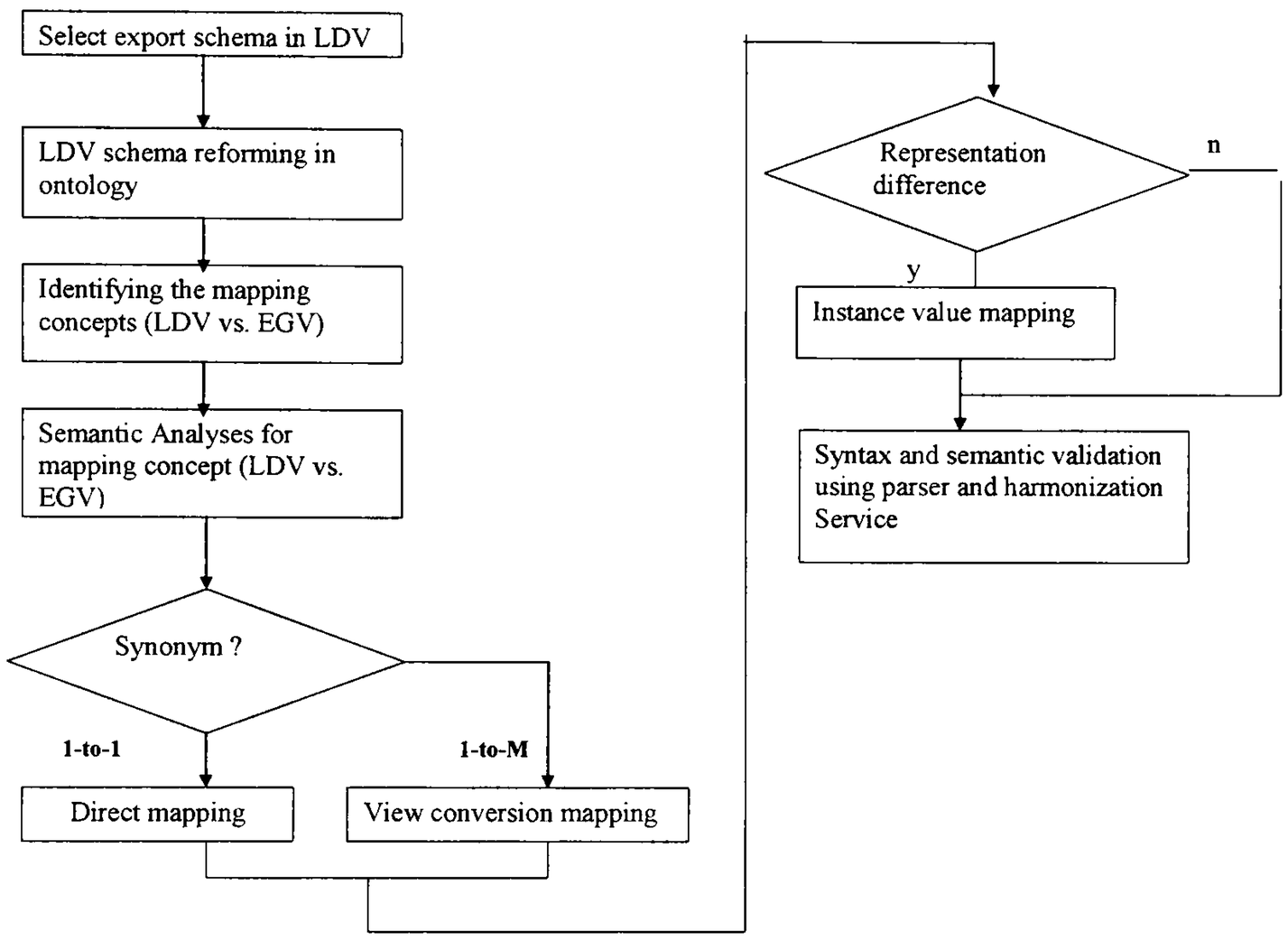

Figure 10 Mapping process for relating local to global Ontology concepts

The mapping process (Figure 10) for relating local to global terms to be able to use queries expressed in global terms to access local terms, is defined as follows.

- Selecting a part of the local schema for export to the semantic mapping. This schema is expressed in an Ontology format (OWL/DAML+OIL).
- Any concepts and properties that have equivalent meaning across a LDV and the EGV are identified.
- A semantic analysis is conducted to determine the mapping relationships.
- One-to-one semantic mappings are marked as direct mappings.
- One-to-many semantic mappings are marked as view conversion mapping where a LDV concept is a view representation of a collection of EGV concepts linked by particular EGV relationships. An intermediate concept is created in the LDV.

- The value coding format across EGV and LDV for those mapping concepts is compared. If their formats are different, a corresponding instance value mapping is defined in the Ontology.
- The syntax and semantics are validated.

The semantic mapping across the global and local Ontology views supports the query transformation between EGV and LDV by giving explicit mapping descriptions of terms, views and instance values. SQL queries to the local database can be generated based upon the LDV terms. The SQL query results returned from database are harmonised into EGV expressions in order to be presented to the user.

4.3.7.1 Terms Translation

Terms translation handles the concept and property translation to local database columns in order to build up the SQL query. The terms translation is executed as the metadata query such as "Which column in NERI domain has the equivalent meaning of EGV term 'determinand'?" to translate the concepts between EGV and LDV. The search for the column name can be executed as to find the concept X satisfying with following criteria:

- X is a concept in NERI local database view
- X is a column name
- X has equivalent mapping (terms/view mapping relation) to core Ontology concept "determinand".

The translation process starts from the search for terms mapping, if no satisfying concept and property can be found, further searches using view mapping relation will be conducted.

4.3.7.2 Value Coding Translation

The data instance in local database may be represented in a different value-coding format, for example Nitrate is coded in EGV with ID 19, in IOW database it is code with ID 1340, in NERI Nitrate it is related to determinand ID 308. The value coding instantiation in LDV defines the value mapping that is specified in the RDF file. Using the name space, the RDF Ontology can refer to the global or local Ontology for its conceptual interpretation.

The Ontology parsing and inference service uses global or local identification to check their value mapping definition in the RDF Ontology. If no value mapping can be found,

which means the global representation is the same as the local representation. For example, some river names can be used directly in databases.

A special action is taken for the common value translation such as the knowledge representation of time and unit. The translation service browses the self-explanation structure of the sub-domain Ontology in EGV and deduces the conversion functions.

4.3.7.3 Determining Join Paths

Most user queries do not clearly specify the table join relations in the SQL statement. In order to generate the SQL query with a corresponding semantic meaning, the query translation between EGV and LDV Ontologies requires identifying the correct join relations in the relevant table. The join relation can be uncertain due to the multiple potential relations paths found in the local database schema. The potential path set can be identified by applying graph theory into the translation process. Choosing an incorrect join path will affect the query results as information loss may occur during the join process. Ideally, the relation mapping between LDV and EGV can be hard-coded in OWL/DAML syntax. However the approach is not scalable and the mapping process is less quality controlled because the difficulty in reasoning about the equivalent mappings.

4.3.8 Ontology Development and Maintenance Issues

At the start of the EDEN-IW project, in 2001, few relevant IW domain Ontologies existed. The concepts in the EGV were related to the collected data in databases and derived from discussion with domain experts. An analysis of the domain of "Inland Water quality" has shown that similar terms are used in the description of monitoring programs and observations. Deeper analysis has also shown that the understanding and implementation of the same concepts does differ in crucial areas and can lead to misconceptions if they are not handled in a strict way.

There are well-defined processes and methodologies for Ontology development such as [36],[73]. Processes differ depending on whether or not the Ontology is developed from scratch, the Ontologies are cooperatively constructed or the Ontologies are re-engineered from existing Ontologies[36].

EDEN-IW focussed on building an Inland Water Ontology from scratch. Ontology development environments that include visual tools for graphically creating, editing Ontologies and then exporting representations for on-line use by application processes

facilitate Ontology development. Because of the requirement of EDEN-IW to focus on XML type Ontologies, EDEN-IW chose DAML+OIL as the Ontology language and later shifted to OWL. Development tools that support DAML+OIL developments include OILEd [17] and Protégé. Of these Protégé [8] was considered to be the most mature. Newer versions of Protégé no longer support DAML+OIL, but have shifted to support OWL.

4.3.8.1 Ontology Creation

When developing Ontologies with Protégé, Noy et al [73] outlines a process for engineering Ontologies that consists of: determining the scope of the Ontology; considering Ontology reuse; enumerating important terms; defining classes and the class hierarchy; defining properties of classes; defining constraints and creating concept instances.

The whole development process may be described as an iterative process of refinement of the Ontology through exchange of domain knowledge between the domain experts for inland water and the Ontology, agent infrastructure, developers. EDEN-IW used a combination of bottom-up and top-down methodology. The bottom-up approach starts from the underlying data sources to generalise the common concept and relations in the knowledge domain. The top-down approach initialises from the analysis of domain knowledge to identify the key concepts and relations. The bottom-up approach is employed during the development of the local Ontology model and the top-down approach is used to create the global Ontology model. In more detail, this is as follows.

- Determine the Scope of the Ontology: The scope of the ontology is for inland water including lakes and rivers. Seas and oceanic water measurements were considered to be out of scope although the scope of the IW ontology could be expanded at a future stage to include these.
- Consider reuse: It can be more effective to reuse an existing domain ontology rather than to construct one from scratch. At the start of the project an XML-based Ontology for the IW domain was not available, so the experts in the project created one.
- Enumerate Important Terms: Define the concepts that are needed their properties and what to say about the terms, e.g., water medium and measured chemical parameter.

- Define Classes and the Class Hierarchy: Associate concepts with classes (collection of concepts with similar properties), e.g., Determinand, Medium and Unit. Define a taxonomic hierarchy to relate classes of related sub-types and super-types, e.g., the super-class is the EDENGlobalConcept and Determinand, Medium and Unit are sub-types of this.
- Define Properties of Classes: Describe attributes of instances of the class and the relations to other instances, e.g., the Medium concept class has attributes of Name, ID and Definition. Simple properties (attributes) contain primitive values, e.g., ID (strings, numbers), but more complex properties may link to other classes.
- Define constraints: Property constraints (facets) describe or limit the set of possible values for properties, e.g., an ID property is defined as a unique Integer Identifier.
- Create instances: For example, Aluminium is an instance of the Determinand class.

Noy is concerned only with the process that addresses aspects of introducing a new Knowledge Management) KM solution into an enterprise, the so-called "Knowledge Meta Process". Her KM is not concerned with the process addresses the handling of the already set-up KM solution, the so-called "Knowledge Process", in this case it does not describe the use of the Ontology to support database integration. This may require an update to he support new conceptualisations in order to align it with the conceptualisation of a new database. At a high-level, this just causes another iteration through the Ontology creation process in order to create or modify the existing conceptualisation

Note that this creation process does not consider the use of the metadata to operate on other different data representations, e.g., to better search the data instances not in the metadata representation. If this is the case then additional steps are needed to map or relate the metadata instances to specific (database) source instances.

An important aspect in defining the classes and the class hierarchy is to be precise about the actual relationship between the different concepts rather than simply importing e.g. a logical data model from e.g. a database, however common that may be. To illustrate this here is an example. When analysing environmental observations from a river station, a number of characteristics that are related to the station will be of interest. These may comprise of the size of the catchment contributing to the flow of

water passing the station and the population living in the catchment. In practical implementations of inland water databases these characteristics may often be gathered and stored with the reference to the station ID. An implementation of the "CatchmentArea" and "Population" as properties of the Station class would conceptually not be correct, and would most properly at later evolution stages lead to a complete revision of the class hierarchy.

The more appropriate approach would be to link the station to a position on a river stretch. Such a point will have an associated catchment. The catchment being a surface will have its area as a natural property. The population or population density is then an observation linked to a spatial object which represents a surface or a volume. There are two key challenges in using Ontologies once they are created: how to maintain the Ontology and how to orientate the Ontology to different sets of applications, and different sets of types of users. Each of these challenges is discussed in turn.

4.3.8.2 Ontology Evolution

It may be supposed that a domain Ontology model should be created and iteratively edited until it is complete and expressive and competent. Only then is it fixed as a knowledge interface for subsequently use by all users and applications. This is seldom the case in practice - Ontologies are likely to evolve. Ontology evolution can be defined as the timely adaptation of the Ontology as well as the consistent propagation of changes. This variety of causes and consequences of the Ontology changes is discussed in[55], Ontologies are living and have a maintenance phase where parts may change. The main sources of change are [73]:

- *Structure-driven change discovery*: Exploits a set of heuristics to improve an Ontology based on the analysis of the structure of the Ontology. For example, if all sub-concepts have the same property, the property may be moved to the parent concept;
- *Data-driven change discovery*: Detects the changes that are induced through the analysis of existing instances. For example, if no instance of a concept C uses any of the properties defined for C, we can make an assumption that C is not necessary;
- *Usage-driven change discovery*: Takes into account the usage of the Ontology in the knowledge management system based on the analysis of the users'

behaviour in two phases of a knowledge management cycle: analysing the quality of annotations, and analysing users' queries and the responses.

In the EDEN-IW project as it largely focussed on legacy database integration, Ontology maintenance was mainly driven through data-driven change discovery.

4.3.8.3 Ontology Provenance

The EDEN-IW system provides provenance to reference terms specified by high quality International organisations that use a process of refinement and peer review to create and maintain the reference terms. Within EDEN-IW, a light-weight IW domain Ontology representations, e.g. XML, has been created for the multiple International standard thesauri or glossaries of accepted terms such as GEMET, (GEneral Multilingual Environmental Thesaurus), TREKS (Thesaurus-based Reference Environmental Knowledge System) and EARTh and have been used to provide provenance. Each concept in the EDEN-IW common or global IW Ontology is linked to one or more terms in on-line glossaries via identifiers. However, the descriptions of the terms are in free text form in these online glossaries and not in a form to support computation.

4.3.8.4 Developing a Multi-Lateral Ontology for Inland Water

The multi-lateral Ontology in EDEN-IW system contains the EGV Ontology and a set of loose-coupled local Ontologies and application Ontologies as described in Figure 11. The local Ontologies and application Ontologies are physically distributed and autonomously managed by data owners and information users. The semantic mapping between EGV and LDV indicates the semantic corresponds across domain Ontologies. A mapping relation is restricted between one LDV and EGV, so that local metadata and data schema update would not affect other data sources. The EGV is further partitioned into sub-domains according to the intended usage of knowledge. In an example shown in Figure 11, the EGV contains multi-lingual thesaurus, water domain knowledge, unit and spatial information[3] and common database schema concepts. User and application Ontologies contain conceptual representation of user and application concerns that are expressed in semantic representation and mapped to EGV via semantic mappings.

---

[3] The study of conceptualisation of spatial information and relevant transformation process is conducted by other partners of the project. It is not in the scope of this research work.

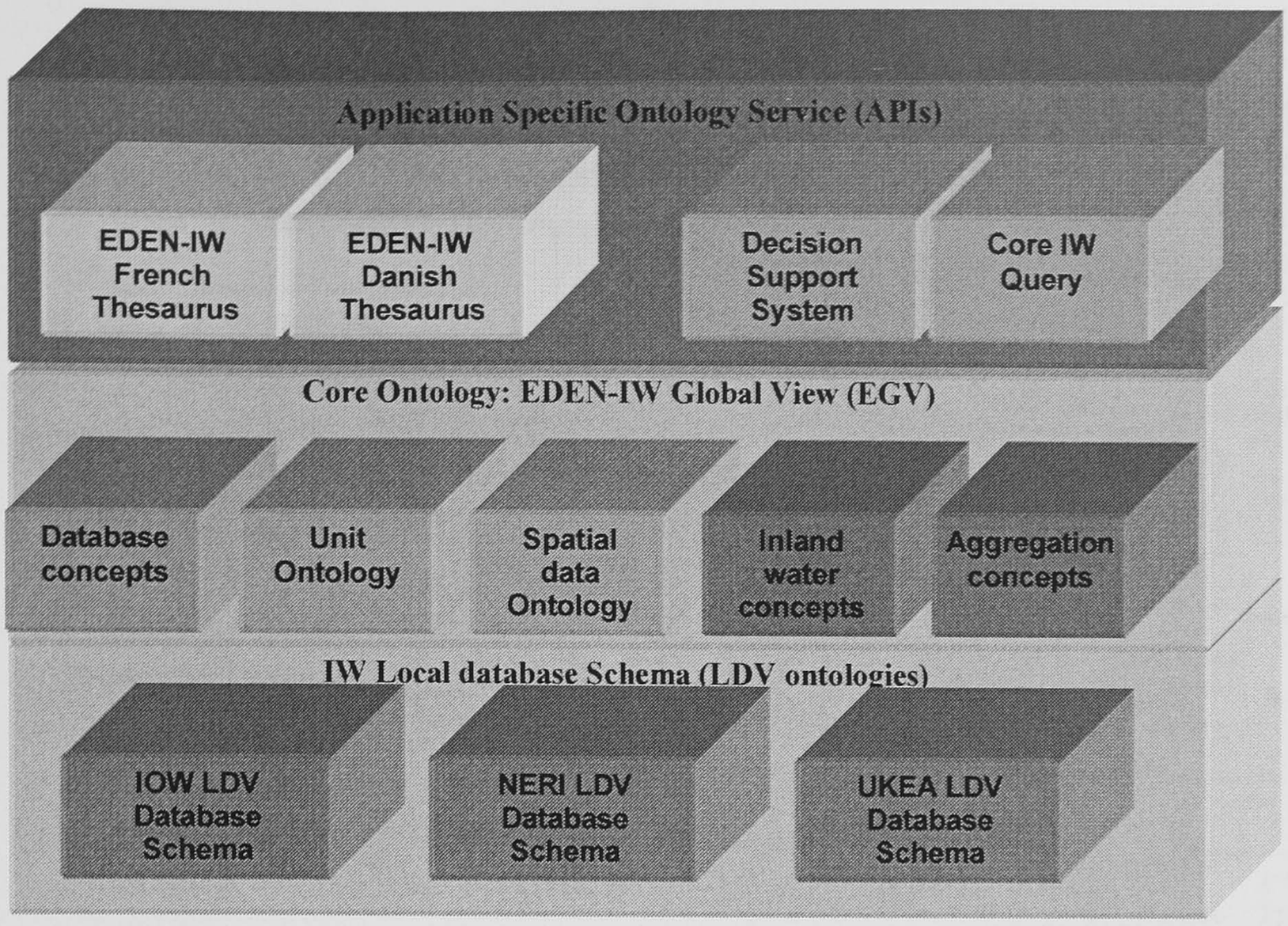

Figure 11 Multi-lateral Ontology in EDEN-IW.

The development of EGV model involved domain experts from the inland water domain. Key concepts and relations were identified to cover common set of domain knowledge, as described in Figure 12. The operation functions upon stored data can be defined for corresponding semantic relation in the Ontology model, e.g. generalisation and aggregation relations. Accordingly, a query request can be split into sub-queries for further estimation in local source.

The Determinand concept is illustrated in Figure 12 giving an example of part of the EGV modelling regarding key scenario of use case 1. Those relevant concepts include all concepts and relations that may be semantically enclosed in a user query context for determinands.

Figure 12 defines the concepts and relations in the determinand sub-domain:

- Each observation has and only has a concentration value for the observed determinand
- Each observation is measured in one and only one medium
- Medium may have one associated analytical fraction
- Each concentration value is expressed in a certain unit according to the determinand name
- Each observation is time stamped specifically
- Each timestamp is an aggregation of date and time.
- Determinand can be grouped into DeterminandGroup

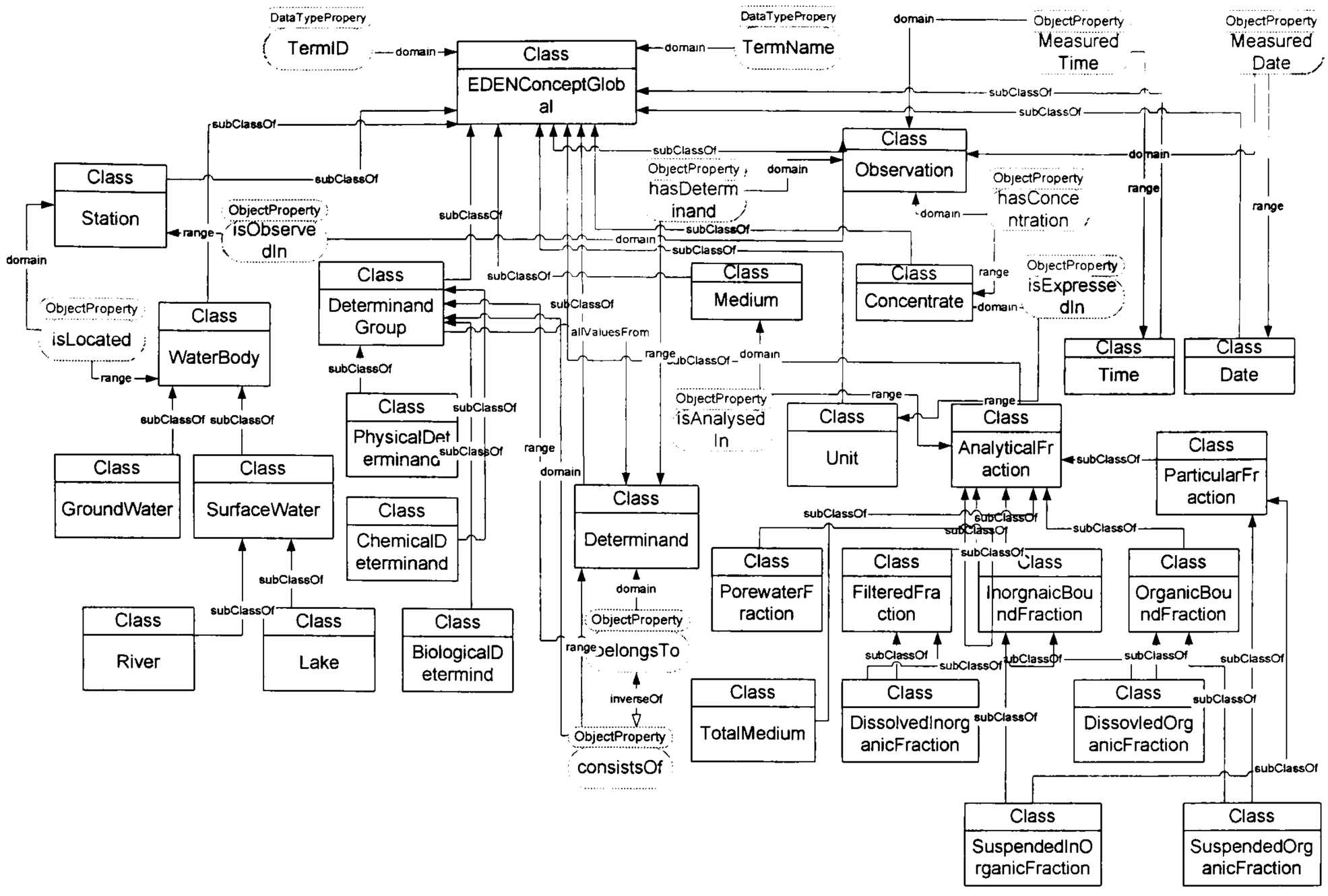

Figure 12 Hierarchy structure of inland water domain (part)

In addition to domain knowledge, common key conceptual entries are also specified for example, EDENConceptGlobal, EDENConceptLocal and EDENDatabase. Those key concepts are defined as the roots in a semantic graph in order to classify different types of information.

The Local Database Ontology provides the metadata information for the local database system consisting of three parts, local database schema, local conceptual model and semantic mapping relations to core Ontology. The LDV conceptual model is defined as an extension of the EGV model using atomic concept in EGV as primary building blocks. The LDV concept and relation are specified as particular query views over the EGV model. The view can be used in query answering process to substitute syntactic and semantic correspondences. The local database schema is an OWL-based representation of database tables and key constrains, including all descriptions about table, column, index, primary key and foreign key relations. The development of LDV follows a semi-automatic process, i.e. database schema can be extracted from legacy databases, whereas domain experts are responsible for importing the database schema into the semantic conceptual model, describing the underlying knowledge, creating intermediate concepts and creating semantic mappings. There are three types of mapping between LDV and EGV, direct mappings, value mappings and view

mappings, see section 4.3.7, used to overcome syntactic and semantic heterogeneities. Mappings are implemented as equivalent classes and properties across Ontologies. The value mapping is implemented as intermediate classes with materialised instance values. The aggregated instance value gives explicit coding format translation for equivalent concepts. View semantics are conducted as a set of enumerated declarative logic rules. The head of a rule indicates the LDV element. . The body corresponds to its view representation over EGV. For example, following rules are defined in NERI LDV as:

View mapping:

$$\forall x, \exists y \mid NERISationLake(x) \Leftrightarrow Station(x) \wedge Lake(y) \wedge isLocatedIn(x, y)$$

$$\forall x, \exists y \mid NERISationRiver(x) \Leftrightarrow Station(x) \wedge River(y) \wedge isLocatedIn(x, y)$$

$$\forall x, \exists i, \exists j, \exists k \mid NERIObservation(x) \Leftrightarrow \det er \min and(i) \wedge medium(j) \wedge analyticalfraction(k) \wedge isMeasuredIn(i, j) \wedge isAnalysedin(j, k)$$

Value Mapping:

$$\forall x, \exists y \mid NERIUnit(x) \Leftrightarrow Unit(y) \wedge equals(y, globalValue(x))$$

By analysis of the semantic relation in EGV, the equivalent queries can be set up across global and local ontology model.

$$Q_{EGV}(o) = obervation(o) \wedge station(p) \wedge \det er \min and(r\ ) \wedge medium(q) \wedge hasDeter \min and(o, r) \wedge hasMedium(o, q) \wedge isAnalysedIn(q, s) \wedge isObservedIn(o, p)$$

For each combination value of variables o,p,q,r,s, there may have corresponding variables x,y,z in LDV, where LDV queries equal to,

$$Q_{LDV1}(y) = NERILakeStation(x) \wedge NERIObservation(Y) \wedge isNERIObserved(x, Y))$$

or $Q_{LDV2} = NERIRiverStation(z) \wedge NERIObservation(Y) \wedge isNERISampled(z, Y))$

where, for any NERIObservation(y), following mapping exists,

$$\forall y_i \in Y, i = 0,...,n \mid NERIObservation(y_i) \Leftrightarrow \det er \min and(r) \wedge medium(q) \wedge analyticalFraction(s) \wedge isAnalysedin(q, s)$$

Mapping a database to the core Ontology through the local database Ontology is performed in a number of steps:

1. The database tables are analysed for concepts that find a direct term mapping counterpart in the core Ontology.

2. For complex local concepts that do not have a term mapping relation to an EGV counterpart, a local conceptual class is defined in order to set up view context mappings.
3. The intermediate class is defined in local conceptual model to accommodate the terms and value translation of class instances in both terms and view context mapping. For example IOW determinand I defined in the IOW local Ontology handles the mapping of local and global determinand names and IDs.

- If a term mapping relation exists between the local and the global concepts. The class will have two key properties, a property related to the local enumeration and a property that related to the core Ontology instances.
- If a term mapping is not possible, e.g., view context mapping, the local class may be defined to include several properties each relating to instances of different classes in the EGV Ontology. The properties will be subproperties of "IsAggregationOf",
- Finally the intermediate classes are instantiated in the RDF file, with corresponding values of the local enumeration and the core Ontology enumeration.

The corresponding LDV models are developed for NERI and IOW respectively, according to the EGV model.

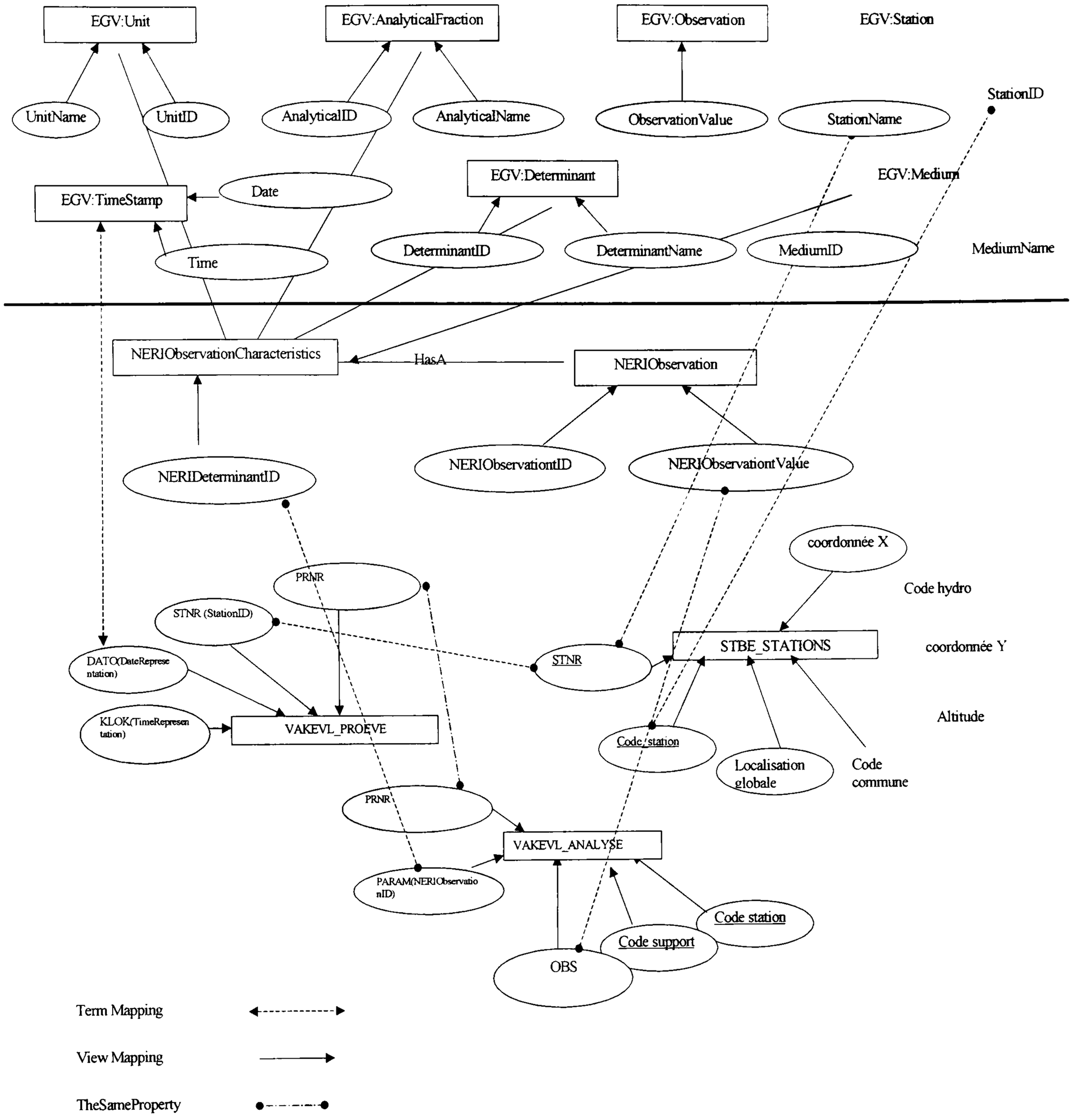

Figure 13 NERI representation of determinand

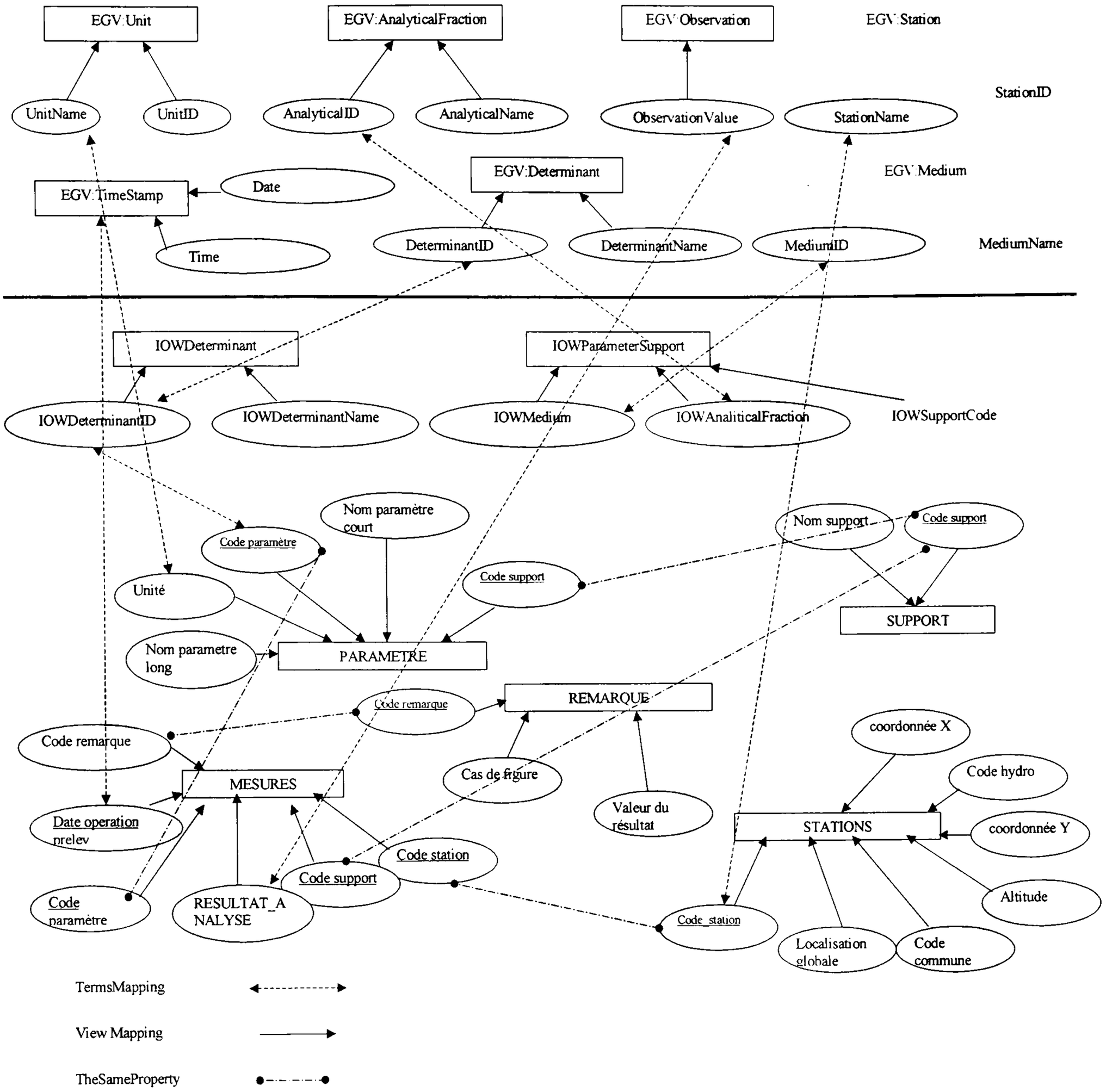

Figure 14 IOW representation of determinand

The local relational database schema has been described in a corresponding conceptual model, i.e. each table is a class containing all column names as properties. Primary key and foreign key could be defined as the particular property in the table class, whereas each key relation may contain one or multiple properties in the current table.

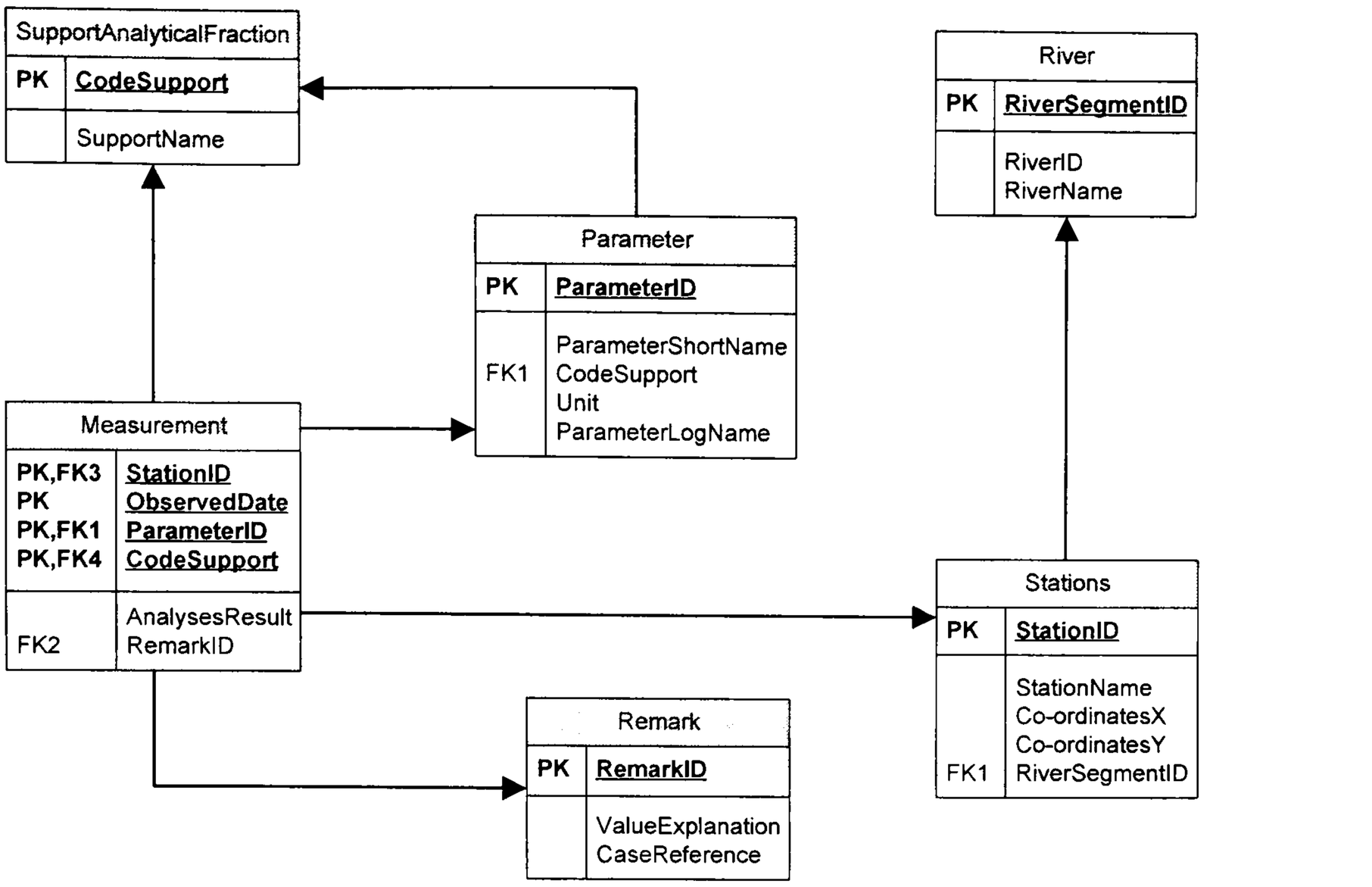

Figure 15 The database schema of IOW database

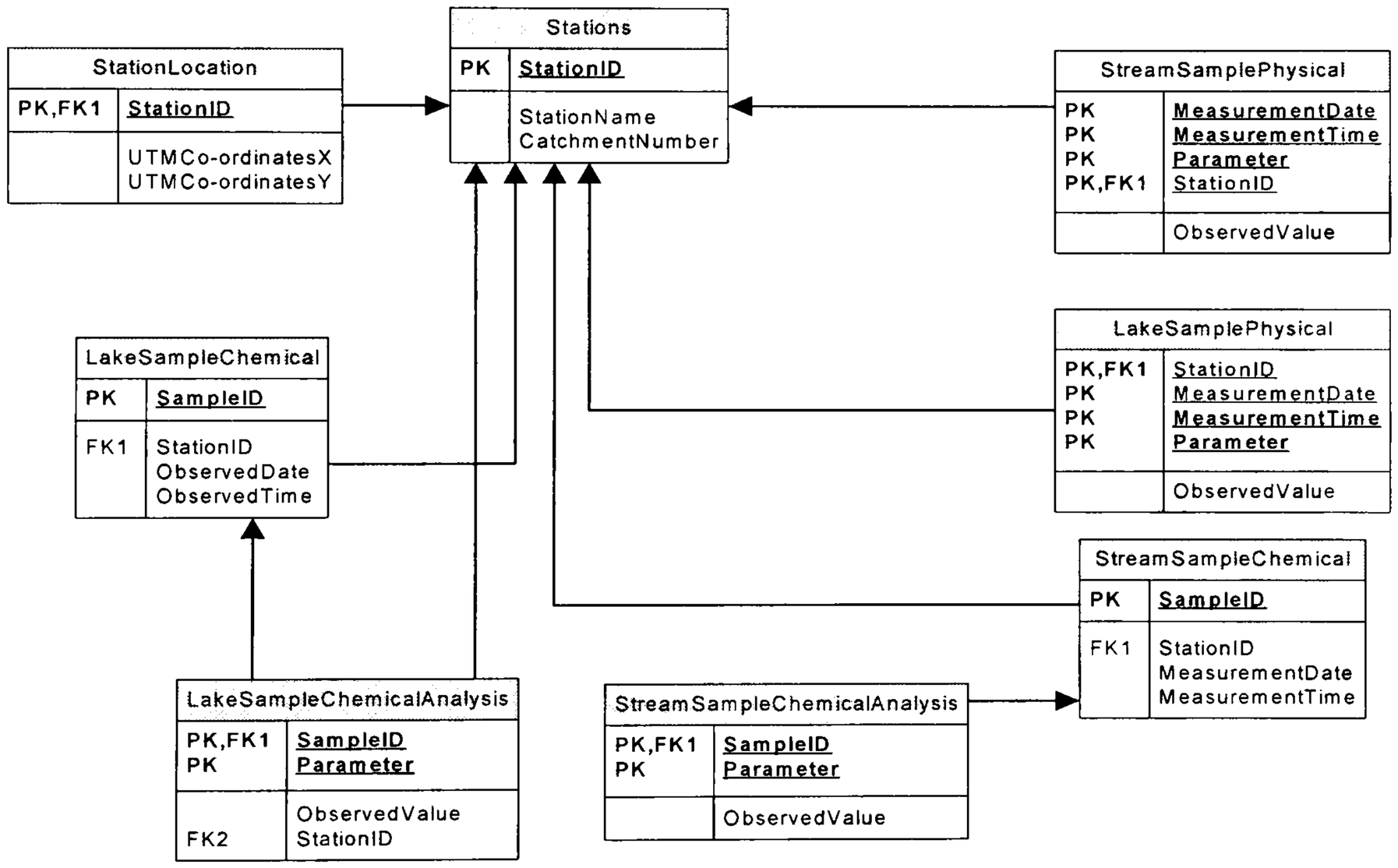

Figure 16 The database Schema of NERI database

Direct terms mapping can be established between the core Ontology model and the local data view to identify the mapping concepts or properties with equivalent meaning in a semantic context, for example the IOWDeterminand can be directly mapped to Determinand in core Ontology model. The direct mapping is tagged in DAML as SameClassAs or SamePropertyAs.

The view context mapping relation deals with more complicated representations of local context that normally can not be directly mapped into the core Ontology concepts. Normally aggregation mapping describes the mapping from a constant query context to a unique concept in the local database Ontology view. For example NERIObservationCharacteristics may represent the context of "Determind X was measured in Medium Y with Analytical Fraction Z and expressed in Unit W". Also the same relation can be used to represent the implied knowledge, for example IOWMedium is defined as the aggregation of medium and analytical fraction, so that the value combination in IOW domain can be explicitly defined.

Table 10 Direct terms mapping for determinand domain

Global Terms	***NERI Interpretation***	***IOW Interpretation***
Station Name	*STAVN*	*Code_Station*
Medium	*No direct mapping*	*Code_Support*
Unit	*No direct mapping*	*Unite*
Determinand	*No direct mapping*	*Code_Parameter*
Date	*Dato*	*Date_Operation*
AnalyticalFraction	*No direct mapping*	*No direct mapping*
Concentration	*Obs*	*Resulta_Analyses*

The concepts and property mapping relations have solved the issue of value mapping and terms translation from core Ontology to the local database view. However, the relation information could be omitted during the context translation because ontological language offers limited support for relations mapping. i.e. only one to one relations are provided in the Ontology language, whereas transformation of query context may involve multiple-to-one mapping , e.g. a local concept is modelled as a context in global model consisting of multiple concepts and relations. A Graph model was adopted to model the problem and calculate the possible answers.

4.3.9 Query Transformation and Metadata Services

A user query posed on the EGV Ontology is able to be translated into LDV expression to access the local databases. The expression transformation for the user query involves a process of mediation and reasoning upon semantic mappings of metadata and data. In the EDEN-IW project, a JADE-based multiple agent system is developed

to support database integration and IR services. The processing of query transformation and metadata reasoning happens in the resource agent, where access to the local database is wrapped in a conceptual schema and appropriate SQL statements are generated in compliance with the semantic expression of EGV queries.

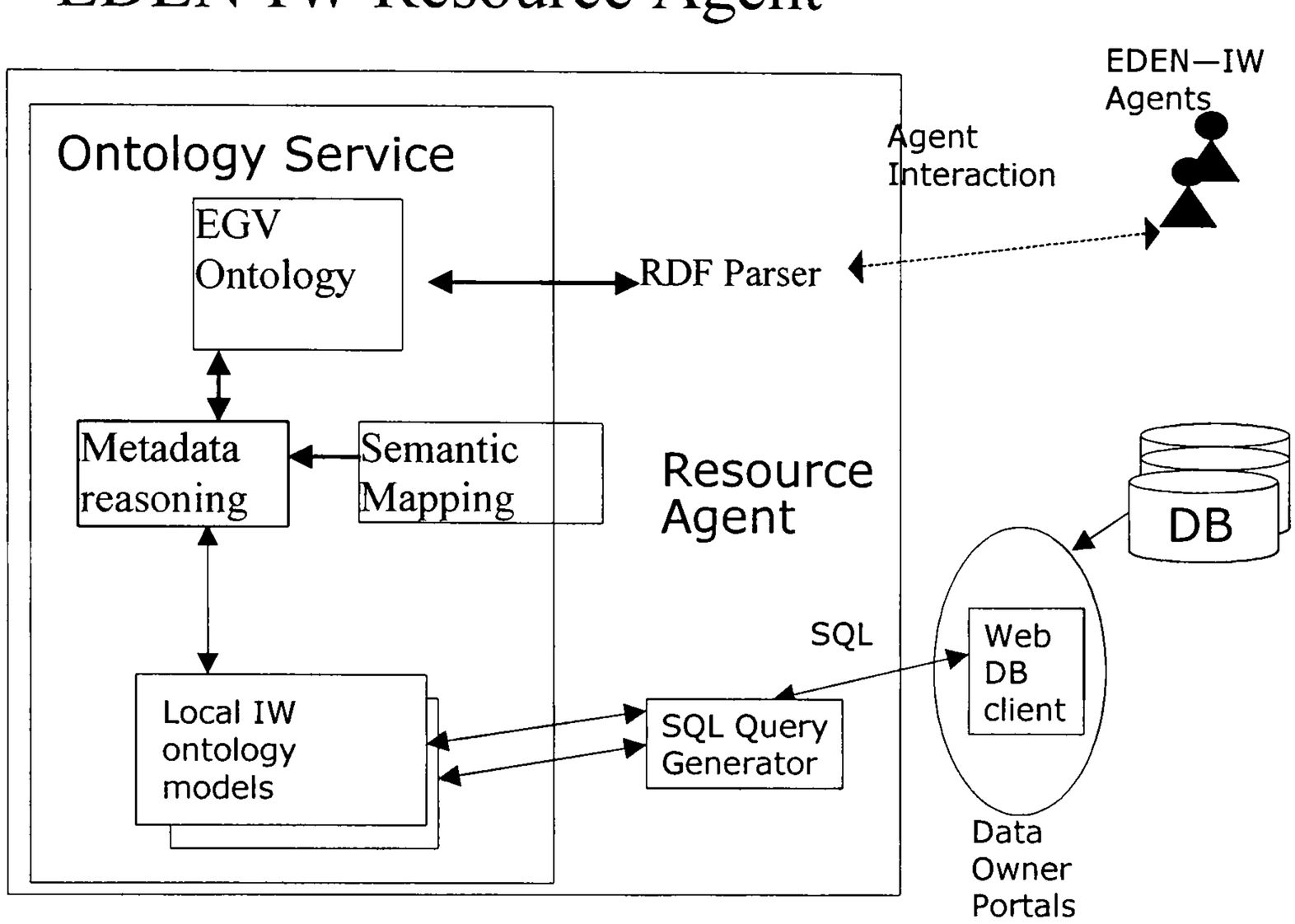

Figure 17 Schematic overview of the database interface / resource agent

As shown in the Figure 17, the Ontology harmonisation service accesses both the local database view and global Ontology view, providing the context translation between them. The resource agent interacts with other agent and application services via a uniform query interface that is represented using the core user query Ontology in RDF format. The RDF query is parsed and loaded into the Ontology service via an RDF Parser, Jena. The user query is translated into local SQL statement with the aid of the local database schema and metadata descriptions in local database Ontology and semantic mapping to global model. The input of the resource agent is the user query expressed in core Ontology terms. The output is the SQL statement for the particular database system. The SQL query generator reads the EGV Ontology and the associated LDV Ontology, maps terms and translates the query statement according to the pre-defined rules. The generated SQL is transformed into the correct syntax according to match the target database type. The SQL query is submitted via an external web portal to the database. The retrieved result is translated back to EGV expression accordingly.

4.3.9.1 Metadata Representation and Metadata Reasoning

The integration of inland water information is a highly dynamic procedure as the system must leverage the plug in of possible new data sources and also be able to accommodate any information updates that occur in local data sources. The knowledge deviation amongst different users and application views need to be defined explicitly in order to automate the information transforming processing between views. The common Ontology model can be mapped and interpreted in terms of the local data sources so that the uniform access and interoperable data services can be enabled.

4.3.9.2 Dealing with Incomplete Mappings

Ontology developers usually have two options to build the core Ontology in a multi-lateral model: either they can specify the core Ontology as an exact union of all local conceptual representation or they describe the global model in general terms and simple plain conceptual model that can be transformed to relate to other conceptual models. The former approach keeps the local understanding intact but involves more development complexity in the core Ontology because the local representation may be inconsistent amongst views. The plug-in or new data source or other local information view might be difficult because some appropriate concepts may be lacking, necessitating the core Ontology to be updated. Also, the former one requires that the information users have exact knowledge about the local domain structure - else they cannot express the correct question. Although the generic query in the latter approach may lose information details useful for local understanding, it is beneficial for the flexible interpretation of heterogeneous local database Ontology models, a crucial scalability factor for open systems. The ontological commitment can be expressed in a certain abstract level so that the upper information processing requirements can be satisfied without loss of too many local representation details.

In the latter case, a user query can be normally expressed into a query of the instance value of a certain class that satisfies the constraints consisting of given relevant instance value and relations. The query expressed in core Ontology terms need to be translated to local representation for data source access and data retrieval. The translation of query context between ontological views is complicated because the Ontology representation may express different abstraction levels. The translation of query context may go beyond the terms mapping approach that have been adopted in

conventional syntactical system, a local concept may represent a view over a set of concepts and relations in another Ontology. A query represented in core Ontology terms can be regarded as another view over the targeted domain knowledge. Some core Ontology query may not be mapped to the local representation view exactly because of the lack of sufficiency to assert the semantic equivalence between two query views. The lack of sufficiency may due to vague or incomplete information, e.g. no corresponding mapping relation is explicitly specified, or the views are too complicated to be compared. The local query representation cannot be simply translated in the case due to a ***vague or incomplete*** Ontology mapping specification.

In the case of vague or incomplete mapping occurs, the complementary information inference functions may be adopted to find a corresponding view in a target Ontology expression with exact or similar meaning for the semantics.

The ideal solution is to find a generic abstraction form to model all the common semantic characteristics of a query context and to further define the relevant algebra operation upon to measure and calculate the similarity. The mathematics approach of Graph Theory is adopted to inference incomplete information and reduce the information loss during context translation. In the case of no exact translation, the graph theory can support query relaxing, i.e. to find the similar query in the targeted Ontology model with relaxed constraints.

4.3.9.3 Graph Theory with Semantic Routing

The processing of information transformation may have to face a mismatched expressivity of conceptual representation across multiple sub-domain Ontologies where a constraint in one data model may not be available in the others' models, e.g. the key relation in database is not understandable in end-user viewpoint according to their semantics. In such case, graph theory can be used to solve the problem by analysing possible routing deviations amongst semantic entities and to select the best matching one for the database schema.

Predicates in an OWL Ontology are expressed as a set of predicate tuples consisting of a subject, an object and a connecting relation. The Ontology model is represented in a connected direct graph. Each object or subject in Ontology is expressed in an individual graph node. Each directed edge indicates the corresponding predicate from subject to object. A valid query should contain a complete sub-graph in the Ontology model. The process to discover the corresponding sub-graph in target Ontology model

is divided into two sub-phases: nodes identification and routing searching. The former one is characterised by the specified mapping that is discussed in section 4.3.7. The latter phase reasons the dynamic links amongst corresponding nodes in order to get the best-matching routes. The query translation across Ontologies requires the generic processing of graph identification with similar semantic meanings in the target Ontology. The semantic transformation uncertainty can be modelled into the determination of matching routing in corresponding semantic graph. In EDEN-IW system, the applying of inappropriate join routes in the generated SQL statement will possibly deviate the semantic meaning of user query and reduce the retrieved result. The weight value of the join relation is assigned for the enumerated user queries. Given an EGV query in the limited knowledge scope, the potential SQL mapping set is deduced via calculation of possible connected sub-graphs in the target LDV and by a comparison of their weight.

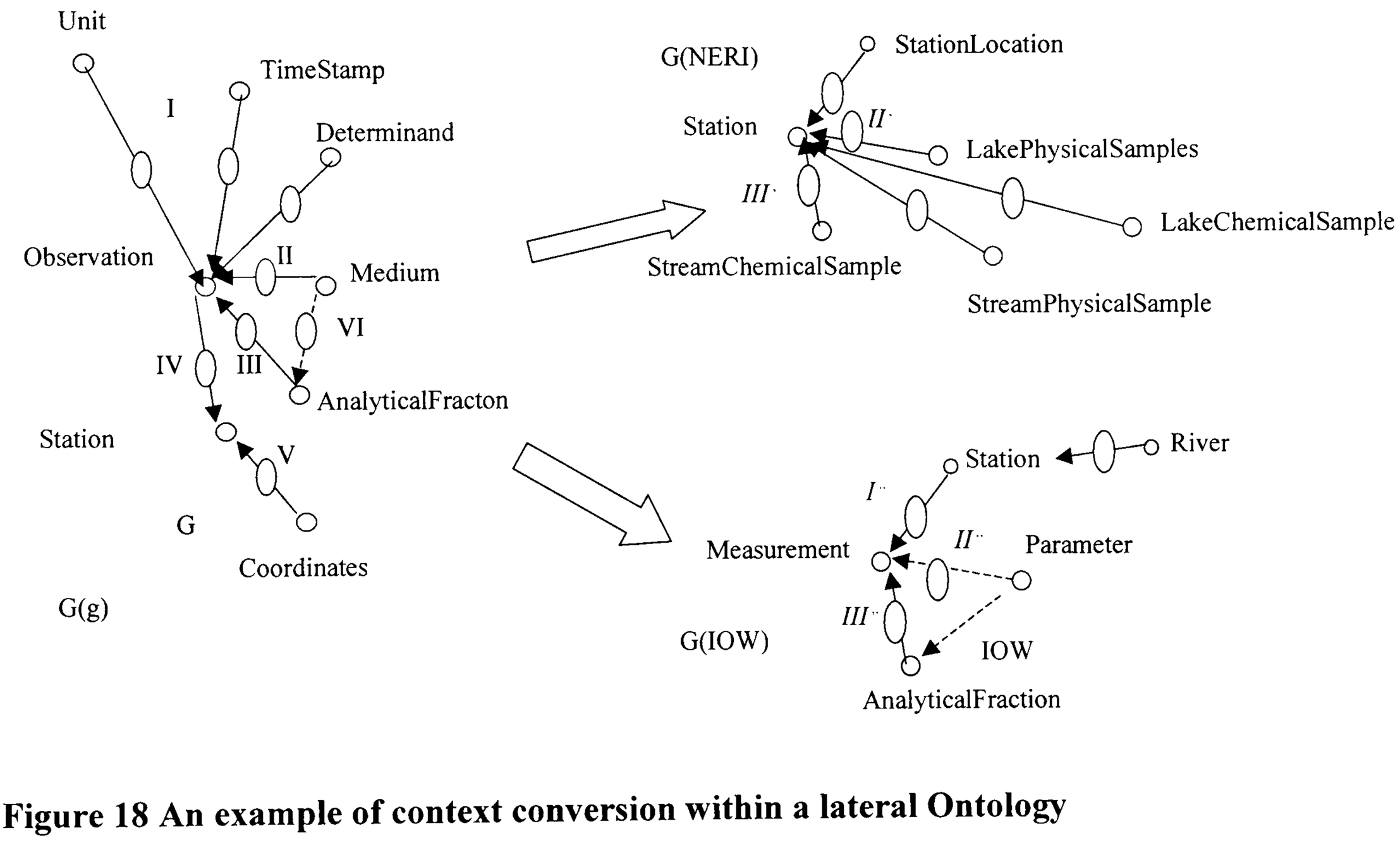

Figure 18 An example of context conversion within a lateral Ontology

During the query transformation in Figure 18, the EGV query Q1 needs to be translated into a corresponding representation of query Q2 in LDV:NERI and query Q3 for LDV:IOW. Notation $Q1 \equiv Q2$ indicates the semantic equivalence is determined for Q1 and Q2, where each node and arc in graph Q2 is determined by Q1 through the ontological mapping functions M(O1,O2). The node names in Q2 and O3 represented

the local database tables with table name translated into English. The validation of such semantic equivalence can de defined as,

$$Q_1 \equiv Q_2 \Leftrightarrow \forall O_1, O_2\, Q_1 \subset O_1, Q_2 \subset O_2 |\ Equals(Result(Q_1, O_1), Result(Q_2, O_2), M(O_1, O_2))$$

where, O1, Q2 are different Ontology models, *Result(x,y)* stands for the function to execute a certain query x in the Ontology model y and returning the tuple set of result back. Function Equals(x,y) compares tuple sets (x,y), giving result true or false. If x can be exactly mapped to y via M(Ox,Oy), then we say x equals to y. The semantic equivalence entails the execution of such queries in any Ontology, must give the identical result set. In Figure 18, given mapping functions between EGV and IOW: M(station, station), M(Observation, Measurement), M((AnalyticalFraction,Medium[4]),AnalyticalFraction), M(Determinand, Parameter), a graph mapping between G1 and G2 can be deduced such as,

{Determinand, I, Medium, II, AnalyticalFraction III, Station, IV } ≡ {Parameter,2, Analytical,3, Station,1, Measurement}

Join relations {1,2,3} are taken to be exclusive compositing routes connecting the mapping entities in LDV:IOW.

The alternative graph may exist when local data source contains multiple paths connecting the corresponding concepts in target Ontology, for example join relation 5 may provide an alternative path between Parameter and Measurement. The determination of path 2 or {5,3} requires the extraction and comparison of semantic characteristics for candidate graphs. The different candidate queries may get reduced or produce extra results from database. In this case Q1 and Q2 are semantic similar queries, denoted as $Q1 \approx Q2$.

The XML/RDF based Ontology language such as DAML and OWL can be represented as the set of triple-tuple statement R = (Subject, Predicate, Object) that can be expressed in the graphic form. The subject and object is an end node in the graph while a predicate can be modelled as the directed arc between two nodes.

Each ontological view can be modelled into a directed graph consisting of object nodes and relation arcs. A query context can be represented as a sub-graph in a particular Ontology graphical model. The query sub-graph should be a connected graph,

---

[4] It denotes the view conversion mapping from global concept analyticalFraction and Medium to local concept AnalyticalFraction.

otherwise, it is an incomplete query. Only the part of connected sub-graph with enquiring property is valid in the IR system.

The mathematics algorithm for the graph model can support powerful information inference functions in the Ontology application, especially for the incomplete or vague mapping case. The complete representation of a query graph Q consists of a set of class node list C={c1,c2…cn}, relation arc list RE={RE1, RE2…REn}, restrictions list R={r1,r2..rn}, while C specifies all related class nodes in the query graph, RE indicates the path to connect all class nodes, R gives the restriction value list associated to the class instance in C. Restriction list contains a set of properties with given or question value. Q is supposed to be a connected graph, if RE cannot connect all C, the additional relations REa has to be inferred to make the graph connected, otherwise the query is not valid for conduction.

During the translation, the application needs to find a corresponding query Q2 in Ontology O2 for the given query Q1 specified in Ontology O1. The queries giving the equivalent meaning can be denoted as $Q1 \equiv Q2$, where Q1 is constructed in different Ontology respectively, O1 and O2, and through the ontological mapping M(O1,O2). The semantic of equivalent query pair can de defined as,

$$Q_1 \equiv Q_2 \Leftrightarrow \forall O_1, O_2\ Q_1 \subset O_1, Q_2 \subset O_2 \mid Equals\ (\text{Re}\ sult\ (Q_1, O_1), \text{Re}\ sult\ (Q_2, O_2), M\ (O_1, O_2))$$

Relation z=*Result(x,y)* stands for the processing to perform a certain query x in the Ontology model y gets the tuple set of result z. Function Equals(x,y) gives the comparison of tuple sets (x,y), giving a result true or false. If x can be exactly mapped to y via M(Ox,Oy), then we say x equals to y.

The similar query graphs are featured by the same nodes and different connection arcs. The similar query Q1 and Q2 are denoted as $Q1 \approx Q2$. If Q1 and Q2 are the connected graph and each node set N_1 and arc set R_1 in Q1 has corresponding node N_2 and arc R_2 in Q2, then N1=N2 via mapping M(O1,O2). The topology of Q2 is uncertain as many relations can be used to connect N2. A graph model is used to deduce and justify different relations in order to build Q2.

In the EDEN-IW system, a local database Ontology is mapped to core Ontology using terms and a view context mapping. The key concepts in the local database Ontology is defined as a view over the core Ontology, but the justification for equivalent relations mapping is more difficult because relations reflect the variation at an abstract level of

the corresponding conceptual model, for example the foreign key relations between tables in local database Ontology cannot be interpreted using core Ontology relations directly. Graphic model can calculate all possible similar queries in the local Ontology in order to map to the core Ontology query.

In the ideal case for good data integrity in the data source that guarantees the unambiguous result of similar queries in the data source, the calculation of the minimum connected graph containing N2 will be the mapping query context in O2.

However, in most real cases, the integrity of data is not guaranteed, similar queries involving the same nodes set using different relations may get different results in the local data source. In order to select which query is better, an uncertainty factor can be expressed as a weighted value for the relation arc. Thus by adjusting the weight value; the calculation of minimum weighted connected graph will have a different topology result. In a limited user query domain, the weight values can be assigned according the enumerated user queries.

4.3.10 Examples of User Query Translation

A user query is a view statement expressed in terminologies of the core Ontology or for other user and application Ontologies. The query needs to be translated into the local database representation, e.g. SQL statement. The Ontology parsing and inferencing function identifies the key concepts for the given user query in the targeted local database Ontology. As described previously, an Ontology defines the conceptual views for the knowledge domain, including all mapping across different views. The translation of user query is performed according to the following sub-processes:

1. Concept translation according to the mapping relations
2. Instance value translation
3. Relation and constraints translation and inference
4. SQL generation and refinement.

The translation of query use case 1 has been analysed in NERI domain as follows:

(Use Case 1): What is the *Observation Value* of *Determinand* Nitrate in *Medium* Water with ANY analytical fraction *measured* at *Station* Z between *time period* T1 and T2?

An additional query of use case 9 is concerning a metadata query about total Nitrogen as follows. "At which the station determinand X been observed above a threshold valueY during period Z?" The determinand totalNitrogen refers wider determinand

compounds of Nitrogen that is different from Nitrate (Nitrogen in the form of NO3- as N).

The UC9 is a typical example of metadata query to index corresponding station information with determinand restriction. A specific example is "At which station has determinand totalNitrogen value above 0.5mg/l during the period between 1980-01-01 and 1985-01-01". This high-level query generates different low-level SQL queries to each database because there are terminological heterogeneities and conceptual heterogeneities. Since the medium and analytical fractions are not mentioned in the query, the system recognises any relevant medium and analyticalFraction as default values. To this extend, the query constraints have been relaxed. According to the mapping relations between EGV and LDV, totalNitrogen may be further extended as aggregated observation of NitrogenCompound such as the sum of InorganicNitrogen, KjeldahlNitrogen, Nitrogens_oxided and TotalAmmonia. This semantic hierarchy structure is defined in EGV ontology. The Resource Agent checks each determinand in the compound group for its availability in the local database. In addition, medium and analytical fraction is also further split to find appropriate interpretation in a local source. The corresponding sub-query will be generated. The NERI database would answer such query with the sum value of nitrate and nitrite observation in water medium with respect to FilteredFraction, DissovledInorganicFraction and SuspendInorganicFraction. The NERI database has stored the river measurement, lake measurement, chemical measurement and physical measurement separately. The relations are expressed as direct mappings between NERI LDV concepts to EGV measurement concepts. Because totalNitrogen is a chemical determinand in EGV, only lake-chemical and river chemical storages are asked. In addition, NERI agent would query IOW database would answer such query with the sum value of InorganicNitrogen, Nitrogens_oxided and TotalAmmonia in both samples of fish and water. The generated sub-queries need to go through the translation processes that are described in section 4.3.10 for term translation, value translation and join path identification. The final product would be the SQL queries in local database syntax. The result set from local database may also contain different information representations, e.g. different unit formats are used in NERI and IOW databases. The corresponding conversion and merging process is performed in Task Agent to harmonise all different value formats according to the associated semantic definition in the result message. Section 4.3.4.2

A second additional example is as follows: which station along water body X has concentration value of Aluminium more than 1mg/l during time period from 1980-01-01 to 1985-01-01?

Resolving this query involves the semantic processing of the vocabulary totalAluminium in local databases. Since the medium and analyticalFraction are not mentioned in the query, all possible combinations of measurement values need to be queried. The metadata search results reflects aluminium may be measured in water, suspended solids and sediments in the IOW database. The corresponding metadata searching in NERI shows only that the relevant concept match is determinand totalAluminium. The sub-queries are generated according to local database schematic syntax. Because the unit value in local database may be different from the input value. The sub-query asks for all measurement values, for example,

```
SELECT mesures.RESULTAT_ANALYSE, parametres.UNITE,
stations.CODE_STATION
FROM stations, [troncons hydrographiques], mesures, parametres
WHERE (((stations.CODE_HYDRO)=[troncons hydrographiques].[CODE_HYDRO])
AND (([troncons hydrographiques].NOM_COURS_EAU)="X") AND
((mesures.CODE_STATION)=[stations].[CODE_STATION]) AND
(parametres.CODE_PARAMETRE = mesures.CODE_PARAMETRE) AND
((mesures.DATE_OPERATION_PRELEV) Between #1/1/1980# And #12/31/1985#));
```

The returned result will be processed in the Task Agent regarding the unit conversion and value comparison. The satisfied results are returned to the user interface.

The NERI LDV shows there are two sub-classes of waterbody w.r.t. river and lake in the NERI domain, the measurement records are stored separately, such that, two sub-queries are generated for river X and lake X w.r.t. chemical measurements, e.g.

```
SELECT feso_maaling.OBS, STBE_STATION.STNAVN,
FROM feso_maaling, STBE_STEDID, STBE_STATION
WHERE ((feso_maaling.STNR= STBE_STEDID.STNR)AND
(STBE_STEDID.STNR= STBE_STATION.STNR) AND
(feso_maaling.PARAM=50)AND (feso_maaling.DATO>19800101)AND
(feso_maaling.DATO<19850101) AND (STBE_STATION.STNAVN="X"))
```

The ontology model and semantic structure has provides a general way to query different database model without knowing more details of the local data source, for

example a simple query of UC9 "Which station has data on determinand X?" can be easily estimated at the global level. The result is shown in the table below.

Table 11. Number of stations found for different determinands

Determinand	WB	NERI	IOW	UK	TOTAL
Antimony			1		1
Aluminium		1	5		6
1,1,1-trichloroethane			19		19
Temperature		522		264	786
BOD		293			293
Oxygen Saturation	2278	92	29		2399
Nitrate	2871	36	29	264	3200
Ammonium	3262				3262
pH	2576	506	29	265	3376

4.3.10.1 Terms Translation

Terms translation handles the concept and property translation to the local database columns in order to build up the SQL query. The terms translation is executed as the metadata query such as "Which column in NERI domain has the equivalent meaning of determinand? ". This helps to translate the concepts between ontological views. The search for the corresponding column name can be executed as to find the concept X satisfying using the following criteria:

- X is a concept in NERI local database view
- X is a column name
- X has an equivalent mapping (terms/view mapping relation) to the core Ontology concepts as given in use case1.

The translation engine starts the search using the terms mapping. If no satisfied concepts and properties can be found, further searches using view mapping relations will be conducted.In the use case 1, following terms translation can be found in NERI database Ontology view, see Table 12.

Table 12 Terms translation for use case 1

User query terms	***NERI local database terms***
Observation Value	*OBS (term mapping)*

Time period	*DATO (term mapping)*
Determinand x? in Medium Water with ANY analytical fraction	*PARAM (view context mapping)*
Station	*STNV (term mapping)*

After term translation, the statement of use case 1 becomes:

What is the OBS value of *PARAM* y? *measured* at STNV Z between DATO T1 and T2? (Use Case 1)

4.3.10.2 Coding Value Translation

The local database system may have different coding values and formats such as data representations, for example Nitrate is coded in core Ontology terms with ID 19, in IOW database it is 1340, in NERI Nitrate may be related to determinand 308. A value coding Ontology defines the coding map between core and local values in the RDF file. Using the name space, the property and concept in RDF Ontology is referred to global or local Ontology for it conceptual interpretation.

The Ontology parsing and inference application uses the global and local terms to check their existing value mapping in the RDF Ontology. If no existing mapping can be found, it means the global value can be used directly in the local representation. For example, some river name can be used directly in a particular local database schema. Further ontological actions can be defined during the value translation as to manipulate time and units. The Ontology can give the semantic interpretation for the time and unit representations.

In use case1, the local representation of value y, Z and T1, T2 can be found .The representation of query is as follows:

What is the OBS value of *PARAM 308* been *measured* at STNV "v1" between DATO "19800101" and "19931023" (Use Case 1)

4.3.10.3 Relation and Constraints Translation

The relation between OBS, PARAM, STNV and DATO is still unsolved at the stage. The translation of relation and constraints across Ontologies is difficult, as the equivalent mapping representation may not exist in another Ontology. Ontology languages like DAML and OWL offer the syntax to link relations together with equivalent semantic meaning. The meaning of semantic relation not only depends on

the definition itself, moreover relies on the representation context and describable concepts related. In most cases, the relation and constraints information cannot be carried during the translation across Ontology.

The graph model was adopted to help the determination of mapping relations as the supplementary method to the current one to one mapping relations.The semantic meaning of a query sentence was represented as the question for the value of certain property while other property values and relations are given as constrains. Represented into graphic model, the Ontology is a directed connect network in which each concept is a node and each relation is a directed arc.

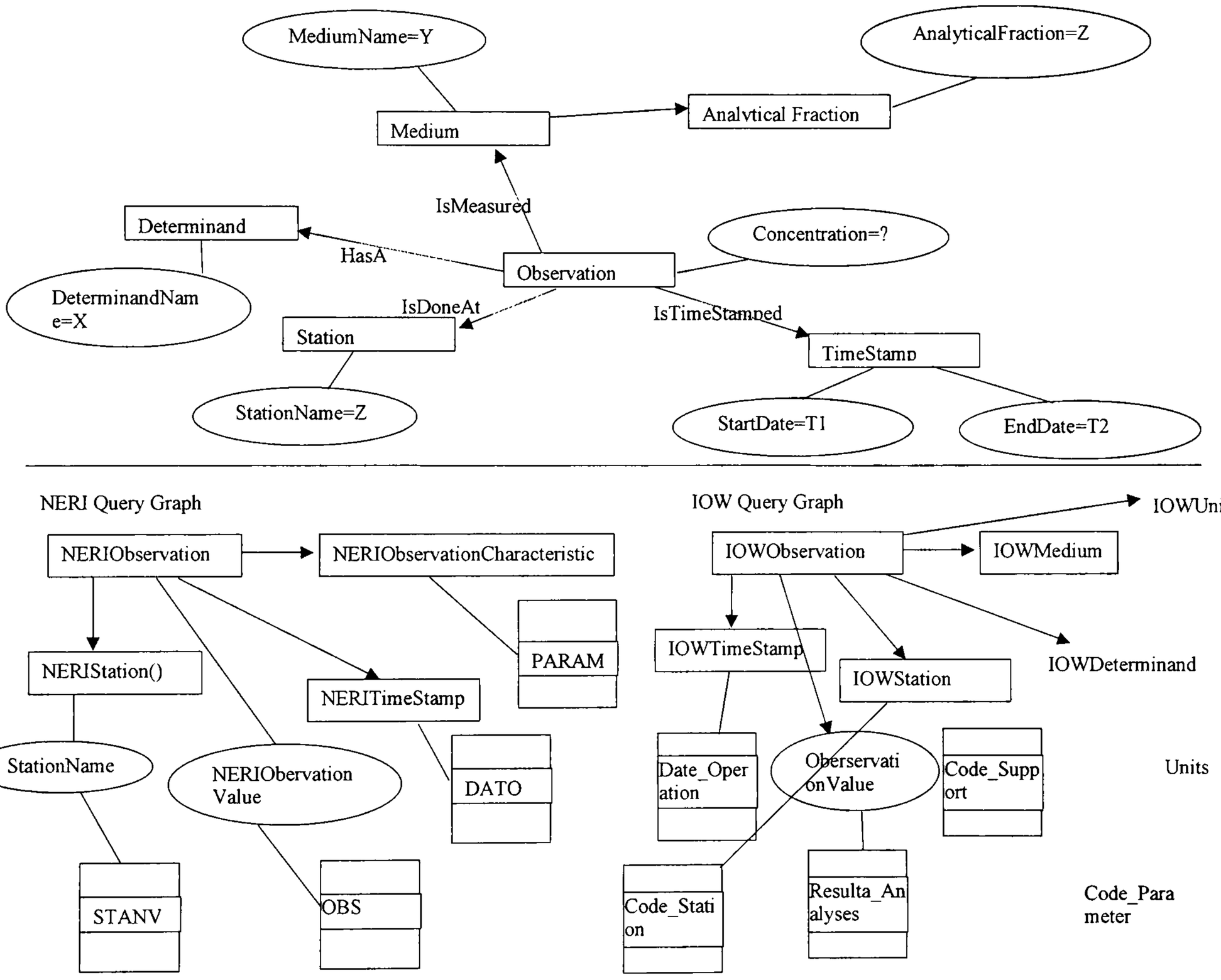

Figure 19 Graphic representation of UC1

4.3.10.4 RDF representation of user query

The system uses an RDF schema as the lingua-franca to encode the SQL query into content that agents could exchange information and tasks about. More specifically, the RDF schema was used as the content language for FIPA agents that exchanged ACL messages. The specification of the RDF schema was given in the FIPA specification 11 [1] that was experimental at the time of the project. In EDEN-IW project, the FIPA RDF content language has been expanded to contain more semantic-rich information

and to allow the correlation to EGV concepts, see Appendix II for the RDF schema. The basic schema is based on FIPA-rdf0 and FIPA-rdf1. There are two different schema, one to support tasks delegation, rdf-1 that allows one agent to request another to perform an action on its behalf and another schema rdf-1 that supports queries and assigns values to free variables in the results. The FIPA-rdf1 class has been expanded to contain two additional RDF classes for the query in terms of an SQL query and the results in terms of an SQL query results.

The user query can be encoded into RDF message according to above schema. For example, a user query is asking for "The names of stations that have observed record for river Thames, and have PH value greater than 7 in year 1980" is expressed in RDF format as below message.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rdf:RDF xmlns:rxsd="http://www.w3.org/2000/10/XMLSchema#"
xmlns:fipa="http://edeniw.elec.qmul.ac.uk/metadata/FIPA/20030627/CL-RDF/fipa-
rdf0#" xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:EDEN="http://edeniw.elec.qmul.ac.uk/metadata/CoreOntology/20030611/EGV
/EDEN_IW_Global.daml#"
xmlns:EDENSchema="http://edeniw.elec.qmul.ac.uk/metadata/UA/20030627/GetPara
meter.rdfs#">
<fipa:Query rdf:ID="DataQuery">
  <fipa:actor>ResourceAgent</fipa:actor>
  <fipa:act>EnhancedUC9</fipa:act>
  <fipa:conversationID>1</fipa:conversationID>
  <fipa:done>false</fipa:done>
  <fipa:status>TAContacted</fipa:status>
<fipa:Rule rdf:ID="EnhancedUC9">
<fipa:selection-result rdf:ID="stations"/>
 <fipa:selection>
  <rdfq:rdfquery>
   <rdfq:From eachResource="&EDEN:Station/">
    <rdfq:Select>
     <rdfq:Condition>
            <rdfq:and>
```

```
            <rdfq:equals>
              <rdfq:Property name="&EDEN:StationName"/>
              <rdfq:String>Thames
              </rdfq:String>
            </rdfq:equals>
<rdfq:equals>
              <rdfq:Property name="&EDEN:DeterminandName"/>
              <rdfq:String>PH
               </rdfq:String>
</rdfq:equals>
<rdfq:greaterThan>
 <rdfq:Property name="ObservedDate"/>
  <rdfq:Integer>1980-01-01</rdfq:Integer>
</rdfq:greaterThan>
<rdfq:lessThan>
 <rdfq:Property name="ObservedDate"/>
  <rdfq:Integer>1980-12-31</rdfq:Integer>
                </rdfq:greaterThan>
                        </rdfq:and>
                  </rdfq:Condition>
                </rdfq:Select>
              </rdfq:From>
            </rdfq:rdfquery>
          </fipa:selection>
        </fipa:selection-result>
</fipa:Rule>
</fipa:Action>
  </rdf:RDF>
```

The corresponding result set returned from resource agent is encoded as below:

```
  <?xml version="1.0" encoding="ISO-8859-1" ?>
<rdf:RDF
xmlns:Agent="http://edeniw.elec.qmul.ac.uk/metadata/da/20030627/agent.rdfs#"
xmlns:EDEN="http://edeniw.elec.qmul.ac.uk/metadata/CoreOntology/20030611/EGV
/EDEN_IW_Global.daml#"
```

```
xmlns:Service="http://edeniw.elec.qmul.ac.uk/metadata/da/20030627/service.rdfs#"
xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
xmlns:fipa="http://edeniw.elec.qmul.ac.uk/metadata/FIPA/20030627/CL-RDF/fipa-
rdf0#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rxsd="http://www.w3.org/2000/10/XMLSchema#">
<fipa:Query rdf:ID="DataQuery">
<fipa:actor>RAIOW:AID</fipa:actor>
        <fipa:act>EnhancedUC9</fipa:act>
        <fipa:conversationID>1<fipa:conversationID>
        <fipa:done>true</fipa:done>
<fipa:result>
                <fipa:selection-result rdf:ID=”1”>
                        <EDEN:StationName>
                                <rxsd:string>
                                <rdf:value>Rodemark</rdf:value>
                                </rxsd:string>
                        </EDEN:StationName>
                        <EDEN:StationID>
                                <rxsd:string>
                                <rdf:value>00452</rdf:value>
                                </rxsd:string>
                        </EDEN:StationID>
                        <EDEN:RiverName>
                        <rxsd:string>
                                <rdf:value>Thames</rdf:value>
                                </rxsd:string>
                        </EDEN:RiverName>
                                <EDEN:StationDescription>
                                        <rxsd:string>
                                        <rdf:value>null</rdf:value>
                                        </rxsd:string>
                                </EDEN:StationDescription>
                </fipa:selection-result>
</fipa:result>
```

</fipa:Query>

</rdf:RDF>

The extended RDF content language supports a SQL-like representation for a user query message. A query message contains the key items: actor, act, conversationID, done and status. When a user agent raises the question about database contents, the element rule defined in RDF query language is used to represent the constraints of user query. Element result is a container of a query result, each tuple set is filled into a container of selection-result in the query result. The semantic of RDF language is defined for general database access.

The semantic meaning of the user query can be modelled in the core Ontology in Figure 19. As the translation is undertaken through the global Ontology to the local database Ontology, the topology of query graph may vary, because the classification of domain knowledge is different in the respective local database views. In a sentence, the meaning of query is determined by the relation definition and related concepts. During the translation, the concepts and properties may have one to many or many to many mapping relations across Ontologies. Graphic model is going to inference the possible relations to link two related concepts in any ontological view. The concept and property in the global terms can be translated to the identified column names in the database view. In order to build up the SQL statement, the joining relations amongst tables in the database schema have to be discovered. The graph model traverses the Ontology graph of local database Ontology to find the joining relations for the different tables. Further research will focus on the justification of proper relations to join different tables according to the query context.

The user query is expressed using terms in the EGV Global ontology and is encoded in RDF. This RDF query needs to be rewritten into SQL syntax for local database access. The process of transforming or rewriting the global query into local queries is described as follows:

1. The equivalent concepts and properties are identified in LDV regarding query input contained in fipa:selection and rdfq:condition. The algorithm checks the RDF semantic network to find a possible connected graph for user query. The sub-graph of user graph is analysed with EGV-LDV mapping to find any vocabulary and view substitution.
2. The constraint value is substituted by corresponding value mapping in LDV regarding the identified concept and properties in step 1.

3. The semantic routing algorithm in section 4.3.9.3 is used to identify the most likely join path to generate join relations in SQL query.
4. The generated SQL syntax is modified to cope with the different database type.

Two more examples are given here to illustrate the translation process to rewrite a user query in RDF into SQL syntax.

The first example is like "Which River has determinand PH value over 7 during time period 1980 to 1990? " The main part of RDF query can be written as follows:

```
<fipa:selection>
  <rdfq:rdfquery>
    <rdfq:From eachResource="&EDEN:River/">
      <rdfq:Select>
        <rdfq:Condition>
        <rdfq:and>
          <rdfq:equals>
           <rdfq:Property
name="&EDEN:DeterminandName"/>
            <rdfq:String>PH
            </rdfq:String>
          </rdfq:equals>
        <rdfq:greaterThan>
           <rdfq:Property
name="ConcentrationValue"/>
            <rdfq:Integer>7</rdfq:Integer>
        </rdfq:greaterThan>
        <rdfq:greaterThan>
         <rdfq:Property name="ObservedDate"/>
          <rdfq:Integer>1980-01-01</rdfq:Integer>
        </rdfq:greaterThan>
        <rdfq:lessThan>
         <rdfq:Property name="ObservedDate"/>
          <rdfq:Integer>1990-12-31</rdfq:Integer>
        </rdfq:greaterThan>
      </rdfq:and>
    </rdfq:Condition>
```

```
        </rdfq:Select>
      </rdfq:From>
    </rdfq:rdfquery>
  </fipa:selection>
```

The EGV terminology in the RDF query is replaced by its semantic equivalence in LDV, for example the following mappings can be found between EGV and IOW LDV (see table below):

Table 13 Identical concepts in query rewriting: example 1

EGV Terms	*LDV Terms*
Determinind = PH	*Code parameter =1302*
RiverName	*Nom cours d'eau*
Concentration Value >7	*Résultat analyse >7*
Date	*Date operation prelev*

The query rewritten algorithm generates a SQL-like query in Select-From-Where syntax with all mapping terms substituted. The join path routing algorithm traverses the IOW LDV semantic network to find appropriate join relations to link all relevant terms. The selected foreign key relations are specified in Where clause to join relevant tables. The final SQL query is like:

```
SELECT mesures.DATE_OPERATION_PRELEV, mesures.RESULTAT_ANALYSE,
parametres.NOM_PARAMETRE_COURT, [troncons
hydrographiques].NOM_COURS_EAU
FROM  measures, parameters, troncons hydrographiques, stations
WHERE (((parametres.CODE_PARAMETRE = mesures.CODE_PARAMETRE)
AND (mesures.CODE_STATION = stations.CODE_STATION) AND
(stations.CODE_HYDRO = [troncons hydrographiques].CODE_HYDRO) AND
([mesures].[DATE_OPERATION_PRELEV]>#1/1/1980#) AND
([mesures].[RESULTAT_ANALYSE]>7)) OR (("AND
[mesures].[DATE_OPERATION_PRELEV]"<#12/31/1990#));
```

The second example is like “What is concentration of determinand Arsenic during time period 1980 to 1990 in station Rodemark? ” The main part of RDF query can be written as follows:

```
<fipa:selection>
   <rdfq:rdfquery>
     <rdfq:From eachResource="&EDEN:Concentration/">
       <rdfq:Select>
         <rdfq:Condition>
           <rdfq:and>
             <rdfq:equals>
               <rdfq:Property name="&EDEN:StationName"/>
                <rdfq:String>Rodemark
                </rdfq:String>
             </rdfq:equals>
           <rdfq:equals>
               <rdfq:Property
name="&EDEN:DeterminandName"/>
               <rdfq:String>Arsenic
               </rdfq:String>
           </rdfq:equals>
         <rdfq:greaterThan>
            <rdfq:Property name="ObservedDate"/>
             <rdfq:Integer>1980-01-01</rdfq:Integer>
           </rdfq:greaterThan>
           <rdfq:lessThan>
            <rdfq:Property name="ObservedDate"/>
             <rdfq:Integer>1990-12-31</rdfq:Integer>
               </rdfq:greaterThan>
         </rdfq:and>
           </rdfq:Condition>
         </rdfq:Select>
       </rdfq:From>
   </rdfq:rdfquery>
</fipa:selection>
```

Because the Rodemark is indicated in Directory Agent as a Danish station, the query only goes to the NERI resource agent. The EGV terminology in the RDF query is substituted by its semantic equivalence in LDV, for example following mappings can be found between EGV and the IOW LDV, see the table below:

Table 14 Identical concepts in query rewriting: example 2

EGV Terms	***LDV Terms***
Determinind = Arsenic	*param =55 or param =56 or param =57*
StationName	*STNAVN*
Concentration	*OBS*
Date	*DATO*

The query rewritten algorithm generates a SQL-like query in Select-From-Where syntax with all mapping terms substituted. The query without medium and analytical fractions is considered in all possible combination to find the mapping concept in NERI LDV. Three relevant mapping concepts are found from the value mapping in the LDV interim classes. The join path routing algorithm traverses the NERI LDV semantic network to find appropriate join relations to link all relevant terms. The selected foreign key relations are specified in Where clause to join relevant tables. The final SQL query is:

SELECT vakevl_analyse.PARAM, vakevl_analyse.OBS, vakevl_analyse.DATO
FROM vakevl_analyse, STBE_STATION, vakevl_proeve
WHERE (((vakevl_analyse.PTNR = vakevl_proeve.PTNR) AND
(vakevl_analyse.STNR = STBE_STATION.STNR) AND ((vakevl_analyse .
PARAM=55) OR (vakevl_analyse . PARAM) OR (vakevl_analyse . PARAM)) AND
(vakevl_proeve Between 19800101 And 19901231) AND
(STBE_STATION.STNAVN= “Rodemark”));

4.3.10.5 Use Case Implementation

```
<?xml version="1.0" encoding="UTF-8" ?>
- <query>
- <column>
  <element>param ID</element>
  <element>station name</element>
  </column>
- <constraint>
  <element name="DeterminandID"
type="Determinand">4</element>
  <element name="MediumID"
type="Medium">*</element>
  </constraint>
  </query>
```

Figure 20 An example of XML Query input

SQL generation supports the semantic context translation between RDF/XML query and SQL statement. Basically, we can give the syntax of user query in a common structure, see Figure 20. A user query specifies the value of concepts with given constraints. The query statement is easily represented in a SQL-like query structure that consists of querying arguments and a constraint statement. The former set is represented in XML tag *column* and the later one is *constraint*. This sort of XML representation actually has hard-coded the semantic logic of user query in a structure: each user query asks for the value of one or more properties or columns with its constraints. In the SQL-like syntax, the XML query above can be stated in global terms as:

```
SELECT  DISTINCT Determinand.determinandName,Station.StationName
FROM Determinand, Station , Observation
WHERE(  Determinand.DeterminandName='PH' AND (Observation
isObservedAbout Determinand) AND (Observation IsTakenAt Station)  )
```

The local SQL statement in IOW domain is:

```
SELECT  DISTINCT
parametres.code_parametre,stations.localisation_globale
FROM parametres, stations , mesures
WHERE(  (parametres.CODE_PARAMETRE=mesures.CODE
_PARAMETRE) And (mesures.CODE_STATION=stations.CODE_STATION) And
( parametres.CO DE_PARAMETRE=1311 ))
```

When translating between local and global SQL queries, the similarity and differences can be found for the case above using the translation process as follows:

1. The semantic meanings of those two queries are equivalent for query execution in the IOW database domain.

2. The meanings of sub-clause of *Select* and *From* are semantically equivalent.
3. The local *Where* gives further information about the access of local data model that is not specified in the global query. Whereas in the global query, the semantic relations between concepts may be given or implied.
4. The clause translation for *Select* and *From* can be easily completed if the one-to-one mapping relations between global and local terms can be detected.
5. If no one-to-one relation is specified between global and local terms, an inference action is required to prove the terms and representation can be chosen.
6. The clause *Where* may get more complicated than term mapping and inferencing, because the relation specifications in two sentences are not consistently normal.

For points 1 and 2, using the Ontology service method, the global terms and values in a XML query can be translated to the local terms and values directly. Browsing the local model, the SQL building service can find the table name for the particular columns. Then the only question is how to join these tables together and form the *where* section in SQL. The graph algorithm helps to calculate the joining path between any tables. We can imagine each table as an individual node in a graph, and each foreign keys as the arcs to link different nodes together, then the calculating of joining path become the calculating to a going through path between given nodes. For example, for the use case 1: *What is the Observation Value of Determinand X in Medium Y has been measured at Station Z between time period T1 and T2? (Use Case 1)* the processing for the NERI case is as follows

1. Direct mapping was defined in Ontology model:
2. Observation Value becomes Table Ti, Column Ci
3. Station becomsTable Tk, Column Ck,
4. Time becomes Table Tl , Column Cl

Logic Conversion actions are defined for the NERI domain,

5. EGV Determinand & Medium becomes LOCAL DATABASE ONTOLOGY NERIObservationCharacteristics
6. Direct mapping was defined in the LOCAL DATABASE ONTOLOGY
7. LOCAL DATABASE ONTOLOGY NERIObservationCharacteristics becomes Table Tj, Column Cj,
8. Then we have value translation from EGV (X,Y) to LDV (Z)
9. And value translation so that (X,Y) becomes Z

Query semantic analysis

10. What's the value(Ti,Ci) with the restriction of Value(Tj,Cj) = X, Value(Tk,Ck) = Z, T1< Value(Tl, Cl) < T2?

SQL generation involves determining how to join tables.

11. The Graph methodology is used to calculate the path to join table Ti, Tj, Tk, Tl.
12. Now we have the information necessary to build up the use case 1 SQL query for the NERI domain:

Select distinct Ti.Ci, Tj.Cj, Tk.Ck, Tl.Cl

From Ti, Tj, Tk, Tl

Where (Value(Tj,Cj)=X, Value(Tk,Ck)=Z, T1<Value(Tl,Cl)<T2) and (joining path of Ti, Tj, Tk, Tl)

4.4 EDEN-IW Middleware Architecture

4.4.1 Motivation for Using MAS

Previous work by other researchers using the InfoSleuth based agent architecture [71] in an earlier related project to EDEN-IW, has demonstrated the potential of multi-agent systems and semantic approaches to enhance environmental information retrieval. In contrast to the InfoSleuth approach, the EDEN-IW system adopted a more open system approach in terms of its use of specifications for the multi-agent systems and Ontologies and in the way the semantic metadata architecture was modelled [100]. The main EDEN-IW system requirements are to support high-level queries in terms of query transparency and data harmonisation. To do this it seeks to leverage two underlying technologies: Ontologies and agents. The use of an Ontology model to support the exchange of semantic machine-understandable structured data, automated processing and to enhance information queries and information searches, has already been discussed in detail Section 4.3. Semantic processing, by Multi-Agent Systems (MAS), supports the opportunity: to expand a user query depending on the context (query augmentation); to integrate and aggregate the content (Content harmonisation); and to use the semantic model to classify, (re)structure and index information.

However, more than a defined semantic data model is needed to achieve interoperability. Semantic services are needed to import, parse and process semantic metadata instances, to map them to data resource instances and to distribute and coordinate the metadata. Hence, secondly, the EDEN-IW system is based upon a MAS

model. The use of the MAS model gives added benefits, beyond providing a semantic metadata processing and distribution framework. MAS communication is usually based upon an underlying communication protocol speech act theory that treats communication as actions [82]. This gives a powerful approach to integrate human intentions and computation system service actions and to communicate about processing and meta-processing, e.g., an analysis of the reasons to communicate in a particular way or to change the communication. Using this underlying model, MAS agents can coordinate messages, and process and reason about the semantic message exchange.

Later developments of speech act theory have included modelling the intentions of the sender in initiating communication. These can range from weak intention such as an intent to send without any consideration of the receiver such as a cry for help, to stronger intentions to initiate a specified reaction from the receiver to the still stronger intention to alter the beliefs of the receiver or for the receiver to take on board the beliefs of a third party [35].

There are several additional potential benefits to using a Multi Agent System (MAS). Fundamental properties that characterise MAS agents are autonomy, reactivity, pro-activity and sociability. Autonomy refers to an agent is able to act without the direct intervention of humans (or other agents), and that it should have control over its own actions and internal state. Reactivity refers to agents being able to adapt to changes in the environment and in response to message from other agents. The pro-activity of MAS system refers that agents should not simply act in response to their environment, they should be able to exhibit opportunistic, goal-directed behaviour and take the initiative where appropriate; The sociability indicates that agents should be able to interact, when appropriate, with other artificial agents and humans in order to complete their own problem solving and to help others with their activities[47]. The EDEN-IW MAS system supports these fundamental properties of agents. For example, the Task Agent provides an autonomous service for flexible and adaptive task planning and query decomposition. The Resource Agent support flexible wrapping service to mediation query between global representation and local data sources. The Directory Agent shows the pro-activity to monitor the metadata change in real data sources on a basis of event-based and periodical mechanism. MAS provides a rich set of messaging protocols to share and converse about the semantic model The message protocols used are independent of the application domain and hence the same communication protocol

or set of actions such as send, acknowledge, refuse and reply can be used across applications supporting a greater consistency and ease of invocation of common actions across multiple applications [79]. Hence, agents support a transparency and virtualisation notion because agents present a common set of communication actions to allow users to invoke different database resources and different data processes transparently. Agents can support the concept of dynamic virtual organisations, acting to a degree autonomously but organising themselves driven by the interaction context, e.g., requestors and suppliers could be organised and interact according to a master-slave relationship or according to a market-place. That is, agents leverage the duality that exists between organisation and interactions in which an organisation is defined by the interactions it supports, and interactions exist and are constrained with respect to a particular organisation [35].

Multiple-agents can flexibly solve complex information retrieval problems such as data harmonisation and aggregation from multiple data sources using autonomous specialised agents that can coordinate their individual actions or compete with each other to solve a problem: agents support cooperative planning to coordinate the actions of others to solve a problem that they cannot solve alone. Agents can support multiple redundant plans, switching to alternative plans if one fails thus offering support for fault-tolerance.

Agents can act as powerful service mediators supporting flexible service requests to capability matches and isolating requestors from providers, e.g., to act as a one-stop shop to hide the requestors from the complexities of composite service invocation and interaction but also to provide privacy, impartiality to requestors and providers.

Agents can reason about messages that contain logical expressions in order to provide the processing to support content harmonisation and to provide the flexibility to optimise the interaction according to the application context.

From this a set of specific useful problem-driven properties of agents (See Table 15) is highlighted, i.e., to support specific problem-driven requirements and to provide a useful model to analyse a problem. This is in contrast to a tendency to introduce solution-driven models of agents, i.e., agents are potential generic technological solutions merely because they have a set of useful characteristics such as autonomy, proactivity, mental deliberation, and an ability to support rich coordination.

Table 15 Information retrieval application requirements and the corresponding agent properties that can be used to support them

Information retrieval application Requirements	***Corresponding Agent Properties***
Usage Transparency	Agent Communication is based upon an underlying communication protocol *(speech act theory) of treating communication as actions* that can integrate human intentions and computation system service actions.
Resource & service virtualisation - a set of universal service actions is supported	Agent Communication uses a *common set of communication protocol actions* used across all service instances
Virtual Organisation that are formed on demand, to solve problems	Agents use *plans to achieve goals, to coordinate* actions of themselves and those of other agents
Fault-tolerance	Plans can contain *redundant plans*; switching to alternative plans when one plan fails.
Protocols for Semantic metadata and knowledge exchange	Common set of *communication protocol actions supports knowledge exchange.* The communication protocol provides a process context to interpret the content.
Dialogues such as flexible service request-provision mediation	Agent communication supports a r*ich set of dialogues* such as contract-net, subscription, auctions etc
Reasoning about logically expressive (semantic) messages	*Agents supports reasoning with proposition, rules and desired states*

4.4.2 EDEN-IW MAS System Design and Implementation

The conceptual architecture of EDEN-IW information system follows a conventional 3-tiered information architecture design (Figure 5) consisting of a resource management layer, an application logic layer and a presentation layer. In a heterogeneous distributed system, such as EDEN-IW, (agent) functional components in each of these layers can be distributed and heterogeneous. In the EDEN-IW system, functions in each these layers are integrated using a Semantic Web metadata model and a multi-agent infrastructure.

Each of the main three layers such as the user portal presentation layer may be complex enough for itself to be internally organised as a tiered sub-architecture. A multi-agent system is a good potential architecture for integrating heterogeneous

databases in that agents are naturally distributed and autonomous; they can use rich explicit communication protocols to interoperate and they can naturally link to semantic models to help resolve interoperability problems.

Each of the agents has a specific task in the complex process leading from the formulation of specific (but database independent) queries, through to the specific queries sent to databases which the agents evaluate as potentially having an answer or part of the answer for the actual question. Such requests for information require a common "language", a list of accepted and well-defined words, that is, the basis for an Ontology based Semantic metadata model relevant for inland water. When results are returned, post processing will be performed in order to furnish the user with information in the most useful form. This post processing consists of harmonising, aggregating, and presenting information in a consistent form, allowing variation in the level of detail presented, and integrating decision support tools for environmental management for the benefit of policy makers.

The EDEN-IW system, see Figure 21, is viewed as a dynamic organisation of software agents that interact using an Agent Communication Language (ACL). The functional roles of agents are dynamic and depend on the interaction of a multi-agent organisation. An agent may play multiple roles in different interactions. Although many services with the EDEN-IW system will be accessible via the agent interface, some lower level services are available via a non-agent interface such as a Web-service. There are two main reasons for this. Firstly, it is too inefficient for some services, e.g., if the message transport service were an agent we would need to send another agent message to send each agent message. Secondly, some services such as the database resources already have robust standard non-agent interfaces such as SQL.

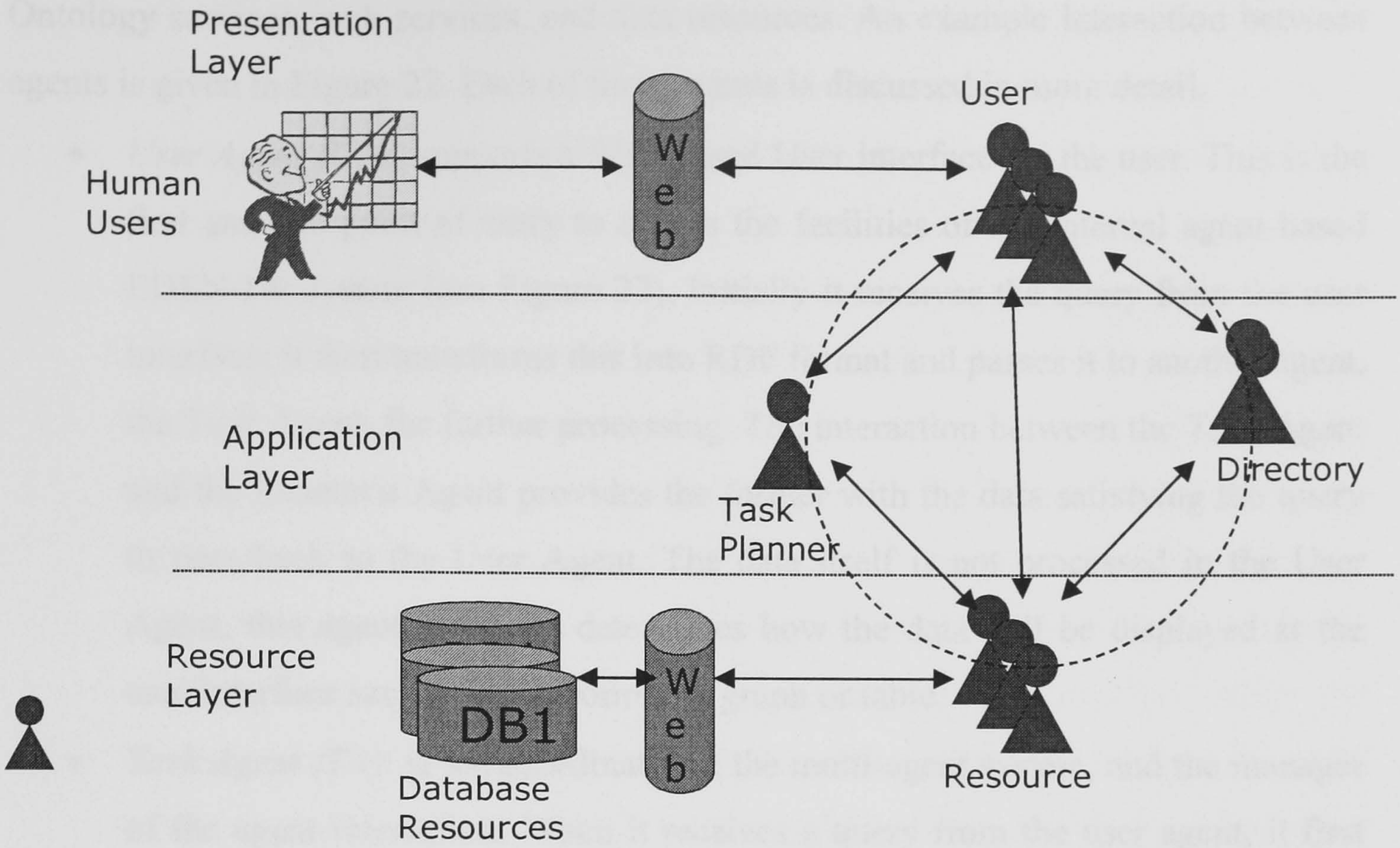

Figure 21 Agents in the EDEN-IW System

In EDEN-IW, MAS design decisions include:

- Selecting interaction protocols, for example, using a query versus using a request or using a subscribe versus using request-when communicative acts;
- The agent mediators, third-parties that can aid interaction between two parties, defined in the FIPA agent management specification only support one type of interaction (request), additional interactions often need to be handled. The directory agent (Figure 22) is derived from the basic FIPA agent rather than sub-classed from the FIPA DF agent in order to utilise additional interaction protocols.
- The design, use and management of domain specific Ontologies (See Ontology section 4.3);
- The design of wrappers to wrap information from non-agent resources such as databases;
- Parsing of ACL messages is quite intensive. Hence cut-down version of the ACL messages and transport protocols, not specified in the FIPA Message transport specification may be used between homogenous FIPA agent platforms within the same domain.

There are four types of agents in the system (see Figure 21): user agent (UA), directory agent (DA), task agent (TA), and resource agent (RA). Non-agent components include

Ontology services, web services, and data resources. An example interaction between agents is given in Figure 22. Each of these agents is discussed in more detail.

- *User Agent (UA):* supports a Web-based User interface for the user. This is the first and last point of entry to access the facilities of the internal agent-based EDEN-IW system (see Figure 22). Initially it receives the query from the user interface; it then transforms this into RDF format and passes it to another agent, the Task Agent, for further processing. The interaction between the Task Agent and the Resource Agent provides the former with the data satisfying the query to pass back to the User Agent. The data itself is not processed in the User Agent, this agent however, determines how the data will be displayed at the user interface such as in the form of a graph or table.
- *Task Agent (TA)*: is the coordinator of the multi-agent system, and the manager of the agent interaction. When it receives a query from the user agent, it first analyses the query, then different plans are made according to different queries, e.g., one or more plans for different use cases such as *What is the concentration of X in River Y at time T ?* or *Which stations have data on Determinand X?*

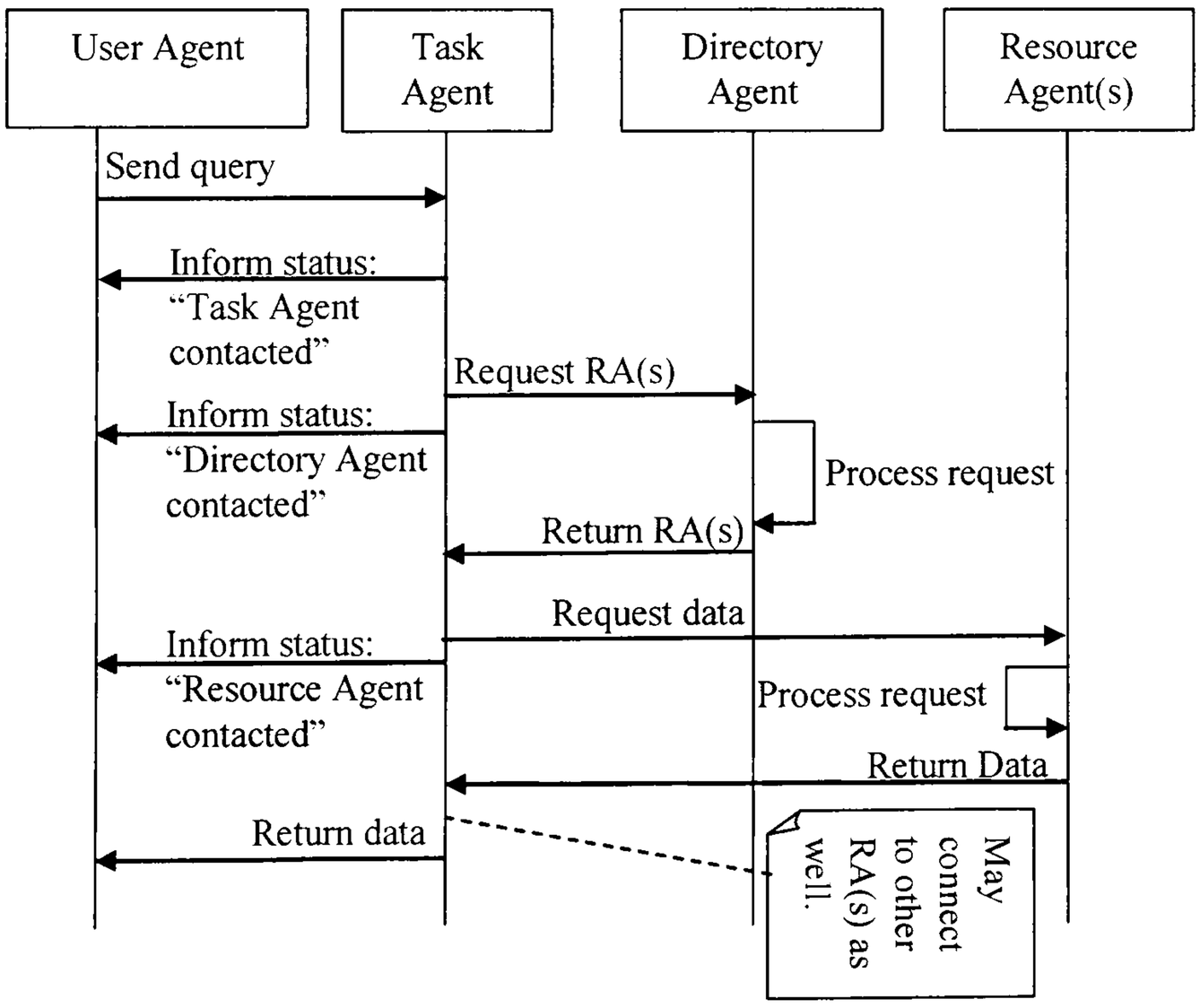

Figure 22 Example of multi-agent interaction triggered by user-queries handled in the EDEN-IW system.

- *Directory Agent (DA)*: is the central repository for metadata. It stores a list of all available stations and active resource agents. It also holds a list of all determinands available at a particular database. These resources are monitored

and this information is regularly updated. The directory agent helps the task agent to locate the appropriate resources for each use cases. At a lower level, it keeps a list of all active agents in the EDEN-IW system. Agents can register, deregister or search the Directory agent for other agents.

- *Resource Agent (RA):* provides the resource wrapping service to access conventional databases to retrieve the IW data. A web access model has been built to secure access to the database information. The wrapping service accesses both EGV (EDEN Global View) and LDV (Local Database View) Ontologies via appropriate Ontology 'adaptation' services. The heterogeneity of the inherent database structure and language representation has been hidden from the common EGV language and semantic representations. Thus the resource agent can translate the global query from EGV to LDV and the local database query result from LDV to EGV in reverse. In the prototype demonstration, two resource agents have been created, IOW and NERI, each agent accesses the corresponding LDV in the Ontology storage for EGV/LDV translations.

4.4.3 Agent Message Interfaces

The content of agent messages in the EDEN-IW system are represented in RDF rather than DAML+OIL because RDF has been proposed as a FIPA content language whereas DAML+OIL has not. Further, service action invocation does not need a very expressive semantics. Instead, it often needs to be simple and quick to invoke.

ACL message headers provide a generic communication context for service specific information and actions, e.g., the ACL message defines a message type (communicative act) such as query and the ACL message body defines the parameters or constraints for the query. As an example, consider the user interface that collects and validates parameters that have been entered by the user. The corresponding User Agent then creates a RDF message to interface to the rest of the agents in the EDEN-IW System. This message body contains all the information required by the agents. It specifies the following:

- A FIPA Action Tag – This provides unique message identification.
- A FIPA actor Tag – This defines the actor (agent) who initiated this action
- A FIPA act Tag – This defines the action to be carried out e.g., "GetParameter".

- The FIPA argument Tag – This gives the input parameters required to perform the query. These constraints are specified using the Global EDEN Ontology.
- The FIPA done Tag – This is used to track the status of the process being performed such as "start , then "pending" and finally "done".

The FIPA ACL messages can be encoded using XML as defined in FIPA specification 71 [FIPA], see Figure 23. When an agent wants to send a XML encoded message, all it has to do is to call the API used by the FIPA platform to create a message to send and to fill in the fields of the message. Agent communication message are transported over HTTP and Java RMI (Remote Method Invocation). Agents located on the same machine use RMI to call methods on each other whereas communication with agents on other machines is done over HTTP, one of the most common protocols used over the Internet and more likely to be accepted by firewalls. Thus, an additional transport envelope header in XML, as specified in FIPA specification 00085 [FIPA] is added when agents are sent by FIPA agent transport services in different agent platforms. Agents do not however see this transport envelope header as it is stripped away by the message transport service.

```
<?XML version …>
xmlns="http://www.fipa.org/schemas/acl#">
<communicative-act> ……
….
<content> … </content>
```

Figure 23 A fragment of an FIPA-ACL header in XML

4.4.3.1The User Agent

The User Agent collects parameters that have been entered by the user at a Web-based user interface, parameters are validated by the user interface and a RDF / XML message is generated from these parameters as described above to enable other agents in the EDEN-IW System to cooperate to solve the task of answering the user query. The RDF part of the message is the content of the ACL / XML header message that is passed from the User Agent to a Task Agent.

4.4.3.2 Agent Tasks and the Task (Planning) Agent

The tasks of multiple agents often need to be combined and coordinated because no single agent has all of the information and capabilities to perform a task itself. A

special agent, called the Task (Planner) Agent or TA is used to plan and coordinate the composite tasks of multiple agents when they are required to achieve a goal.

To co-ordinate the interactions between different Agents, the Task Agent plays a role in harmonising communications. According to its functional requirements, the task agent consists of four major parts: decomposition unit, assembly unit, exception handling unit, and task scheduler unit. The Task Agent knows the role of other agents, i.e. what other agents could do, what problems they can solve. The task description is updated and maintained in directory agent, and provided to task agent. The decomposition unit breaks down the incoming queries into those that can be handled in the Ontology service and by resource agents. It then allocates each sub-query some actions accordingly. The scheduler unit schedules the actions while the exception handling unit monitors and deals with any exceptions that may happen. Finally results from resource agents are assembled in the assembly unit and sent back as replies to the UA. The TA is currently implemented as a JADE agent and simple plans are implemented using Java language constructs. As far as scalability of the system is concerned, there may be a pool of task agents available or task agent is designed multi-threaded to handle a greater number of incoming queries.

4.4.3.3 The Directory Agent

The EDEN-IW directory agent acts as a repository for Agent Descriptions and Service Descriptions. Agent descriptions define the Agent-name, Agent-locator (One or more transport-descriptions), the tasks they support, the domain Ontologies and the Interaction Protocols they support [1]. The service descriptions for the resource service entries in the directory are expressed in terms in the EGV Ontology such as the key concepts of geographical regions (e.g., groups of stations), Inland Water parameters and time.

The directory agent compiles summary metadata about each database resource. For example it knows the determinands, the time range and the list of stations that a database covers. When a service agent such as a resource agent is started, it can advertise itself in the directory agent and thus makes itself available for use (client data pull). In addition, user agents can register their preferences and the directory agent will periodically contact data resource providers for updates and then notify user agents when new service capabilities come online to match user preferences (client data push). The latter interaction is provided through a FIPA Subscribe Interaction Protocol

Specification [1]. The directory agent supports the registration, deregistration and modification of registrations of agents and services. It also performs agent and service matches when queried. In addition to the above basic function the directory agent can periodically check for the existence of registered agents and update its internal directory about their status.

4.4.3.4 Ontology Services and the Resource Agent

The EDEN-IW system uses a multi-agent system to process and distribute the EDEN-IW semantic metadata models. In order for agents to function as metadata processors they must import Semantic data messages using a HTTP transport, then parse and verify them in order to process them. The EDEN-IW agents use HTTP to import Ontology documents from known locations. The Jena Semantic Web Java toolkit [4] is used to parse DAML+OIL messages to extract data of interest. Generally validation is not performed.

The mapping between different parts of the Ontology model that need to be related, and between the Ontology model and the non-ontological external database resource instances is not always simple. For example, the same data concepts may be structured quite differently between databases

It is sometimes not sufficient just to use XML namespaces to link terms between different Ontologies. Often some conversion or the use of some formulae may be needed to link terms; for example when converting one set of measurement units to another set of measurement units. For this reason, EDEN-IW implements Ontology mapping services, e.g., the EGV-to-LDV-IOW (Local Database View of the IOW database) mapping service. On start-up of the Resource Agent, it reads in and parses the global and local Ontologies and is able to translate terms between instances of the EGV and LDV-IOW Ontologies. This mapping uses graph theory and was implemented using Java.

4.4.3.5 Introducing a New Database Resource

There is a well-defined process to introducing a new IW database resource into the EDEN-IW system. This consists of following major steps:

Semantic Data Model instantiation:

1. Convert local database schema to ontology representation (OWL).

The database owner is responsible for the conversion of exported database schema into the OWL format. The building-up of database ontology has to comply with generic rules of global ontology: each table is modelled into a sub-class of class *Table* in the Global Ontology, where two object properties can be inherited, hasPrimaryKey and hasForeignKey. Each database column is modelled into a sub-property of data property *Column* and is associated with the table class where it belongs to. Constraints are added into key properties: The Cardinality constraint specifies the number of both primary and foreign key in the corresponding table; The SomeValueFrom constraint specifies a collection of columns that form a key relation for either primary or foreign keys in the table. The development of database ontology is conducted manually using OWL editor such as protégé 2000.

2. Expand the database ontology into Local Database View ontology.

The OWL representation of database schema needs to be further expanded to be able to be associated with the EGV ontology via semantic mappings. The development of a LDV ontology is performed by a local database owner and domain experts together. Semantic analysis and vocabulary mapping is a major step during development. Three types of mapping are defined in section 4.3.7 with respect to direct mapping, view mapping and value mapping. The concepts and properties identified as synonyms across he EGV and local database ontologies are mapped using direct mapping. Other local concepts with equivalent meaning of aggregated EGV concepts or a restricted EGV context are mapped using view mapping. The concepts without mapping relations are not accessible from global view. Interim concepts are created in LDV providing value mapping for both direct-mapped concepts with different value representation and instance mapping for view mapping relations. The process is conducted manually using protégé 2000.

3. Configure the local resource agent

A new resource agent is configured to wrap the new database for both data and metadata access. The configuration includes important information for local database access such as URL of LDV, international language coding, interaction protocols supported, and information for JDBC connection to real database. The configuration can be changed dynamically without changes to the existing system.

4. Run the new resource agent

When the new resource agent set up, it will automatically register with the directory agent, extract the key metadata content and load that into the directory agent metadata registry enabling the new database to be assessable via the global query interface.

4.5 Implementation and Validation

The EDEN-IW system is implemented as an open source Java agent platform called the Java Agent Development Environment or JADE [2]and a set of domain specific EDEN-IW application agents and non-agent software services, see Figure 24.

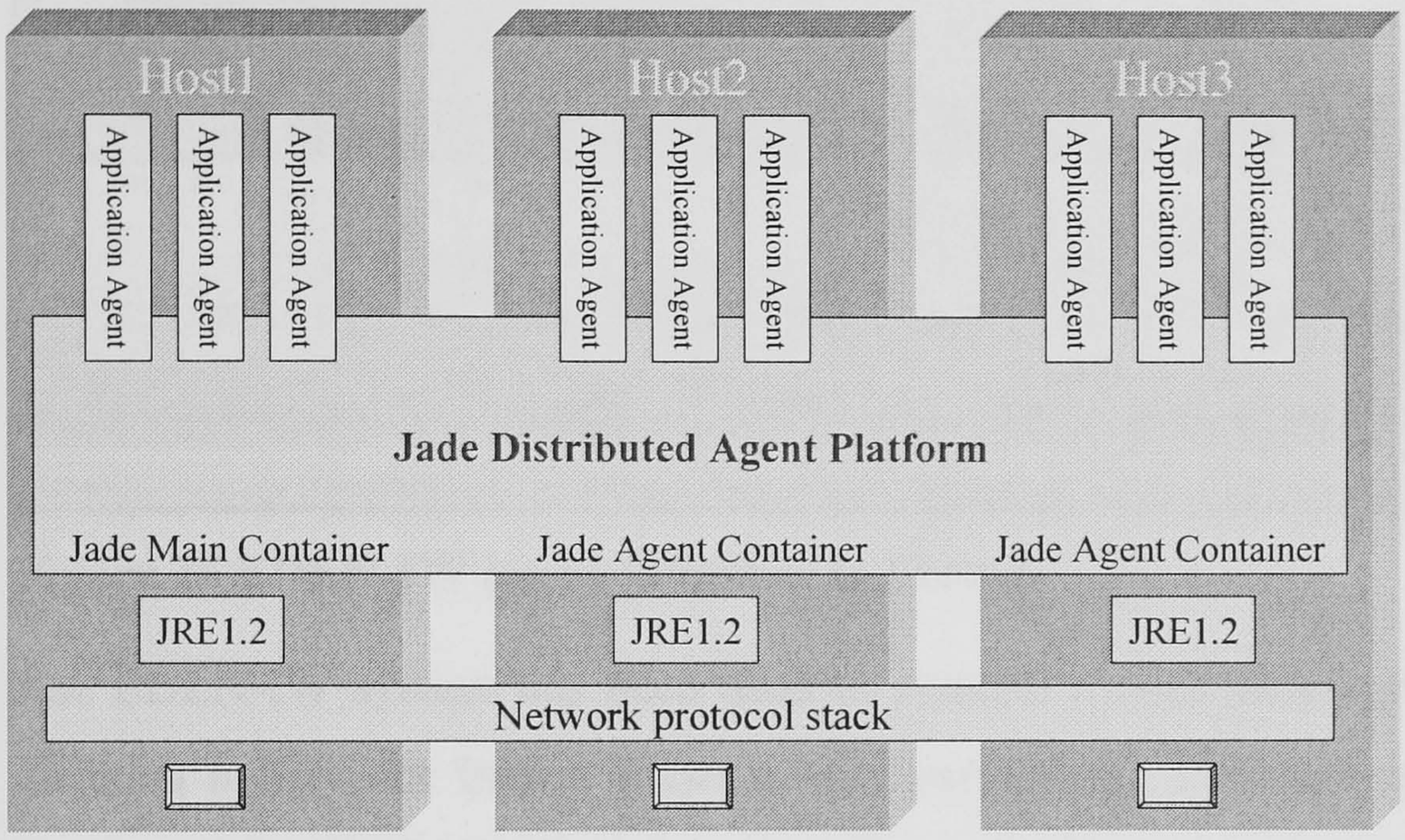

Figure 24 JADE Agent technology view of the EDEN-IW System

The JADE Agent platform provides the following core agent service: agent communication and Message Transport, Agent Name Service, Agent Directory Service and Agent Management Services, monitoring and configuration tools. To traditional software engineers, the agent technology appears as a set of distributed Java applications that are interlinked using a combination of Java RMI and XML messaging over an agent transport such as HTTP.

The EDEN-IW Java application agents run within the JADE distributed agent platform infrastructure. The agents exchange IW metadata in a common form called the EDEN Global terms or the EDEN Global View (EGV) terms - this insulates the majority of the agents and the user from needing to be familiar with the local database terms.

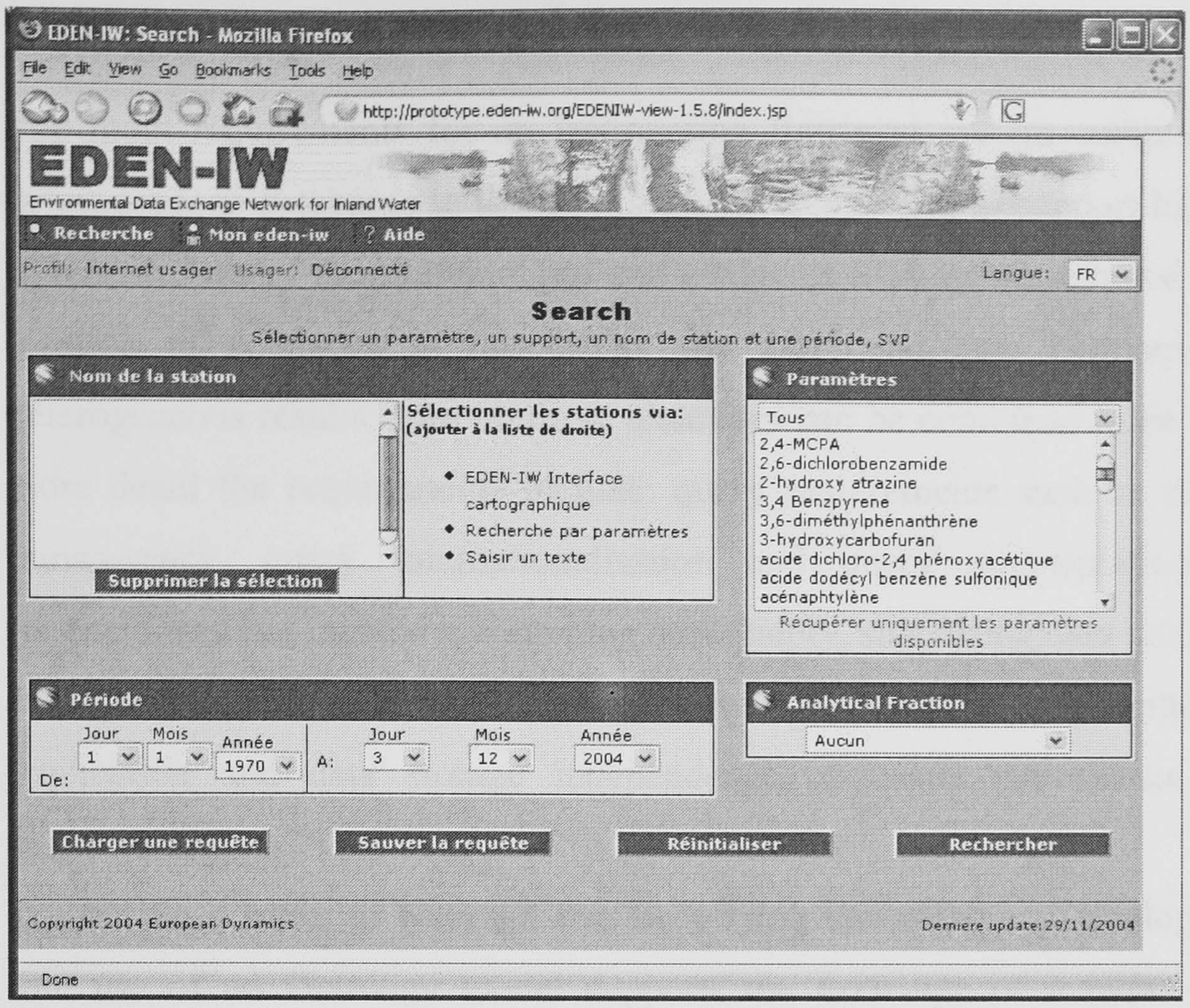

Figure 25 EDEN-IW query interface in French

The EDEN-IW system has been used to connect to four heterogeneous databases in different nations for Inland Water quality monitoring. The connected databases were developed in Oracle RDB, Oracle 9i, Microsoft Access, and SQL Server with more than one million measurement records. The semantic model supports the representation of domain knowledge in Danish, French and English. The system has been trialled and demonstrated to the environmental monitoring user community as several events and open days. Figure 25 shows the multi-lingual supported query interface of EDEN-IW system in French.

The EDEN-IW system is developed using a Rapid Application Development (RAD) software development approach that models the different system configurations from simpler core configurations to more complex. Several trial systems were built starting from the 'shallow' knowledge model with the subset of two connected databases, IOW and NERI. The validation and revision of application features were undertaken in an iterative process to test the approach performance in each sub-phase. The final integrated system consists of 4 different connected databases.

4.6 Summary

The main requirements for an Information Retrieval system composed of multiple heterogeneous databases in the Inland Water domain are to support high-level queries in terms of query transparency (no knowledge of SQL or the detailed data schema or location of databases is needed by the user) and data harmonisation (so that heterogeneous results from multiple databases can be combined to be meaningful). In more detail the requirements include: query requirements such as supporting query transparency, query internationalisation and query augmentation; data source requirements that include maintaining data quality, supporting data integration, flexible data presentation and data harmonisation; and metadata requirements to support application and data storage independence, metadata provenance and metadata restructuring.

To do this it seeks to leverage two underlying technologies: Ontologies and agents. The use of an Ontology model supports the exchange of machine-understandable structured data and supports automated processing and enhances information queries and information searches. There are several potential benefits in using a semantic approach to enable information integration. The main one being that interoperability is eased as there are likely to be fewer semantic than syntax differences and the semantic model can support the above requirements. The particular semantic approach used is based upon using a global schema called EGV that can be mapped to terms in the local data views, called LDV, using a global-as-view approach. The global schema not only provides descriptions of local data sources, it also provides a domain conceptualisation for the IW domain and a data dictionary of common synonyms to support provenance for term names and internationalisation. The semantic EGV and LDV models are expressed in DAML+OIL and created in an iterative process and require input from IW domain experts and database owners. The methods and processes for creating and accessing multi-lateral structure of the combined EGV and LDV model are given. A data query is posed using the terms of the EGV, the data sources that can answer the query are then identified from information in a metadata repository, and the queries are translated into LDV terms and then into SQL to be issued to the actual database services. The results of the query returned are translated into LDV terms and then into EGV terms for presentation to the user. The mapping from the EGV model to the LDV model is very complex as it must not only deal with direct mappings of terms but also

value mappings, when terms are part of different data structures, and composite terms or view based mappings, when some terms require multiple terms to be combined. This mapping is based upon Graph Theory and semantic routing.

Semantic services are needed to support the parsing and processing of the EGV model, for mapping to the LDV models and then to SQL commands to be issued to the data resource instances, and to distribute and coordinate the metadata. Hence, secondly, the EDEN-IW system is based upon a MAS model. However, the use of the MAS model gives added benefits, beyond providing a semantic data processing and distribution framework: agents can converse about the processing and meta-processing and can coordinate message exchange and the processing and reason about these. The Ontology based and agent based framework for information retrieval from heterogeneous databases has been implemented using the Jena Ontology system and JADE MAS system respectively and has been applied to four European heterogeneous databases for the IW domain, demonstrating the utility of the semantic approach to transparently query them and to harmonise their query results.

Chapter 5 A framework to Support Multiple User Views

In the previous chapter, a semantic framework was developed to support Information Retrieval (IR), more specifically to handle and mask, distributed relational database heterogeneities. It used a single domain conceptualisation with an associated single level of semantics to project a single information viewpoint of the individual database data and was targeted to answer core IW queries by domain experts and by scientist type users. This had the advantage that it avoided the necessity of these types of users needing to be familiar with the local-level viewpoints of the stored data schema and associated query interfaces based on relational data schema and SQL interfaces. Such an approach however, still has several main limitations. Users may not be familiar with that particular single information viewpoint in order to initially pose and to subsequently be able to optionally refine their queries under the strict constraints of a single conceptualisation. The high-level conceptualisation may also not adequately capture the operational nuances, side effects and errors in the results of queries. A more flexible approach is to be able to derive more targeted views, adapted for different types of users and applications. The development and application of a framework to support multiple user views of the data is the subject of this chapter. This research work has been undertaken as an extension to the EDEN-IW system to support customised information retrieval to environmental information.

An information viewpoint represents a given representation for some reality of interest, among the set of possible representations[89]. It reflects the understanding, classification and operations upon the domain knowledge pertaining to a particular usage. A viewpoint representation may contain conceptual information such as user terminology, conceptual structures and logic rules representing information interests at a specified level of abstraction.

Adaptation of retrieved information to produce customised information viewpoints is akin to matching service provider service descriptions or capabilities to user preferences. To facilitate this matching, a shared conceptualisation between the information provider and the information user is useful. A key design issue is how many combinations of user preference and provider capabilities are available in the match process, how the user preferences and provider capabilities are created and

imported and how a domain independent model that supports customised user viewpoints of the data can be constructed.

Generally, (Information) service providers publicly make available their descriptions and capabilities with respect to a finite set of dimensions that they think the typical users understand, e.g., in the IW domain, at a high–level these dimensions could typically be time, space and water quality indicator such as concentration of lead or mercury. But providers are often unable to understand or model the variability and range of usages at database design time. Hence user viewpoints vary even if the provider capabilities are fixed, as in the framework described in the previous chapter. To support generic adaptation, customisation along the dimensions of coverage, granularity and perspective is chosen [22]. To support and computationally constrain the options for domain specific adaptation, users are stereotyped in the IW domain to scientist, aggregator and policy maker. Adaptation becomes a matching process to orientate the queried data to the user preferences. We have already seen that this matching is complex. It most likely involves reasoning because of the various heterogeneities and the variety of mappings that must be handled. To some extent, the heterogeneities that are supported increase in the framework developed in this chapter as multiple abstractions and their conceptualisations associated with the different user viewpoints must also be handled.

With respect to the generic adaptation of information, *Coverage* identifies the user interest within a portion of domain knowledge. *Granularity* gives a level of abstraction covering the conceptual details of a user's understanding and representation. *Perspective* indicates a usage bias with respect to tailoring, evaluation and processing of the information. Each of these dimensions may be further distinguished as follows, illustrated using examples within the EDEN-IW domain:

- *Coverage:* describes a user's interests for a specified subset of the conceptual domain, e.g., in EDEN-IW, coverage could concern different clusters of observed parameters within a particular geographical area.
- *Granularity:* is divided into Concept and Processing Granularity
    - *Conceptual granularity* refers to super-concept and sub-concept relations according to classification hierarchies. The conceptual granularity is expressed in the form of IS-A and IS-PART relations in the semantic model,

e.g. FilteredOrganicMedium is part of total medium, and ammonia is a Nitrogen compound.

- *Processing granularity*: indicates the variation of information representation related to a particular level of analysis and processing. The relevant information is extracted from data sources via a specified analysis algorithm and processing functions, in order to fit certain user viewpoints. E.g., queries about the water quality need an appropriate interpretation of quality status in which the classification of mean or aggregated observation value is derived from the individual observations in the back-end data sources. The mean or aggregated water quality observations and individual records can be considered as representations of the same information set at different processing granularity levels.

- *Perspective*: is partitioned into interpretation model, information presentation and multi-lingual support.
  - *Interpretation model:* concerns the variation of terminology and relation definitions. This variation goes beyond synonym and homonym relations, in that it can involve a particular usage of the underlying knowledge , e.g. water quality can be classified into 6 grades that are measured by average observation of relevant determinands such as total phosphate and total nitrate, according to UK standards [92], whereas another European standard specifies another set of criteria for classification of quality status into 5 grades [41].
  - *Information presentation,* the query results of environmental information can be presented in various forms according to the user preference and intended usage.
  - *Multi-lingual:* the query interface and retrieved results can be presented w.r.t. a supported human language in written form.

The remainder of this chapter is organised as follows: Section 1 discusses the motivation for multiple viewpoints of environmental information retrieval within the context of EDEN-IW system. User preferences are analysed and viewpoints for specific user groups are identified. Section 2 specifies the functional requirements to support viewpoint adaptation and query transformation. Section 3 gives an overview of the method for viewpoint modelling and representation adaptation used. Section 4 presents some preliminary work on a formal approach to support user view queries and

the mapping to process them into back-end relational database queries. Section 6 describes the implementation details in the IW domain and describes the validation results of the method. The conclusion is given in section 6.

5.1 Motivation for Multiple View Support

In traditional IR systems involving relational databases, queries are performed w.r.t. a flat (relational) data schema representing the stored logical data structures of the database and the user requires knowledge of the schema terminology and constraints to form queries. The relational data schema are designed to separate the low-level physical structures of data on storage disks from logical structure of data access. However, the flat data storage schema does not capture the more object-oriented, higher level conceptualisation of a user's domain, e.g., the query "What's the worst polluted drain basin in England? " is difficult to answer, because it also involves domain knowledge that was not defined in the flat database schema.

Multiple viewpoints improve the usability of information retrieval by customising the knowledge representation to better suit the user requirement. However, such an open information viewpoint service isn't supported in traditional information systems because they are not designed to manage the dynamic adaptation of viewpoint representations and to support query refinement.

In the EDEN-IW system, different data sources are wrapped in conceptual models and integrated into a common Ontology, called EGV or the EDEN-IW Global View. Various user interests focus on different portions of the EGV and are represented using user-defined terminology, structure and relations in the associated viewpoint. The query posed on different viewpoints reflects the user's domain knowledge and relevant evaluation criteria that have been applied, for example the status classification of water quality can be classified in accordance with national standards and user stereotypes or user group. Three main types of user group are identified in IW domain, scientist, legislators or aggregators and policy makers. These user groups have been compared with respect to the coverage, perspective and granularity of conceptualisation they cover in Table 16.

The representation of individual user's expectation is associated with one of three user groups, with further adaptations to the coverage and perspective dimensions, e.g. water quality may be measured regarding biological, chemical and nutrients indexes. each sub-type classification may connect to observations of certain group of

determinand defined in common agreement of domain knowledge. These relations can vary dynamically according to the concrete roles of applied users. The translation of a query posed on a viewpoint is related to corresponding expressions in the common knowledge representation and compatible concepts and constraint relations. Adaptation of viewpoints may also conflict with the common agreement of knowledge representation that would mean any query to find common objects satisfying both representations always give an empty result. The dynamic changing of such roles and confliction rules makes management of viewpoint adaptation and query refinement even more difficult.

Table 16 User group classification

User Group	***Coverage***	***Perspective***	***Granularity Level***	***Featured query***	***Result presentation***
Scientist	*Data comparison to validate experiment hypothesis, e.g., variation of parameter w.r.t location and time*	*Individual observations in particular time and location*	*Detailed Concept, relation and properties*	*Concentration of Nitrite in river Y at time T?*	*Query result tables, Summary table.*
Legislators, aggregators	*General quality information for mean, aggregated observed determinands.*	*Aggregated or mean value in particular time period within observed area.*	*Summarisation of processed information.*	*Monthly mean concentration of total Nitrite in basin X of river Y?*	*Trends diagram, summary table*
Policy maker	*Quality grade to support further policy decisions.*	Change of water quality in various dimensions.	*High level report organised w.r.t. relevant criteria.*	*Trends of Nutrient pollution in river X for the last 3 years?*	*Trends diagrams, pie charts*

Individual user points represent instances relating to being a member of a user group. The usage may entail that users need to behave as members of multiple groups. An environment manager responsible for designing a River Basin Management Plan needs

fairly detailed data that could be associated with a scientist viewpoint but also needs to work within the constraints of water policies in the policy maker group. The individual user viewpoints may also need to be oriented to specific user constraints, rules and policies. These rules may relate to an organisational role that a user plays. Rules can be changed dynamically according to usage scenarios for that user. Here is an example of the use of user rules. An environment manager concerns about the variation of pollution effects with respect to land usage. In such a case, rules define the meaning of *pollution* in terms of certain chemistry indicators of given determinand groups. A user assessing the state of environment at national level will need data aggregated and water quality indicators at a higher level, e.g. monthly values and simple pollution levels for the major streams. This user will also need to access rules that specify the classification standards of status classification associated with a member of the policy-maker user group.

5.2 Requirements for Multiple User Views

Functional requirements have been derived for a system driven by the motivation and challenges highlighted above and grouped into viewpoint development, viewpoint management, query refinement and viewpoint presentation.

Viewpoint creation and representation

- User viewpoints can be created independently of the representation of the common Ontology model, using user-defined terminology, structure and relation and constrains.
- User preference information and user roles or user groups need to be modelled.
- Viewpoints need to adapt to user preferences and user groups.

User viewpoint management

- Detect and resolve conflicts between user viewpoint and common knowledge representation.
- Support evolution of user viewpoints.

Query adaptation to user viewpoints

- *Query construction:* queries can use a specific viewpoint conceptualisation.
- *Query augmentation*: queries can be expanded according to the user preference context and user role information.

- *Query mediation:* A high level general query may be decomposed into sub-queries available for data sources.
- *Query validation*: Conflicts in user queries should be detected and reported to the user, for example, an illegal query such as "discharge of fish" can be detected as semantically incorrect in that there is no semantic relation between discharge and fish..
- *Customised data presentation and processing*, result sets for user query can be oriented to the perspectives specified as part of user preferences.

The aims of multiple viewpoint support focus on adaptive viewpoint representation according to user's role and preference, to support user query refinement and transformation and traceable viewpoint evolution.

5.3 Computational Multiple User View Framework

A general framework has been developed to manage query transformation and presentation adaptation across multiple viewpoints based upon Semantic Web and in particular Description Logic or DL models and technologies. The semantics of user information viewpoints is explicitly defined in terms of a conceptualisation and its relation to data instances, within a scoped knowledge domain, using DL. Query transformation and representation adaptation can be automated and computed.

This framework supports the following properties:

- Multiple viewpoint vocabularies, categorisations and user stereotyping or groups.
- User preference and usage descriptions
- Viewpoint generation
- User queries to be answered in related to an associated user view.

When distributed data sources are integrated into a common conceptual model, how user viewpoints are modelled and where they are adapted from, need to be considered. User queries posed w.r.t. a particular viewpoint is aligned to the common domain conceptualisation. Conversely, the results of user query are also aligned to the user presentation preferences associated with the viewpoint. The approach given here focuses on viewpoint development, viewpoint management, query transformation and result adaptation in order to enhance usability of the information system.

The candidate's contribution is to investigate the flexible approach for viewpoint modelling and query adaptation over the stereotyped conceptualisation in a single knowledge domain. The formal theory of database view modelling and query answering approach [57] is the basis that the conceptual viewpoint has been extended from. There are also other inspiration such as schematic evolution approach supporting view adaptation [62], see section 5.3.2 and schematic operators [97], see section 5.4.4.3. The novelty of the proposed approach is the process of conceptualisation and terminology tailoring to support multiple user viewpoint and resolve semiotic heterogeneity. The approach supports an adaptive conceptual model with hierarchy structure and semantic constraints over the common ontology that can makes user query and viewpoint representation more flexible.

5.3.1 Design Issues

In the previous chapter a Global-As-View or GAV, that supports a global view that is derived from and can be expressed using the terms in the local databases, approach was used to support interoperability between heterogeneous databases. It is assumed that the global view remains static throughout query sessions. There is a single user view of the databases that uses the global view conceptualisation. This single conceptualisation in the user view is defined a priori to be consistent with the conceptualisation in the global view and it is assumed that the user understands this conceptualisation.

In this chapter, this method is enhanced to support the requirements for multiple views given above. Potentially, there are infinite numbers of arbitrary user views that can be projected from the global schema and these would lead to an infinite number of mappings between the user view and global view. In order to constrain the user view to global view mappings, only a limited number of stereotype user or group user views are supported that individual users of these groups are allowed to adapt in a finite set of ways. This also makes it easier to resolve any conceptualisations in the user view that are inconsistent with conceptualisations in the global view.

Forming multiple user virtual views of the multiple local data sources potentially represents a many to many mapping problem. A standard way to simplify the design of many-to-many data entity mappings in database modelling is to introduce an associative or mediating data entity to convert one many-to-many mapping into two mappings, a many-to-one and a one-to-many-mapping. The global view can be

considered as an intermediate data view in order to simplify the process to support many user views to many local data view mappings. Separating the user view to global view mapping, from the global view to the local view mapping, has the advantage that user views can be isolated from changes in the local data source semantic models and any changes in their mappings to the global view. Similarly, local data views can be isolated from changes in the user view semantic models and any changes in their mapping to the global view.

There are further design challenges to contend with that concern the differences between how the data is modelled and managed in relational databases versus how data is modelled and managed in the Ontological models. Typically, the expressivity and usage of constraints in a relational database model differs from their expressivity supported in an ontological model.

5.3.2 Modelling Stereotypes of Users or User Groups

User viewpoint modelling can be regarded as a process that adapts an existing viewpoint starting from the complete conceptualisation in the global view or EGV as the base user view. The process of creating a new user view consists of a sequence of primary steps applied in original view in order to manage redundancy, inconsistencies, concept omission and addition, multiple object classifications, structure and property changes and terminology and multi-lingual support in the user view.

McBrien and Poulovassilis [23, 62], have developed a bi-directional adaptive approach to handle schematic evolution in the context of database integration. Their approach is extended here to manage end-user viewpoint evolution. The viewpoint derivation process can be defined in terms of a unique adaptation path from a source viewpoint to a target viewpoint consisting of a sequence of operations. Suppose a target viewpoint V_t is derived from a source viewpoint V_s. The adaptation process can be transformed into a sequence of operations consisting of adding or removing concepts, properties and constraints. So that we have a normalised form of operation:

- $addConcept(C,(k_1,k_2...k_n)\,|\,o\,|\,v)$ indicates an addition operation for a new concept C into V_s with ambiguous attributes $(k_1,k_2...k_n)$. Argument o gives a operational view of $(k_1,k_2...k_n)$, where instance value of $(k_1,k_2...k_n)$ can be derived from V_s with a tag v indicating the type of operational view to be either complete, sound, exact or unknown.

- *add* Pr*operty*$(C.p \mid o \mid v)$ indicates an addition operation for a new property C.p into C in V_s. Argument o gives a operational view of C.p over V_s , where the instance value of p can be derived from V_s with a tag v indicating type of operational view to be either complete, sound, exact or unknown.
- *add* Re*lation*$(C.p, D \mid o \mid v)$, indicates an addition operation for a named relation C.p into C in V_s connected to a concept D. Argument o gives an operational view of C.p over V_s , where instance value of p can be derived from V_s with a tag v indicating the type of operational view to be either complete, sound, exact or unknown.
- *addConstra* int$(con, C(C.p) \mid r)$, indicates an addition operation for a new constraint *con* on concept C or relation C.p in V_s. Argument r gives a logic definition of r in the sense of C or C.o.
- *removeConcept*$(C, (k_1, k_2 ... k_n) \mid o \mid v)$, indicates a removal operation for concept C from V_s with ambiguous attributes $(k_1, k_2 ... k_n)$. Argument o gives an operational view of $(k_1, k_2 ... k_n)$, where instance values $(k_1, k_2 ... k_n)$ can be recovered from V_s with a tag v indicating the type of operational view is either complete, sound, exact or unknown.
- *remove* Pr*operty*$(C.p \mid o \mid v)$, indicates a removal operation for property C.p from concept C in V_s. Argument o gives a operational view of C.p over V_s , where instance value of p can be recovered from V_s with a tag v indicating the type of operational view is either complete, sound, exact or unknown.
- *remove* Re*lation*$(C.p, D \mid o \mid v)$, indicates a removal operation of relation C.p from C in V_s which is connected to concept D. Argument o gives an operational view of C.p over V_s , where instance value of p can be recovered from V_s with a tag v indicating he type of operational view is either complete, sound, exact or unknown.
- *removeConstra* int$(con, C(C.p) \mid r)$, indicates a removal operation for a new constraint *con* from concept C or relation C.p in V_s. Argument r gives a logic definition of r in the sense of C or C.o.

The process to derive a new group viewpoint from existing similar ones is conducted in an evolutionary style from a base viewpoint as it is decomposed into a sequence of atomic operations, given above, during the adaptation. In this case, the mapping relations can be reused at maximum level without the necessity of going through the developing process in a similar way to that described in Figure 26.

5.3.3 Modelling Individual Users

The representation of group viewpoints can be further tailored according to the particular demands of individual users, where preferences for conceptual representation are specified in an individual model separate from the group viewpoint. Each individual profile is associated with a certain user group, such that user preference can be interpreted correctly. The connection between a user profile and associated viewpoint is derived from key terminology concepts in the user profile and group viewpoint regarding synonym relations.

Individual user profile contains preference descriptions for individual users consisting of query preference, access permission, presentation preference, and user identification. A user is required to provide a concept set that relates to the view of the domain knowledge in terms of key concepts such as time, territorial area and inland water quality parameters, e.g. a concern about status of Nutrient quality in UK River B during time period C. Presentation preferences may also be specified in the individual user viewpoint to indicate the manner by which the retrieved results are presented in the user interface. User preferences for the presentation style, language support and possible standards for result classification can be specified.

User identification contains personal information for the individual user including identification of a user group or business role in the environmental organisation. Only pre-defined roles are recognised. The role identification contributes to the evaluation of appropriate rules in the viewpoint model. For example, a user profile for a French Policy Maker may contain the following information, see Table 17.

Table 17 User profile for a French Policy Maker

Attribute Name	*Value*
Business Role	Policy Maker
Associated Viewpoint	Policy Maker's Viewpoint
Access Permission	France territory
Preferred Language	French

Applied Classification Standard	EU

A reasoning process is carried out in the light of terminology similarity regarding the synonym relations given in an external glossary. The output of the reasoning is to identify a predefined group viewpoint that individual user can be associated with.

5.3.4 Rules for Individual Roles

Group user viewpoints may be further restricted by explicit rules that have been applied to the domain knowledge to specify the processing strategy and user preferences for information retrieval under certain circumstances. The explicit rules are specified in logic formulae. Explicit rules are specified in associated relations as dynamic constraints complementary to object properties. A user is allowed to specify their own perspective upon the domain knowledge by introducing a set of logic rules in the viewpoint Ontology. The interpretation of the viewpoint conceptualisation may further be adapted in compliance with the individual user's roles that are defined explicitly in a set of utilisation rules in addition to the viewpoint model.

The rules are of the form of an implication between an antecedent (body) and consequent (head). Rule has the form, *antecedent* $\Rightarrow$ *consequent*. The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold. Rule-based knowledge is considered as additions to a KR system of TBox (intensional data) and ABox (extensional data) such as OWL and some types of logical inferencing are not allowed, for example, the conjunction of two concepts implying a new class relationship.

Deployment of such rules in a viewpoint can further tailor the presentation of IR results and improve the information usability with respect to requirements for targeted user. Role-specific underlying knowledge is explicitly defined by rule specification that can be easily shared and reused across different user groups. Role-specific underlying knowledge is a set of supplementary expression of operational conventions and utility functions during the process of information analysis regarding particular information usages. User-specified terminologies are expanded regarding their query context and underlying knowledge in an explicit manner using rules such that the query expression can be adapted into an appropriate viewpoint for further IP processing. For example, the status of water quality may imply a classification of the average measurement for different types of aggregated observation in a specified time

period according to certain criteria. The classification standard may vary according to the concrete roles of the querier.

Rules are developed manually with the aid of domain experts and stored away from user profile in a separate base. Rule conditions are verified against the user profile content during the process of query answering and result adjustment, see Section 5.5.4.3. Other specifications such as preferred natural language and applied classification standards are also provided as optional information regarding the availability of user roles.

5.3.5 Mapping of User View to Database View

A user view of a database can be mapped to ontological group view and then to the databases themselves. The mapping between these conceptualisations is first given at a high-level in terms of an iterative set of procedural processes independent of the actual models or implementation of the mappings themselves. The mappings themselves can be implemented in an ad hoc way in terms of mapping functions and rules, or the mappings can be formally modelled. Whereas some processing and even reasoning about the rules can be performed in the ad hoc model, more rigorous reasoning to ascertain additional properties, e.g., about equivalence, soundness and completeness of models can be performed in a formal model, presented later in section 5.4.

The interpretation of a new instance of an IR query w.r.t. a conceptual model can be seen as a process to validate subsumption relations, to check if a concept in the user view is a subset of another data class, for some data instances in databases, they need to be interpreted as compositional expression of terms in the common Ontology in contrast to other data instances that do not require compositions. The data model for virtual integration used in the EDEN-IW system focuses on the semantics of data types rather than data instances. Each database table schema is regarded as intensional data (this equates to the TBox in the formal DL model) and is aligned to the common model which is a data type model. Extensional data (this equates to the ABox in the formal DL model) that relates to rows or data instances in the database are not aligned to the common Ontology model. Queries about the extensional data must be sent to the database to be answered. However, queries about the metadata could be handled by querying the common Ontology model.

Since the common Ontology model essentially wraps access to each of its data sources into conceptual structures, a conceptual viewpoint over that then provides a schema

with a derived conceptualisation, describing embedded data instances in the data sources. Relational schema can be conceptualised, akin to virtual tables in database views in the semantic model, acting as a semantic wrapper to a database and capturing named relations and constraints for the metadata related to the database schema.

The following mapping rules are used to facilitate conceptual operations over virtual table viewpoints to support reasoning about database schema and instances:

- Each relational table is mapped to a concept in the virtual table model and properties of the concept are mapped to attributes of a single column in the table such as datatype;
- Each foreign key constraint across tables in relational model is mapped to a virtual relation (object property) that joins different concepts.
- Constraints about the range of the domain of instance data value may be recorded in the virtual table to support reasoning about legal and illegal ranges or instance values, e.g., pH of water can't have a value 2 as this is too acidic to be water[5]

Under such assumptions, relational approaches such as LAV and GAV can be extended to support query answering over the virtual table conceptual model. In addition relations and constraints that express the semantics of hierarchy structures, e.g. generalisation and specialisation relations, aggregation and resolution relations and logic constraints on foreign key relations have been introduced into the framework.

5.3.6 The Mapping Process

The multi-viewpoint framework supports multi user viewpoints of a domain firstly in terms of the set of sub processes given in Figure 26, that seeks to find equivalences between the user view and the local data views via the common global view of the data:

1. *Synonym mapping*: To find any direct semantic correspondences between the user viewpoints concepts and the local data view concepts via the associated global Ontology concepts.
2. *Composite View Mapping*: To find any indirect semantic correspondences between concepts in the viewpoint terminology as a composition of concepts in the local data source Ontology via the global Ontology.

---

[5] Incidentally such values where found in database during tests of the system indicating incorrectly entered data or badly calibrated instruments.

3. *Containment and Integrity consistency check*: To check the containment or equivalence and consistency of concepts and their relations of the user query compared to those in the global view and then for query transformation to the query submitted to the database resource and the corresponding results.
4. *Application of user viewpoint rules*: To adapt the viewpoint representation to relate to instances of user groups using associated rules and knowledge, for example, using a specific instance of a water quality indicator, referring to a specific water classification specification or calculating derived values such as averages in a particular way.

The modelling process is conducted in an iterative style with a cycle consisting of sub-processes of synonym mapping, compositional mapping, consistency checks, conflict resolution, and viewpoint adaptation. The aim is to eliminate all possible semantic ambiguity and conceptual inconsistency between the common Ontology and viewpoint Ontology. For the case of unsolved consistencies, the system will simply report to the user about that the alternatives and allow the user to guide the system to make a choice to resolve the ambiguity.

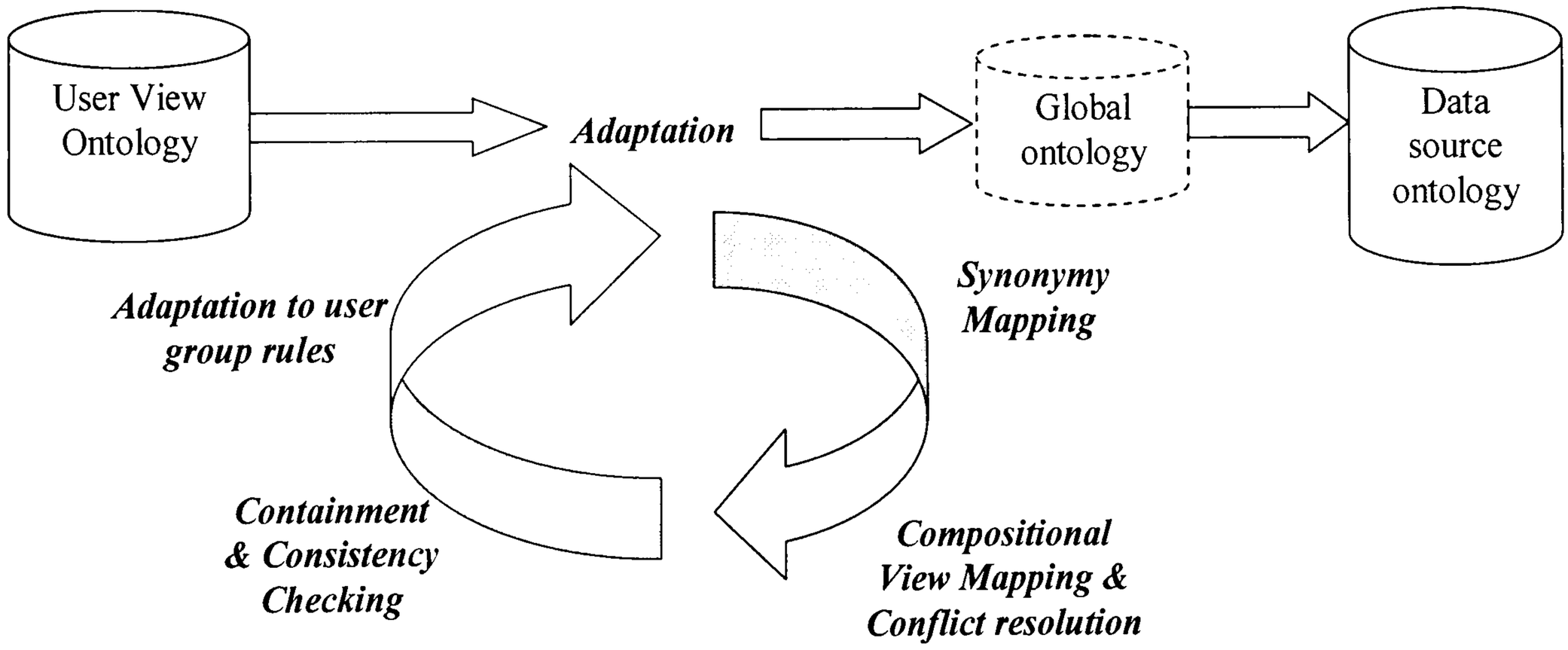

Figure 26 Ontology alignment of viewpoint conceptualisation

There is a similarity for this transformation to the one needed to transform the individual database views into a common Ontology conceptualisation or viewpoint, researched in chapter 4. In EDEN-IW, the conceptual constructs of the viewpoint can be derived from EGV either in a declarative form for process 1 or procedurally. Translation of a query corresponding to a particular viewpoint to the common

Ontology uses a set of mapping rules called rule unfolding to substitute query concepts and any constrains with corresponding ones in the common Ontology.

Synonym mapping focuses on conceptual alignment between Ontologies via consistent interpretation functions and semantic mappings, i.e. it seeks to find the corresponding expressions in the global Ontology that have the equivalent semantics and that subsequently result in a non-empty instance data sets retrieved from databases.

Terminology heterogeneity reflects different naming conventions amongst user groups. In EDEN-IW system, terminology heterogeneity also involves multi-lingual term usage. An independent terminology glossary has been developed on the basis of contents analyses of standard terminology sources such as EARTh and IOW thesauri [90] The main task is to build synonym equivalents amongst different term sources in different languages and involves comparing the meaning of different terms with their explicit definitions.

Compositional mapping in an ad hoc computation framework equates to the value mapping and view conversion described in section 4.3.7 in chapter 4 or can be modelled formally, see Section 5.4.5.1. An example of a compositional mapping is the reverse derivation of a *monthly-report* in EDEN-IW viewpoint of aggregator that is created by combining observations of determinands or even determinand groups and averaging these over time and space/

Consistency Check: A new constraint and rule may introduce unsatisfiability and inconsistency problem to a multiple viewpoint system, in such case end-user is prompted with the conflict and the rule is tagged to be ignored during relevant processing of query.

Following a hierarchical tree structure in a viewpoint Ontology:

- Child concepts will automatically inherit all properties, relations and constraints from parent concepts if it is applicable and no explicit declaration is necessarily required in child concepts.
- An instance query upon parent concept implies a thorough instance searching of all applicable child concepts including leaf concepts and non-leaf concepts.

Conflict Resolution: After a consistency check, any conceptual conflict and violation detected between the viewpoint and common Ontology need to be resolved before any viewpoint query can be evaluated with respect to a knowledge domain. In a multiple viewpoint system, suppose that adaptation of a conceptual structure and classification

in individual viewpoints is much more desirable because re-evaluation of the conceptualisation for the whole of the knowledge domain and related semantic mappings can be avoided.

The conceptual conflict and violation may appear in the following forms:

- Incompatible data types
- Contradictory hierarchical structures
- Cardinality constraints
- Other Conceptualisation conflicts.

The conceptualisation may be inconsistent because the mapping information was developed independent of any conceptual constraints upon the viewpoint constructs. When a viewpoint query is aligned to a common Ontology, constraints in the viewpoint need to be translated into the representation of the common Ontology, where conflicts may occur. Conflicts in the viewpoint conceptualisation are either removed or reconciled according to a common model. To this extent, the viewpoint is considered a more restrict conceptualisation on the basis of their semantic correspondence in the common Ontology.

5.4 A Formal Framework to Support Multiple Views

It would be useful if the semantic model for IR from databases could incorporate and formalise the relational operations used in IR. This would be given the added expressivity of not just reasoning about direct concept name mappings from database terms to EGV terms but also supporting meta-reasoning about the relational algebra used to retrieve database data. For example, the model can reason about the fact that there is redundancy in the relational model and different relations such as a certain join and a certain where clause in SQL selections are equivalent and give the same result. Also table relations in the database that are actually a subsumption relationship can be explicitly tagged as so in the semantic model.

The following framework, in contrast to the one described in the previous chapter, models the conceptual semantics of database integration system more formally and enables conceptual viewpoints to be created and managed over a common Ontology model.

5.4.1 Design Issues

A formal semantic model that supports multiple user views over multiple local data via a global view needs to handle:

1. Mappings between user view to data source via global view
2. Data operations on properties and constraints of concepts in the user and global data view models and reasoning about these.
3. Data operations on properties and constraints of concepts in the local data views and reasoning about these.
4. Combining 3 and 4 so that data operations in the user and global view lead to equivalent data operations in the local view and reasoning about any knowledge loss when mapping between 3 and 4.

Note also that for point 2 above, the global and user conceptual model, its constraints, operations and reasoning are fundamentally different from those for 3. These are summarised in Table 18. These differences need to be handled, and there are several main approaches to do this.

Table 18 Difference between semantic global view models and database models

	Semantic model	***Database model***
Type of concepts	Classes and properties of classes, instances	Tables, table attributes or columns, instance values of attributes
Data operations	Conceptual and logic operations	Set of relational operations
Class based hierarchy and relationships for classes of concepts and properties	Yes	No
Flat or point to relations between concepts independent of hierarchy	.Yes, compositions of classes in different hierarchies and heir instances	Yes, 1-1, 1-many or many-many relationships between attributes Functional dependencies between attributes.
Constraints	Classes, properties of classes	Integrity based: entity and referential
Derived data	Logical entailment	Preset arithmetic data functions
Closed versus open world	Open world: absence of data indicates information is unknown	Closed world: absences of data indicates a negative information
Multiple views of data	Yes, can instantiate parts of	Yes, derived data

supported	class hierarchies and compositions of classes	
Consistent conceptual model	Can have inconsistencies but can be designed to be consistent across different data stores using alignment and merging	Is designed and maintained to be consistent within a databases but across databases, consistencies can occur

There are several possible ways to work with both logic, semantic and database models:

1. To keep logic and semantic models relatively separate with specific point to point mappings between them but no structural or semantic equivalence between them
2. Use a logic for relational model such as a Datalog algebra that is extended to support a logic for semantics
3. Use a logic for semantics such as description logic that is extended to support database relational support
4. Use a logic or some bridging logic theory to combine description logic and database logic

5.4.2Viewpoint Model

Suppose, we have a system that supports a system $A = \{V, G, M\}$ with multiple viewpoints, where $V = \{v_1, v_2 ... v_n\}$ is a set of viewpoints where each element $v_i, 0 < i < n$ indicates a unique conceptual viewpoint. G is a common conceptual representation within a single domain, where atomic terminologies are defined in the conceptualisation as primary building blocks and further decomposition of terminologies is not allowed in G. $M = \{\vec{m}, \vec{p}\}$. is the union of semantic mapping $\vec{m}$ and adaptation path $\vec{p}$. $\vec{m}$ is a vector of semantic mapping, $\vec{m} = \{m_1, m_2 ... m_n\}$, where each element $m_i, 0 < i < n$, indicates a set of mapping from v_i to G in a term of $q^{Vi} \to q^G$, indicating a query over viewpoint is equivalent to a semantically corresponding query over G. $\vec{p}$ is a vector containing adaptation paths to establish the viewpoint V from G. Each element p_i , o<i<n, indicates a unique operation sequence over the common Ontology where v_i is derived.

A viewpoint model v_i conforms to a conceptual interpretation I_v over a knowledge domain Δ , denoted to be Δ^{IV} , whereas G gives another interpretation over knowledge

domain Δ, denoted as Δ^{IG}. Both domains D_v and D_g can be represented in virtual databases. It is supposed that $\Delta^{IV} \subseteq \Delta^{IG}$. G is a valid model of D_g, if D_g satisfies all relations and constraints of G via conceptualisation mapping as described in section 0. A database D_v is said to be legal or logically consistent with respect to D_g, if

- D_{Vi} satisfies all constraints in v_i
- D_{Vi} satisfies mapping M_{Vi} to G with respect to D_g

A non-empty tuple set T_{Vi} in a legal database DV corresponding to viewpoint V, associated with a semantic mapping of M_{Vi} from v_i to G and its non-empty semantic correspondence, a tuple set Tg of DG, constitutes a valid interpretation of semantics $A = \{v_i, G, m_i\}$, where Tg is a so-called valid interpretation of v_i via approach m_i with respect to DG.

Thereafter, the formal semantics of queries over multiple viewpoint system can be formalised. A user query q posed on viewpoint v_i is expressed using terminology set of v_i with respect to virtual database D_{Vi}. The answer set of query q in D_g is denoted as q^{DG}. The answer set of query q in D_{Vi} is denoted with q^{DVi}. The evaluation of query q over system A={V,G,M} is such that each tuple $t \in q^{DG}$ is validated against q^{DVi}, to check if $t \in q^{DVi}$. A non-recursive datalog query is defined by: $Q(\vec{x}) \leftarrow conj_1(\vec{x}, \vec{y}_1) \vee conj_2(\vec{x}, \vec{y}_2) ... \vee conj_m(\vec{x}, \vec{y}_m)$ Each conjunctive query $conj_i(\vec{x}_i, \vec{y}_i)$ is a logic predicate indicating a sub-query or sub-goal of query Q, where $\vec{x}$ is a variable vector of Q over v_i with arity n that is also a union of subset attributes $\vec{x} = \{x_1, x_2 ... x_n\}$. The variable set of query Q is denoted as VAR(Q)= $\vec{x}$. $\vec{y}$ is consists of attributes v_i or constant via comparison predicates to $\vec{x}$. $\vec{y}$ is a union of attribute and constant subsets $\vec{y} = \{\vec{y}_1, \vec{y}_2 ... \vec{y}_m\}$. Query Q in viewpoint D_{Vi} is a process to find set of n-tuple constants $c = \{c_1, c_2 ... c_n\}$ satisfying D_{Vi}. If we substitute value of $\vec{x}$ with c, the evaluation of following predicate gives a true result.

$\exists \vec{y} \mid conj_1(\vec{x}, \vec{y}_1) \vee conj_1(\vec{x}, \vec{y}_2) ... \vee conj_m(\vec{x}, \vec{y}_m)$.

Hence we can define the conditional and necessary relation for *containment mapping* between two query expressions. Query Q1 is contained in query Q2. if and only if there is a mapping relation *m* that maps VAR(Q1) to VAR(Q2) and each sub-goal of

Q1 has a corresponding sub-goal in Q2 regarding *m*. If Q1 and Q2 has a containment mapping and every result of Q1 is contained in result of Q2 for any database extension according to a schema S, we can say Q1 is contained by Q2. Q1 is equivalent to Q2 if and only if Q1 contains Q2 and Q2 contains Q1.

Maximally-contained query, Query Q' is a maximum containment equivalent of original query Q on any database D using V{v1,v1…vn}, if

Q'(v1,v2…vn) $\subseteq$ Q(D) There is no such query Q'' that adhered to condition Q'(v1,v2…vn) $\subseteq$Q''(v1,v2..vn)$\subseteq$Q

5.4.3 Viewpoint Conceptualisation and Semantic Mapping

Semantic mapping is a core technique in the semantic conceptualisation system A={V,G,M} used during the process of query answering over viewpoints.

In traditional methods for database integration, view-based approach such as LAV and GAV define a set of mapping assertions for relational tables between global schema and local schemas that have the form $g \to q_s$ (GAV) and $s \to q_g$ (LAV) where g is an element in a global schema; q_s is the corresponding query expression for g in local database terms; s is an element in local data and q_g indicates its corresponding query expression in global schema terms.

A View-based approach enables explicit semantics to be used to evaluate a corresponding query and to retrieve the corresponding results from a collection of databases The approach introduces a user-oriented, independent, viewpoint of the domain knowledge that is referred to as a TBox Ontology $V = (T_v, R_v, C_v)$, where viewpoint V is constituted with terminology set T_v, a unary or binary relation set R_v and constraints set C_v that is associated with T_v and R_v. The Semantic meaning of the viewpoint is indexed via an interpretation I of the ontological terms T_v and R_v into a non-empty domain knowledge set Δ^I, where $T_v^I \subseteq \Delta^I$, and $R_v^I \subseteq \Delta^I \times \Delta^I$. As described previously in chapter 4, the mapping information has to be developed to connect semantic correspondences between an end-user viewpoint V and a common Ontology model $G = (T_g, R_g, C_g)$. If G is regarded as a primary viewpoint V_g of conceptual terms in domain Δ, a viewpoint V defines a compositional terminology and a relation set of domain knowledge with additional constraints and rule specification upon V_g

that is similar to the GAV model used in a conventional data schema integration approach. The indexes of terminology T_{vi} can be expressed in a conjunctive logic form with equivalent relations such as $T_{vi}^{\Delta^I} = (disjunc_{i1} \wedge disjunc_{i2} ... disjunc_{in})^{\Delta}$, that specifies the interpretation I of terminology T_{vi} regarding domain Δ as an ***exact view*** of the right-hand logic expression over Δ, where each element of the conjunctive expression $disjunc_i, 0 < i < n$ is a disjunction of terms T_g, R_g with value or existential quantification. Following conditions must be held,

$$\forall t \mid t \in T_{Vi}^{\Delta^I} \rightarrow t \in (disjunc_{i1} \wedge disjunc_{i2} ... disjunc_{in})$$

and $\forall t \mid t \in (disjunc_{i1} \wedge disjunc_{i2} ... disjunc_{in})^{\Delta} \rightarrow t \in T_{Vi}^{\Delta}$.

Additionally, we have equivalent relations for relation in an exact-view mapping: $R_{Vi}(a,b)^{\Delta} = R_g(a^{'}, b^{'})^{\Delta}$ iff, $R_{vi}.a^{\Delta} = R_{gj}.a^{'\Delta} \wedge R_{vi}.b^{\Delta} = R_{gj}.b^{'\Delta}$, where R_{Vi} is a relation in V_i and a,b are concepts associated to it , R_{gj} is a relation in G with associated concept $a^{'}$ and $b^{'}$. This specifies that an exact-view mapping between two conceptual relations R_{Vi} and R_g w.r.t a viewpoint interpretation domain Δ can be established if and only if there are exact-view mappings between their corresponding properties over viewpoint representation.

Regarding other consumption relations between T_v, R_v and T_g, R_g . The above relations can also be applied with subset and superset relations, where $T_{vi}^{\Delta^I} \subseteq (disjunc_{i1} \wedge disjunc_{i2} ... disjunc_{in})^{\Delta}$ and $R_{Vi}(a,b)^{\Delta^I} \subseteq R_{gj}(a^{'}, b^{'})^{\Delta}$ are named as a ***complete view*** mapping from V to V_g. This indicates that the interpretation of left side over knowledge domain is a superset of its view correspondence on right-side expression, i.e. any instance that satisfies left-side expression must satisfies right-side expression over domain Δ. Similarly, a sound view defines a subset relation.

$T_{vi}^{\Delta^I} \supseteq (disjunc_{i1} \wedge disjunc_{i2} ... disjunc_{in})^{\Delta}$ where $R_{Vi}(a,b)^{\Delta^I} \supseteq R_{gj}(a^{'}, b^{'})^{\Delta}$ are named as a ***complete view*** from V to V_g.

5.4.4 Conceptual Operations

Viewpoint conceptualisation can be derived from other viewpoint or from the common Ontology model via a sequence of primary operations. Primary operations include:

- Terminology and relational operations: rename (suitable for multi-lingual support). selection, projection, natural join, rename, composition and decomposition, union, aggregation
- Granularity operations: building-up of concepts in hierarchy structures with rules and operations such as generalisation, e.g., deriving water quality indicators from individual observations and specification, composition and decomposition, e.g., converting catchments to sets of stations. It can also involve value expressions such as to split or merge data values.

Adaptation steps applied in the Ontology model are categorised in terms of Relational operations, Hierarchical operations and Attribute and instance value operations

5.4.4.1 Relational Operations

Since the conceptual model is considered as an extension of the relational model, all standard relational operator can be reused in the Ontology for the retrieval of data instances. If we align a relational structure with a conceptual structure, we have each table mapped to one or more individual concepts, each foreign key constraint is a relation (object property in OWL), each tuple in a specific table is modelled as an instance of the corresponding concept in the Ontology. In this situation, the relational operators Selection σ, Projection π, and Rename ρ can be reused at the level of intra-concept operations, other operators Union ∪, Intersection ∩, Difference-, Natural Join ∞ can be used as inter-concept operator at the same granularity level.
Some other algebra operations are defined as Marco operations of relational algebra in order to simplify the operational expression, for example, the path join operation.

Path Join operation is defined in a form of $\kappa_{x,y}(R_1, R_2 ... R_n)$, indicating a specific join path across relational tables, where R indicates joinable relations via foreign key constraints, x is an attribute set of R_n, y is an attribute set of R_1, that have arity of x equal to the arity of y. The operational meaning of Path Join is
$\kappa_{x,y}(R_1, R_2 ... R_n) = \rho_{x \to y}(\pi_{(R_1 - y),x}(R_1 \triangleright\!\triangleleft R_2 ... \triangleright\!\triangleleft R_n))$, indicating that all tuple sets of y in R_1 are substituted by the corresponding tuple set of x in Rn to form a new relational table.

5.4.4.2 Hierarchical Conceptualisation Operator

A relational model is a flat data model without a hierarchical structure and type classification in its conceptualisation. Conceptual operators are seen as additions to the relational operators to form a formal Ontology model. Relational operators can not handle hierarchy operations across different granularity levels, e.g. relations of generalisation and aggregation. A relational operator of union, A = B ∪ C, which indicates a process to combine all tuple sets belonging to B and C into a tuple set of A. On this basis, conceptual operators of aggregation and generalisation are introduced representing the semantics of HAS-A and IS-A relations. The aggregation defines the combination relations between concepts from both the instance and abstract perspectives. For example, A is an aggregation of B and C, which refers to the abstract level such that A combines all attributes of B and C and at the instance level such that an instance (tuple) of A contains an instance of B and an instance of C. Generalisation defines an extraction of a new concept or relation at a higher granularity level with given concepts (relations). A common set of attributes is retained in the generated concept and relations. An instance set of given concepts or relations become a subset of generated concepts or relations. For example, a river is an aggregation of all relevant catchments, whilst concept *Waterbody* is a generalisation of concept *Lake* and Concept *River*.

Operator ***aggregation*** is denoted by $C = \alpha_{A,B}$, where arguments A and B are given concepts with associated attributes where $A = \{a_1, a_2 ... a_n\}$ and $B = \{b_1, b_2 ... b_m\}$. The concept $C = \{c_1, c_2 ... c_n\}$ is an aggregation of A and B, iff the following condition holds:

$\forall a_j \in A, \forall b_k \in B \rightarrow a_j \in C \wedge b_k \in C$,

and $\exists c \in CT, \forall a \in AT \mid a \subseteq c$ and $\exists c \in CT, \forall b \in BT \mid b \subseteq c$

Where, CT is instance set of C, AT is instance set of A and BT is instance of B. a,b are individual instances in respect with A and B. The operators specify the semantics of the aggregation operation at both the terminology and instance levels: every attribute must be present in the aggregated concept, every instance of the argument concept must be a contained in a corresponding instance of the aggregated concept.

Operator *generalisation* has the form $C = \varsigma_A$ indicating a super concept relation in hierarchical structure where A is a given concept with associated attribute set

$A = \{a_1, a_2 ... a_m\}$. The concept $C = \{c_1, c_2 ... c_n\}$ is generalised from of A. It is necessary if the following condition holds for attributes: $\forall c_i \in C, 0 < i < n \mid c_i \in A$

and for instance set CT of concept C and instance set AT of concept A, it has, $\forall a_i \in A \rightarrow a_i \in C$. Operator specialisation and resolution can be defined as inverse operators with respect to generalisation and aggregation.

Operator *specialisation* is defined as $\varepsilon_{\vec{x}}(C) = C^{'}$ indicating a sub class relation in hierarchical structure, where $C = (c_1, c_2 ... c_n)$ and $C^{'} = (c_1, c_2 ... c_n, \vec{X})$, $C^{'}$ is a sub-class of C, $\vec{X}$ is set of additional attributes of C in $C^{'}$. For any data instance $t^{'} \in C^{'}$, there is a instance $t \in C$, we have $t = t \cap t^{'}$.

Operator *resolution* is defined as $\gamma_{rule}(C) = \{CR_1, CR_2 ... CR_n\}$. An attribute set of concept C has been decomposed into sub-groups to form corresponding concepts. For all instance value TCR_i of a concept $CR_i, 0 < i < n$ and TC standing for instance set of C, there is a relation $TCR_i = TC_{\pi(\vec{x})}$, where $\vec{X}$ is the attributes for CR_i.

Functional dependency

Concept A is functionally dependent upon Concept B, $A = B_{fx}$ if there is a functional morphism relation *func(x)* that maps instance set of A to instance set of B, i.e. each instance of A can be mapped to a subset instances of B via a unique processing function *func(x)*.

Instance Merging

If each instance value of one concept is functionally dependent upon a set of instance value of another concept satisfying with a given logic condition, it is called *instance merging* . The Instance merging operator is defined as $C = \omega_{funct(x),cond}(C^{'})$ that indicates a merging operation for attribute value x of $C^{'}$ to attribute y of concept C through $func(x)$ satisfying condition *con*. Condition *con* is a conditional filter for instance of C. Only satisfied instances are taken into account.We can not define an inverse operator such as instance splitting as not all dependent functions may have an inverse function.

5.4.4.3 Attribute and Instance Value Operator

Other operators are defined to extend the operational expressivity for concept instance and attribute value set manipulation: Skolemisation φ, Composition λ and Decomposition γ.

Skolemisation operator φ

When an existential quantifier is within the scope of a universal quantifier, the quantified variable can be replaced with a Skolem function of the universally quantified variables in order to simplify the computation. While in the formula,

$\forall x \exists y R(x, y) \Leftrightarrow \exists f \forall x R(x, f(x))$ indicates that the choice of a value for X is dependent on the choice of a value for Y since the formula asserts that for each Y there is an appropriate value for X. In this case, the variable X would be replaced with a Skolem function of y=f(x) . The formula can be presented in the form of $F = \forall x_1 \forall x_2 ... \forall x_n \exists y, R(x_1, x_2 ... x_n, y)$, hence, there is a $y = f(x_1, x_2 ... x_n)$. If we have attribute vector $\vec{x} = \{x_1, x_2 ... x_n\}$, $y = f(\vec{x})$. The above definition can be adopted for an attribute operation for a concept. The Skolimisation operator is defined as $A = \varphi_f(\vec{X})$, where A is a generated attribute in concept C from an attribute set $\vec{X} \subseteq C$, hence $C(\vec{X}, \vec{Y})$=C(A, $\vec{Y}$).

Composition λ

Operator composition is a special form $A = \gamma_{order}(\vec{X}, c)$ of the Skolem operator where instance value t of generated attribute A equals to the concatenation of instances value for $\vec{X}$ and relevant constant c in a specified order. λ is used to express heterogeneous expression conventions such as full address, full name and time clock.

Decomposition γ

Operator γ is an inverse operation of composition λ in a form of $(\vec{X}, \vec{C}) = \gamma_{order}(A)$, where attribute A is decomposed to a attribute set $\vec{X}$ and a constant set $\vec{C}$ in a specified order.

A group viewpoint represents a typical understanding of conceptualisation of a particular user group, e.g. scientist, national policy maker or EU risk manager. The interpretation of such an Ontology model may further vary according to the individual user. For example, the classification policy for water status may vary in different counties. The individual user is specified in terms of additional constraints or rules upon group viewpoints. User-defined rules can represent an underlying convention and the user preferred understanding and processing during information retrieval.

5.4.5 Use of Logical Operators

Logic operators are useful in order to handle the processes for the adaptation process that takes a query expressed in a user view and maps it to one that can be submitted to the local databases. The synonym mapping is given in an external glossary.

5.4.5.1 Compositional Mapping

The terminologies of a viewpoint V can be modelled in a form of compositional terminologies using atomic terminologies in a common Ontology model G and be represented in a semantic mapping of M. Terminologies defined in G become a primary set of conceptual building blocks consisting of atomic concepts (unary predicates), atomic roles (binary predicates) and individuals (constants). The targeted aim of setting up a compositional terminology in the viewpoint model is to structure viewpoint conceptualisation in compliance with common conceptualisation in formal semantics according to subsumption relations, which can be used for reasoning about further expression transformation.

Suppose, each concept, property and relation in viewpoint V is defined in atomic terms in a common Ontology with atomic logical operators such as union, intersection, negation, universal and existential quantifier and comparative operators. The semantic definition of terminology normally has the form

$C \subseteq D$ ($R \subseteq S$), $C \supseteq D$ ($R \supseteq S$), or $C \equiv D$ ($R \equiv S$) regarding the interpretation upon knowledge domain, where C is a concept or property in the viewpoint, R is a relation in the viewpoint, D is a compositional expression of atomic concepts or properties of the common Ontology and S represents a relation in the common Ontology.

There are three types of mapping axioms denoting the consumption relations of interpretations between viewpoint terminologies and their semantic correspondences within a defined knowledge domain. Axioms of first type refer to a complete view according to section 5.4.3. Similarly, Axioms of the second or third type refer to a sound or exact view.

In realising concept adaptation such as adapting queries to user types, common conceptualisation and user preference, although specific individual data functions as the user query may be expressed declaratively in description logic or its corresponding conceptual algebra as defined in section 5.4.4, this may trigger a process or process orchestration to execute a function such as mapping that is often better expressed procedurally but can be related to the declarative form.

e.g. a property *parameter* of concept *monthly-report* in EDEN-IW viewpoint of aggregator may be defined in procedural manner:

MothlyRpt.parameter ≡

$$\rho_{\det er \min andName \to parameter}(\pi_{\det er \min andName}(\sigma_{observationTime \subseteq monthlyRpt.time}$$
$$(Observation \triangleright\triangleleft \det er \min and \triangleright\triangleleft \det er \min andGroup))) \cup \rho_{groupName \to parameter}$$
$$(\pi_{groupName}(\sigma_{observationTime \subseteq monthlyRpt.time}(Observation \triangleright\triangleleft \det er \min and \triangleright\triangleleft \det er \min andGroup)))$$

or in declarative manner:

MothlyRpt.parameter ≡

$$\{P \mid \exists O \in Observation, \exists D \in Deter \min and, \exists G \in Deter \min andGroup$$
$$(O.\det er \min andID = D.\det er \min andID \wedge D.\det er \min andID = G.\det er \min andID \wedge$$
$$(P.parameterName = D.\det er \min andName) \vee (P.parameterName = G.\det er \min andGroupNa$$

5.4.5.2 Consistency Checking

An end-user viewpoint $V = (T_v, R_v, C_v)$ upon EGV $G = (T_g, R_g, C_g)$ contains a set of user-defined constraints and rules C_v that gives user-defined restriction upon T_v and R_v.

Constraints C_v need to be validated before it can be applied in viewpoint V in order to solve logic conflictions between C_v and C_g. The confliction may lead to an illegal interpretation to a knowledge domain across viewpoints; hence a proper query answer may be reduced or empty. The detection of logic conflicts involves a reasoning process across different viewpoints via given mapping relations. The conflict reasoning can be conducted in the following way in an adaptive viewpoint system A=(V,G,M):

- ***Satisfiability***: A concept C of viewpoint V is satisfiable with respect to V over V_g if there is an interpretation model I_v of C via mapping relation M to the semantic correspondence C_g in V_g, such that C_{Iv} is nonempty over domain Δ. In this case we say also that I_v is a model of C over Δ.
- ***Consistency***: A new rule r in viewpoint V and its interpretation rule r_g are consistent with existing system A=(V,G,M) if the evaluation result holds false for $\neg r \wedge V$ and $\neg r_g \wedge V_g$.

A new constraint and rule may introduce unsatisfiability and inconsistency into a multiple viewpoint system. In such a case, an end-user is prompted with the conflict and the rule is tagged to be ignored during relevant processing of query. OWL-DL has been chosen to be the representation language for viewpoint modelling. Hence the conceptual satisfiablity and constraint consistency between views can be processed using powerful logic algorithms upon Ontology constructs.

5.4.6 View-based Query Answering and Result Adjustment

User queries posed on viewpoint system need to be evaluated over the common Ontology in order to get the result sets from the underlying data sources. The process of query evaluation needs to reason about the containment relations between result sets over the common Ontology with respect to the initial viewpoint. The maximum-contained answers are computed semantically in order to find a proper expression over the Ontology that can be executed further in distributed databases.

The process of query answering using views is divided into sub-processes and performed in order as described in Figure 27. The process starts when a query is constructed in user's terminologies and associated with a selected viewpoint model, where synonymy and multi-lingual terms are translated according to the synonymy glossary and multi-lingual thesaurus that are developed and maintained independently and commonly used within the knowledge domain.

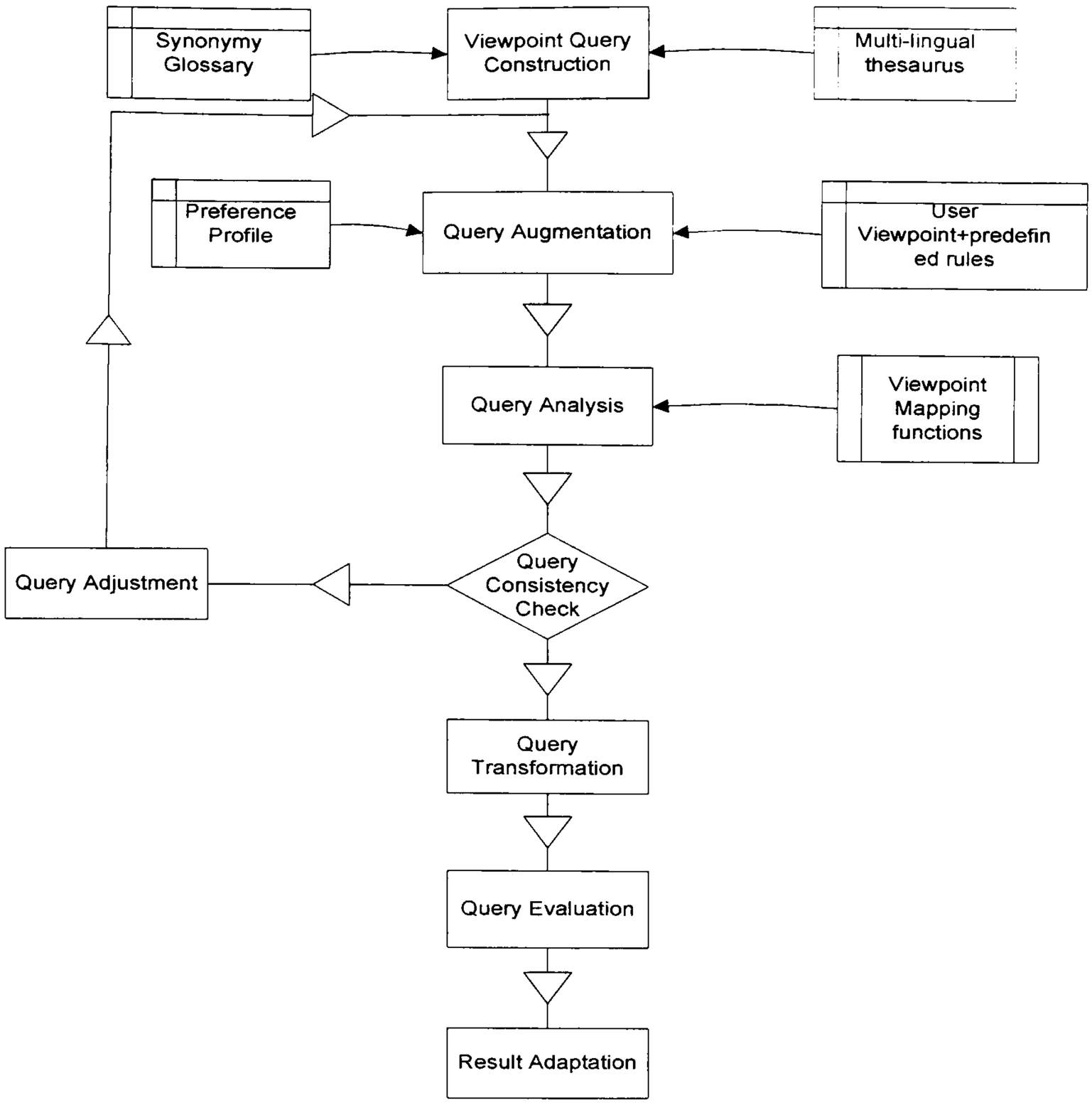

Figure 27 Query answering and result adjustment of viewpoint query

The query is further expanded using the underlying knowledge and default values defined in the user profile and role-specified rules, such that, a user query is fully expressed and interpreted according to the conceptualisation of the associated viewpoint. Thereafter, the semantics of a user query is analysed and mapped into equivalent or contained query expression in terms of a common Ontology with mapping relations computed via TBox reasoning. The transformed query is validated against viewpoint model and common Ontology to ensure conceptual consistency, so that theoretically evaluation of such a query would produce non-empty results set over multiple viewpoint system. When a query returns results from the database systems, the results need to be checked if they satisfy the viewpoint conceptual constraints. Any necessary adjustments are carried out in compliance with the viewpoint conceptualisation, user preference and role-specified rules.

Containment reasoning for the query expression is a core technique for query transformation based upon semantic mappings for terminologies between different Ontologies. Query expressions are compared regarding their subsumption relations for the terminologies and their associated interpretations over instances set respectively.

The result of this is used to validate query reformulation across different conceptual models to find equivalent queries i.e. the results of a reformulated query that are identical to the answer of the original query. However, the computation of an equivalent query is difficult due to restrictions on independent conceptual models: the exact mapping between conceptual viewpoint and common Ontology is often so strict that very few mappings would exist. *Maximised containment query* rewriting is considered instead.

Containment mappings for terminologies are defined in section 0 and provide necessary and sufficient conditions for testing query containment. Since each user query has a conjunctive form of $Q(\vec{x}) = conj_1(\vec{x}, \vec{y}_1) \vee conj_2(\vec{x}, \vec{y}_2) ... \vee conj_1(\vec{x}, \vec{y}_n)$ according to a query form of Datalog, where $\vec{x}$ is variable set of query with arity m, each $x_i \subset \vec{x}, 0 < i < m$ is a single attribute of $\vec{x}$. Each $\vec{y}_i, 0 < i < n$ are other variables and constants regarding the same viewpoint that have $\vec{x} \cap y_i \equiv \mathrm{O}$. The evaluation of such a query Q over a multiple viewpoint system implies the retrieved data instances that need not only consider query expression in relational schema, but also need to satisfy the interpretation of constraints both in the common Ontology and in the viewpoint model.

The traditional view-based approach in database systems considers relational schema as a flat relational structure with key constraints. Queries are answered w.r.t. views, e.g., given a query Q and a set of view definitions V = V1, . . . , Vm, a rewriting of the query using the views is a query expression Q' whose body predicates are either V1, . . . , Vm or comparison predicates[78]. Thus, global-as-view approaches may focus on solving each sub-goal of query expression in database extension by unfolding terminologies using inclusion relations defined in the mapping. Such that, in a database integration system, answering queries is essentially an extended form of reasoning in presence of incomplete information[65]. The query answering over multiple viewpoint models may involve more complicated scenarios, e.g. the hierarchical structure and conceptual constraints need to be considered as further restrictions upon answering approach of GAV and LAV. The following situations may happen in a multiple viewpoint system:

- No coherent model exists in the viewpoint conceptualisation. This happens because retrieved instance set of common Ontology may not satisfy conceptual constrains in the viewpoint.

- Conflicts may be found for which no valid interpretation of the conceptual model can be established.
- Multiple coherent models exist in a viewpoint conceptualisation. This happens. for example, because sound views are defined for all terminology mappings, then retrieved result for the common Ontology needs to be further tailored to fit into viewpoint conceptualisation by adding more instances or attributes. Then each way actually yields a new extension of viewpoint conceptualisation in the knowledge domain. For example, IS-A relation in certain viewpoint may not be satisfied in common Ontology used for instance retrieval.
- Exactly one coherent model exists in the viewpoint conceptualisation; however, this rarely happens because independent viewpoint conceptualisation normally takes place.

The above examples show that query answering over multiple viewpoint system needs to consider more than subsumption relations between terminology correspondences. Viewpoint constraints could further restrict the query answering in viewpoint conceptualisation. The process of query reformulation consists of reasoning of mapping relations in a hierarchy structure and the validation of result against viewpoint constraints.

The containment mapping between query expressions can be formally defined: A mapping τ from Vars(Q2) to Vars(Q1) is a containment mapping if: (1) τ maps every subgoal in the body of Q2 to a subgoal in the body of Q1; and (2) τ maps the head of Q2 to the head of Q1, where Vars(Q) denotes the variable set of query Q. The query Q2 contains Q1 if and only if there is a containment mapping from Q2 to Q1 [78].

Regarding the conjunctive query over viewpoint $Q(\vec{x}) = conj_1(\vec{x}, \vec{y}_1) \vee conj_2(\vec{x}, \vec{y}_2) ... \vee conj_1(\vec{x}, \vec{y}_n)$, each variable of $\vec{x}$ and $\vec{y}$ would have predefined mapping functions to their semantic correspondences in the common Ontology that each $\vec{x}_i \approx q(g)$ and $\vec{y}_i \approx q^{'}(g)$, where $q(g)$ and $q'(g)$ denote a query expression over common Ontology as described earlier. Only sound and exact views are considered in this case, because reasoning over a complete view mapping would cause extra complexity in the computational reasoning. The query $Q(\vec{x})$ can be reformulated into an expression of containment query over common Ontology consisting of terminology substitutions by attribute and concept unfolding. Thus, if the

reformulated query has a valid interpretation over common Ontology, it can be proved that the original query would have a valid interpretation over the same domain. However the reasoning approach considers only terminology subsumptions, the actual conceptual constraints in viewpoint are ignored in the phase.

To solve this situation, retrieved results are validated against mismatched conceptual constraints in the post-processing phase. As a retrieved instance set corresponds to a valid interpretation of a transformed query over common Ontology, so all constraints of common Ontology must be satisfied. However, constraints of viewpoint conceptualisation are not enforced in the common Ontology during information retrieval. Suppose viewpoint constraints are only further restrictions, because conceptual conflicts have been filtered out in design phase. The post-processing approach mainly focuses on solving hierarchy-based constraint and functional dependencies throughout the viewpoint conceptualisation, whereas other mature logic-based reasoning is conducted in a well-developed logic algorithm that is embedded in a third-party reasoning engine. The following rules are presumed to conduct hierarchical and functional constraints:

- IS-A relations for generalisation and instance sets of child-class are combined to generate a new instance set.
- IS-PART relations for aggregation means that each instance is aligned with relevant instance according to aggregation relation to form a new instance.
- Functional Dependencies are processed in compliance with specific operations such as average, maximum etc.

5.4.7 Applying Preference and Rules in Query Answering

Explicit role-specific rules contribute to the representation adaptation between viewpoint conceptualisation and user preference that are applied in the process of terminology expanding and result adjustment. For example an EU Policy maker may want to ask: "What is the status of the water quality of the river Thames in 1980? "

In order to solve this, the following relevant rules would be taken into account for pre-processing and post-processing of query evaluation.

$$\text{River}(?r) \wedge \textit{Country}(?n) \wedge \text{isLocatedIn}(?r,?n) \Rightarrow \text{appliedQualityStandard}(?n)$$

The above rules specify that country name of a given river associated with the query determines the classification standards for water quality. If the UK standard for water

quality is applied, then Nutrient Grading can be derived from NatriteGrading and PhosphateGrading classification. Thereafter, the following rules can be applied to further interpret the grading of Nitrite and Phosphate according to the UK standards.

$$appliedQuailityS\tan dard(?n) \wedge Country(?n) \wedge (equal(?n,"uk") \Rightarrow ($$

$$NutrientsGrading(?x) \Leftarrow NatriteGrading(?x) \wedge PhosphateGrading(?x))$$

The following rules can further affect the meaningful interpretation of *NatriteGrading* and *PhosphateGrading* in context of UK standards.

$$LessThan(averageValue(?t,?t-3,?c,?x),5) \wedge observe(?c,?x) \wedge totalNitrate(?c)$$
$$\wedge catchment(?x) \wedge inUnit(?c,mg/l) \Rightarrow NatriteGrading(1) \wedge appliedQualityS\tan dard(UK)$$
$$LessThan(averageValue(?t,?t-3,?c,?x),0.02) \wedge observe(?c,?x) \wedge totalPhosphate(?c)$$
$$\wedge catchment(?x) \wedge inUnit(?c,mg/l) \Rightarrow PhosphateGrading(1) \wedge appliedQualityS\tan dard(UK)$$

According to UK GQA standard, totalNitrate can be further defined in viewpoint Ontology:

$$Observation(?x) \wedge hasDeter\min end(?x,Nitrate) \wedge hasMedium(x,water)$$
$$\wedge isAnalyse(water,totalMedium)) \Rightarrow totalNitrate(x) \wedge appliedQualityS\tan dard(UK)$$

The implicit facts associated with a user query are put into a knowledge model during the process of query expanding. The satisfied rule in an associated rule-based processing engine is fired. Then the effected facts are replaced in the knowledge model. The process runs in an iterative manner until all effected rules have been fired.

A rule-base system fulfils two aims. Firstly, it enables the pre-processing query generation to adapt to the representation gap between the individual knowledge and the viewpoint presentation and to interpret user query in a recognised expression according to the viewpoint conceptualisation. Secondly, it supports post-processing of the retrieved information to adjust the representation according to user preferences.

5.5 Multi-view Implementation

5.5.1 Overview

Viewpoint conceptualisation and aggregation concepts are added as an extension to the EDEN-IW project system to support a multiple viewpoint conceptualisation for retrieving environmental information, see Figure 11. The multiple viewpoint system is implemented as a Java-based application consisting of two sub-systems, one for viewpoint adaptation and management, the other for query processing and query result adjustment. A general conceptualisation framework consists of four key components: Ontology, Ontology parser, rule-base reasoner, and logic inference engine.

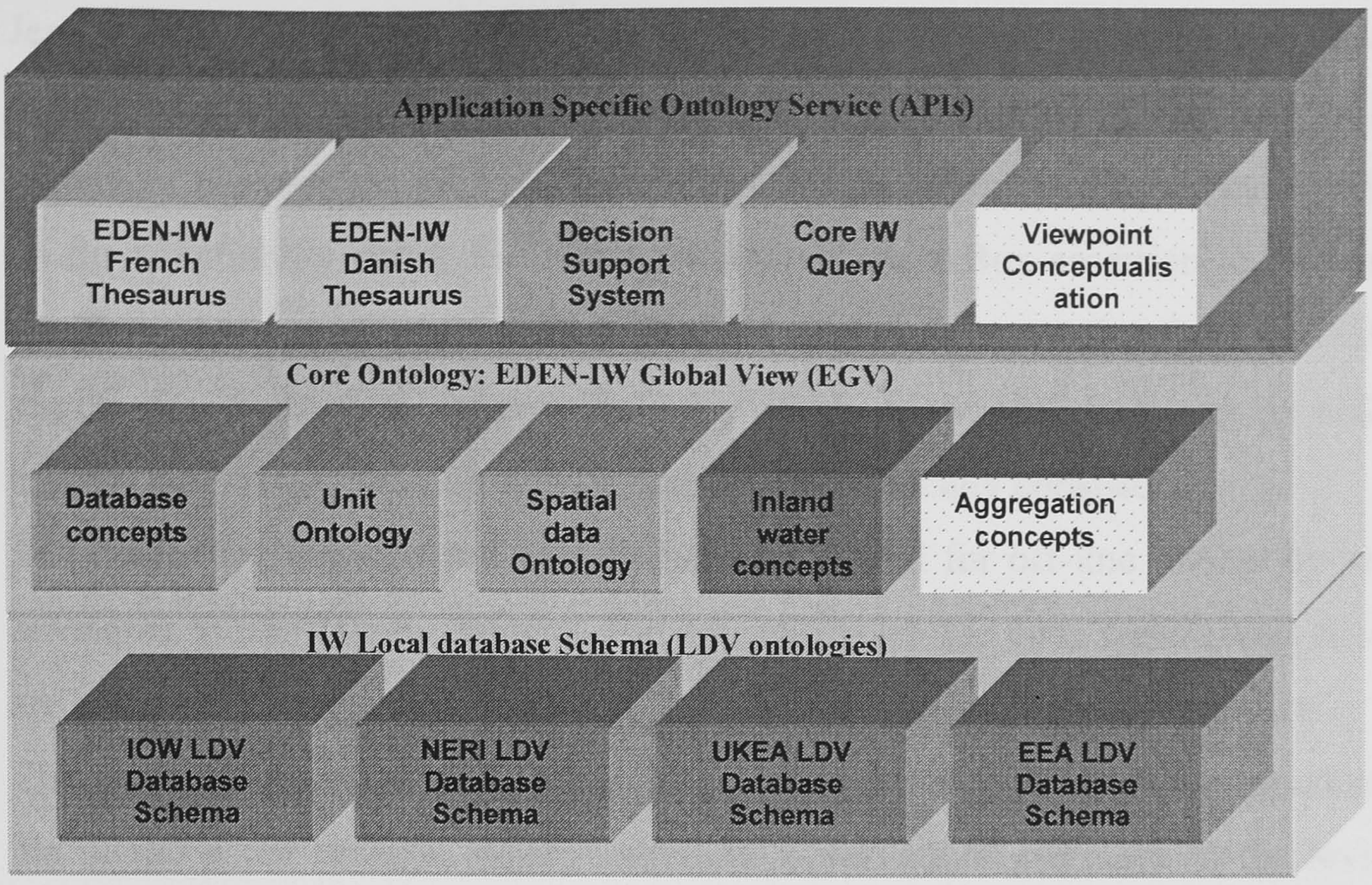

Figure 28 Multi-lateral Ontology in the EDEN-IW system[6]

The Ontology model is written in OWL-DL or Ontology Web Language-Description Logic, a subset language of OWL that was designed to support the existing Description Logic business segment and to provide a language subset that has desirable computational properties for reasoning systems [16]. The Ontology model is edited using Protégé, an Ontology editor with a graphic interface that can mask the syntax details of Ontology language for Ontology developers. The OWL Ontology is parsed and loaded into internal programming structures for operational processing using Jena, a Java-based Ontology parser with a limited inference capability for reasoning about OWL-Lite. More complicated logic inference is processed in an external inference engine, Pellet [7] in which, Ontology validation and subsumption reasoning is reduced to a satisfiability problem of description logic SHIOQ. The DIG [14] interface of Jena supports the ability to plug-in any standard logic engine, such as Racer and Pellet, so that Ontology application needs only to call Jena's reasoning functions API instead of talking to inference engine directly. This allows different reasoning engines to be changed without impacting the system implementation.

5.5.2 Viewpoint Management and Adaptation

The conceptualisation management and viewpoint adaptation is processed in an iterative manner. The output of Protégé is exported in OWL format and loaded into

---

[6] The boxes in white indicate the extensions to the EDEN-IW given in this chapter.

Jena and then into Pellet for any consistency checking. Consistency checking is conducted in two phases: for the stand-alone viewpoint Ontology and for the Ontology alignment between the viewpoint Ontology and common Ontology. Phase one focuses on the satisfiablity check for TBox declaration and ABox assertion to find any logic conflicts in the conceptual model. Phase 2 deals with inter-Ontology consistency to ensure the conceptual expressions of a single viewpoint Ontology is able to be transformed into a common Ontology that is semantically consistent.

The output for any detected conflicts can be logged for off-line review by viewpoint developer to improve the system. The process cycles and ends when either all conflicts are solved or the user is notified and instructed to ignore existential conflicts.

The scientist viewpoint, see Figure 29, provides a fairly detailed conceptualisation for the inland water domain, where individual observations are made at the level of the regular measurement of determinand concentration in a particular medium and analytical fraction. Observations are associated with spatial information such as monitoring station, river and catchment. The scientist viewpoint can be exactly mapped to the EGV or common Ontology concepts model to provide detailed information at a low level granularity with complete coverage. No intended perspective has been defined. Each virtual database table can be mapped to an EGV concept using exact mapping relations and each attribute has an equivalent semantic property defined in the EGV Ontology. Other viewpoints for the environmental information can be derived using basic concepts in scientist viewpoint via schematic or terminology adaptation steps.

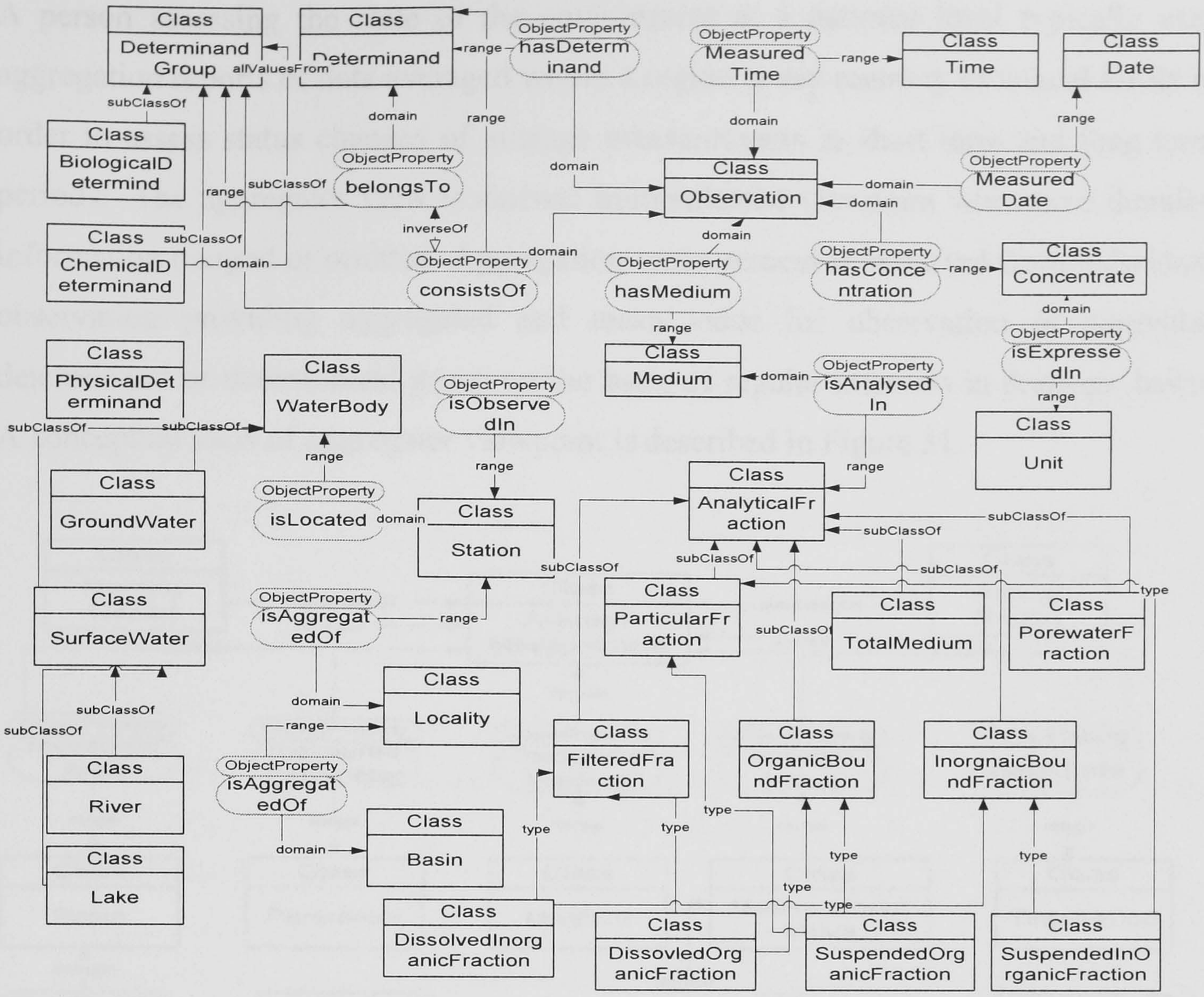

Figure 29 The conceptualisation of scientist viewpoint

The conceptual model of the scientist viewpoint upon inland water domain given in Figure 29 can be related to the original relational schema in Figure 30. Expanding focuses on constructing hierarchy structures and OWL-DL constraints. This process has been described in section 5.4.5.

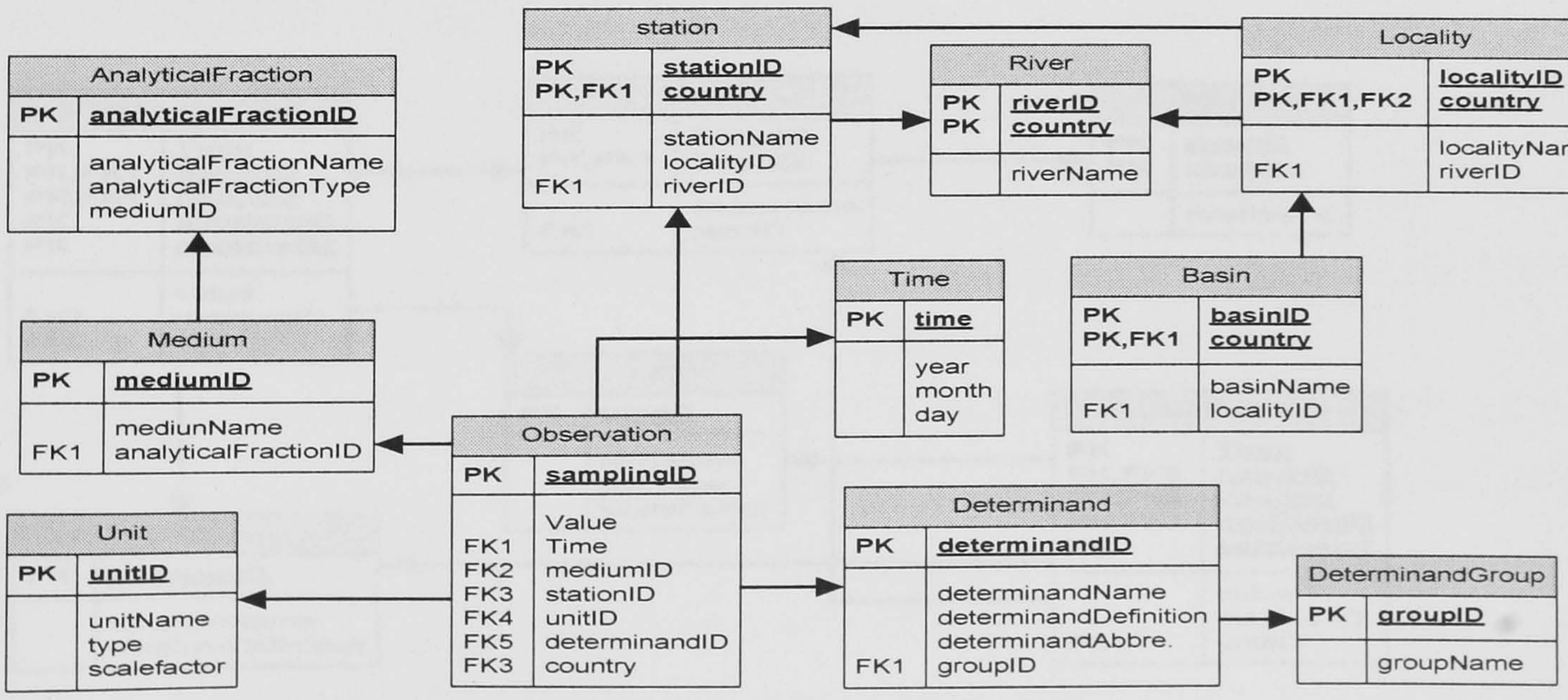

Figure 30 Relational Schema of Scientist viewpoint

A person assessing the state of the environment at a national level typically uses aggregation reports of data averaged within a region at say monthly or annual levels in order to assess status changes of average measurements in short term and long term periods. The aggregator view is derived from scientist viewpoint with some detailed information merged or omitted. Aggregation measurement is extracted from individual observation providing aggregated and mean value for observation of particular determinand or determinand group on the basis of regular intervals in drainage basin. A conceptual view of aggregator viewpoint is described in Figure 31.

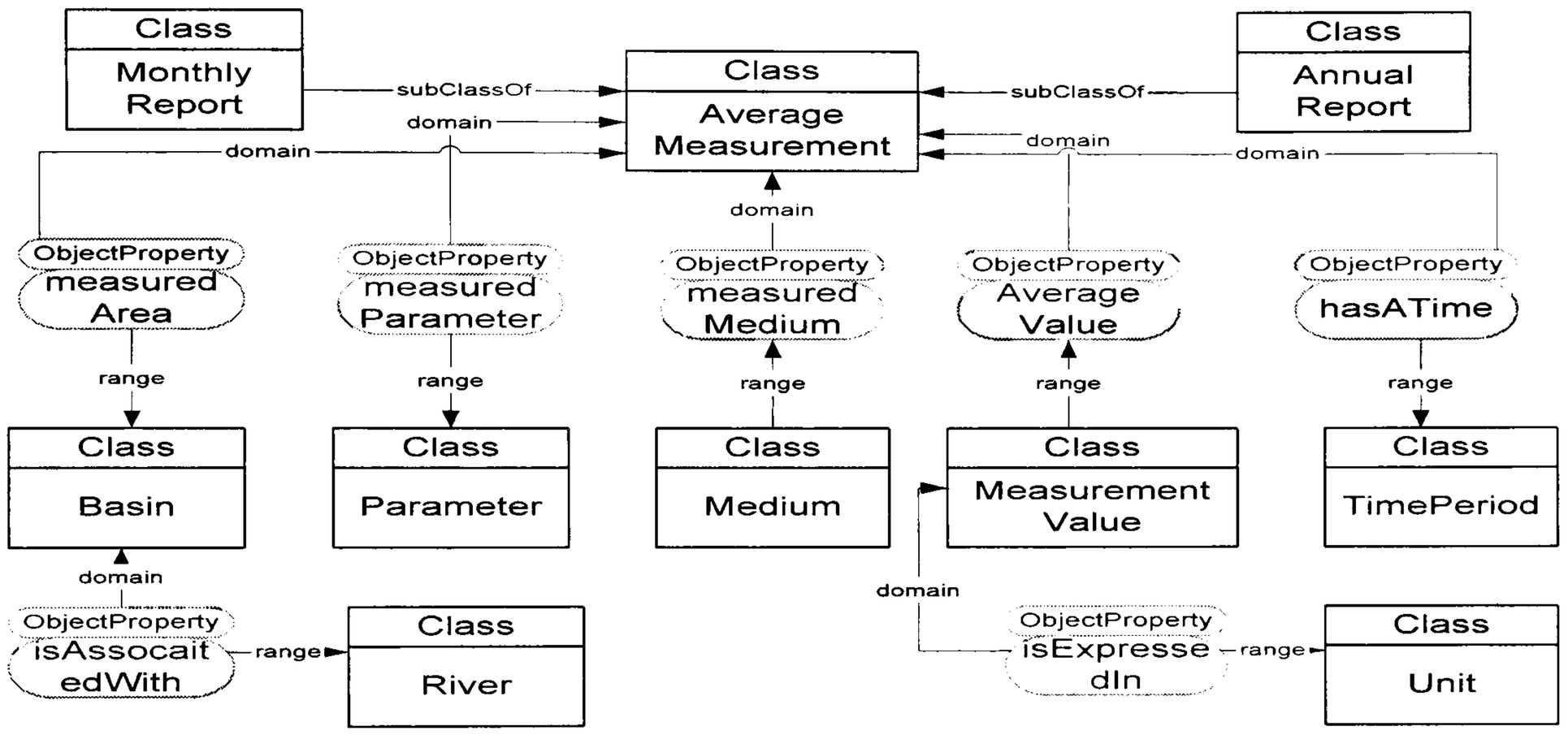

Figure 31 The conceptualisation of aggregator viewpoint

Similarly a virtual relational schema can be derived from the conceptual model, see Figure 32, to allow SQL queries of he aggregator to be posed.

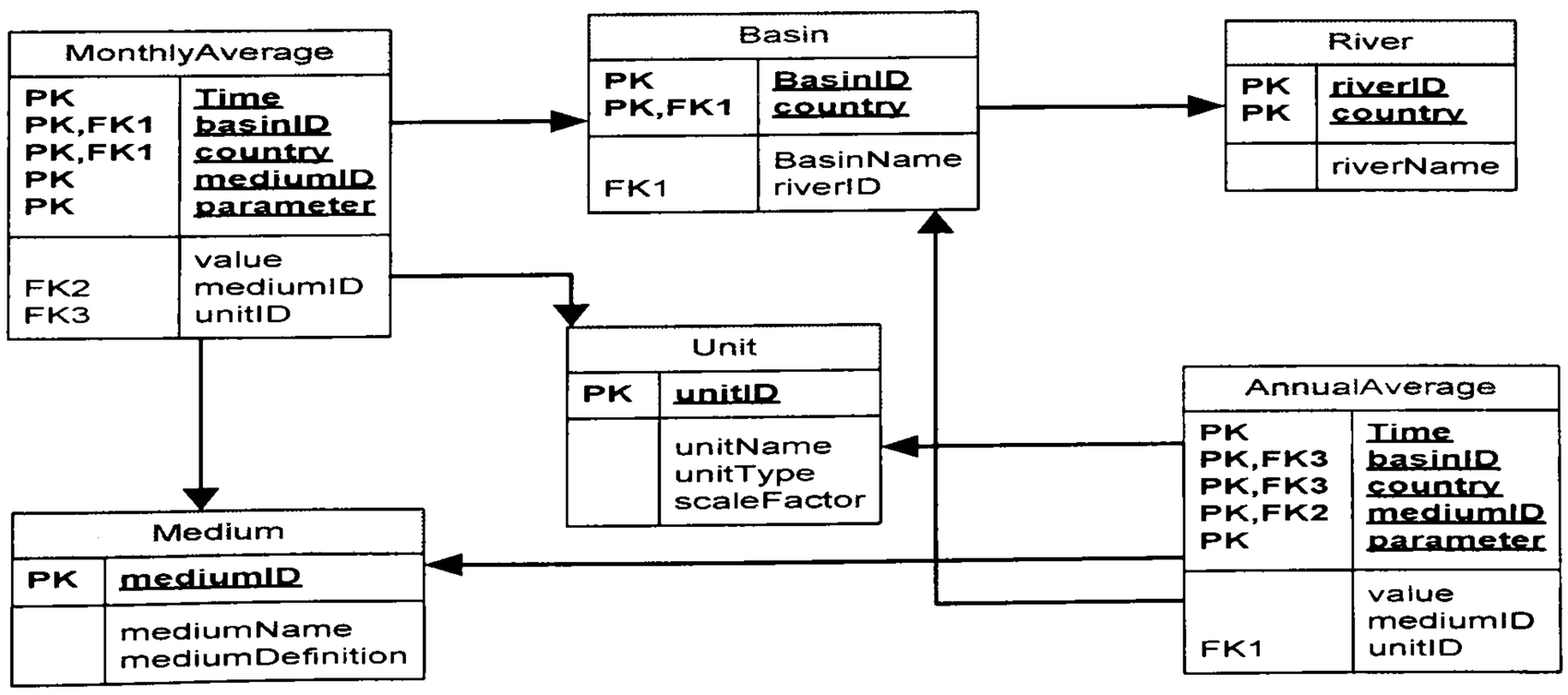

Figure 32 Relational schema of Aggregator viewpoint

In order to establish a proper semantic interpretation of the aggregator viewpoint, the semantic mapping links have been defined as an adaptation path from the scientist's viewpoint to form an aggregator viewpoint, where conceptual constructs in aggregator viewpoint can be derived from scientist viewpoint via a sequence of operations at both the data instance level and the metadata concept level. The process for deriving the Terminology of MonthlyReport is as follows:

$$\text{MonthlyRpt.time} \Leftarrow \gamma_{year+month}(Observation.time)$$

rule on instance level

$$\text{MonthlyRpt.basinID} \Leftarrow \pi_{basin\,ID}((\sigma_{observatoin.time \subseteq monthly\,Report.time}(Observation)) \triangleright\triangleleft Locality \triangleright\triangleleft Basin)$$

$$\text{MonthlyRpt.country} \Leftarrow \pi_{country}((\sigma_{observatoin.time \subseteq monthly\,Report.time}(Observation)) \triangleright\triangleleft Locality \triangleright\triangleleft Basin)$$

$$MonthlyRpt.medium \Leftarrow \gamma_{forward}(\pi_{mediumName,analyticalFractionName}(\sigma_{observatoin.time \subseteq monthly\,Report.time}(Observation \triangleright\triangleleft Medium \triangleright\triangleleft AnalyticalFraction)))$$

$$MonthlyRpt.parameter \Leftarrow \rho_{determinandName \rightarrow parameter}(\pi_{determinandName}(\sigma_{observationTime \subseteq monthlyRpt.time}(Observation \triangleright\triangleleft determinand \triangleright\triangleleft determinandGroup))) \cup \rho_{groupName \rightarrow parameter}(\pi_{groupName}(\sigma_{observationTime \subseteq monthlyRpt.time}(Observation \triangleright\triangleleft determinand \triangleright\triangleleft determinandGroup)))$$

$$MonthlyRpt.value \Leftarrow func_{average}(\pi_{value}(\sigma_{observatoin.time \subseteq monthly\,Report.time}(Observation)))$$

Similarly, an annual report can be derived from the scientist construct. The process to create a Path to form aggregator viewpoint is as follows. In order to derive aggregator view from scientist view, a continuous conceptual change operating upon scientist view needs to be considered, such that the following adaptive path is defined.

- $addConcept(Medium'(mediumName) \mid \lambda((Medium.mediumName, AnalyticalFraction.analyticalName), (Medium \triangleright\triangleleft AnaliticalFraction)) \mid exact)$

  $add\,Property(Medium'(mediumDef) \mid \lambda((Medium.mediumDef, AnalyticalFraction.analyticalDef), (Medium \triangleright\triangleleft AnalyticalFraction) \mid exact)$

  $addConcept(Monthly\,Report(time, basinID, country, mediumID, parameter, value, unitID) \mid \rho_{time' \rightarrow time, stationID \rightarrow basinID, determinandID \rightarrow parameter}(\pi_{time', stationID, country, medium, determinandID}(\gamma_{month+year}(time', \lambda_{medium}(mediumName, analyticaName, (\kappa_{(basinID, country), (stationID, country)}($

  $\kappa_{determinandName, determiandID}(Observation, Determinand) \cup \kappa_{groupName, determinandID}(Observation, Determiand, DeterminandGroup)), Station, Locality, Basin) \triangleright\triangleleft Medium \triangleright\triangleleft AnalyticaFraction)))) \mid exact)$

- $add\ \mathrm{Pr}operty(MonthlyRpt(value\ )\ |$
  $\varpi_{average,groupby(time,ba\sin ID,country,mediumID,parameter)}(Observation.value))\ |\ exact)$
- $add\ \mathrm{Pr}operty(MontlyRtp(unitID)\ |\ \pi_{unitID}(Observation)\ |\ exact)$

By now, we have got a combined schema $V_s \cup V_a$, we are going to remove schematic constructs of V_s and finally to reach schema V_a.

- $remove\ \mathrm{Pr}operty(Observation(UnitID)\ |\ \pi_{unitID}(MonthlyRpt)\ |\ exact)$
- $remove\ \mathrm{Pr}operty(Observation(value)\ |\ void\ |\ unknown)$ [7]
- $removeConcept(Observation(time,\det er\min andID,mediumID,)\ |)$
  - addProperty(MonthlyRep(value)|)

$addConcept(C,(k_1,k_2...k_n)\ |\ o\ |\ v)$

- $add\ \mathrm{Pr}operty(C.p\ |\ o\ |\ v)$
- $add\ \mathrm{Re}lation(C.p,D\ |\ o\ |\ v)$
- $addConstra\mathrm{int}(con,C(C.p)\ |\ r)$
- $removeConcept(C,(k_1,k_2...k_n)\ |\ o\ |\ v)$
- $remove\ \mathrm{Pr}operty(C.p\ |\ o\ |\ v)$
- $remove\ \mathrm{Re}lation(C.p,D\ |\ o\ |\ v)$
- $removeConstra\mathrm{int}(con,C(C.p)\ |\ r)$

A Policy maker's viewpoint , see Figure 33, is derived from aggregator viewpoint again by defining another path.

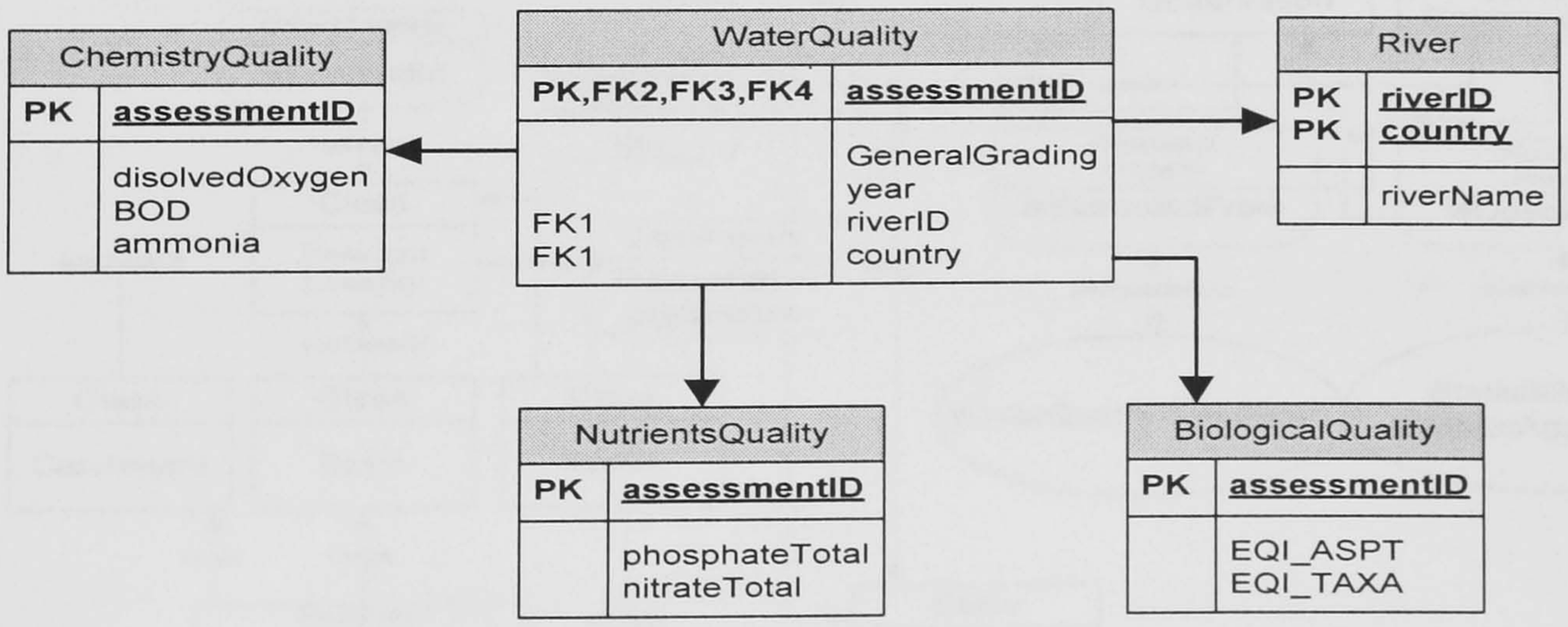

Figure 33 Viewpoint Schema of Policy Maker

[7] Observation value is not recoverable from other attributes because average function is not reversible. An unrecoverable attribute is denoted as void.

The semantic mapping of viewpoint terminologies to the common Ontology model forms a valid interpretation of viewpoint conceptualisation in the knowledge domain. The viewpoint conceptualisation can be further restricted by local constraints and rules.

5.5.3 Modelling of User Profile and Role-specified Rules

User preferences are stored in a construct called the individual user profile represented using OWL. Figure 34 specifies a user preference as a conceptual interest of a subset of domain knowledge via a certain terminology set and presentation forms. An individual user profile is semantically connected to key concept trees such as a time series, geographic information and measurement of water quality. Such interest can be expressed in corresponding vocabulary set associated with one of three targeted user groups regarding scientist, policy maker, and legislator and aggregator. A presentation preference specifies different formats such as a summary table, trends diagram and pie chart.

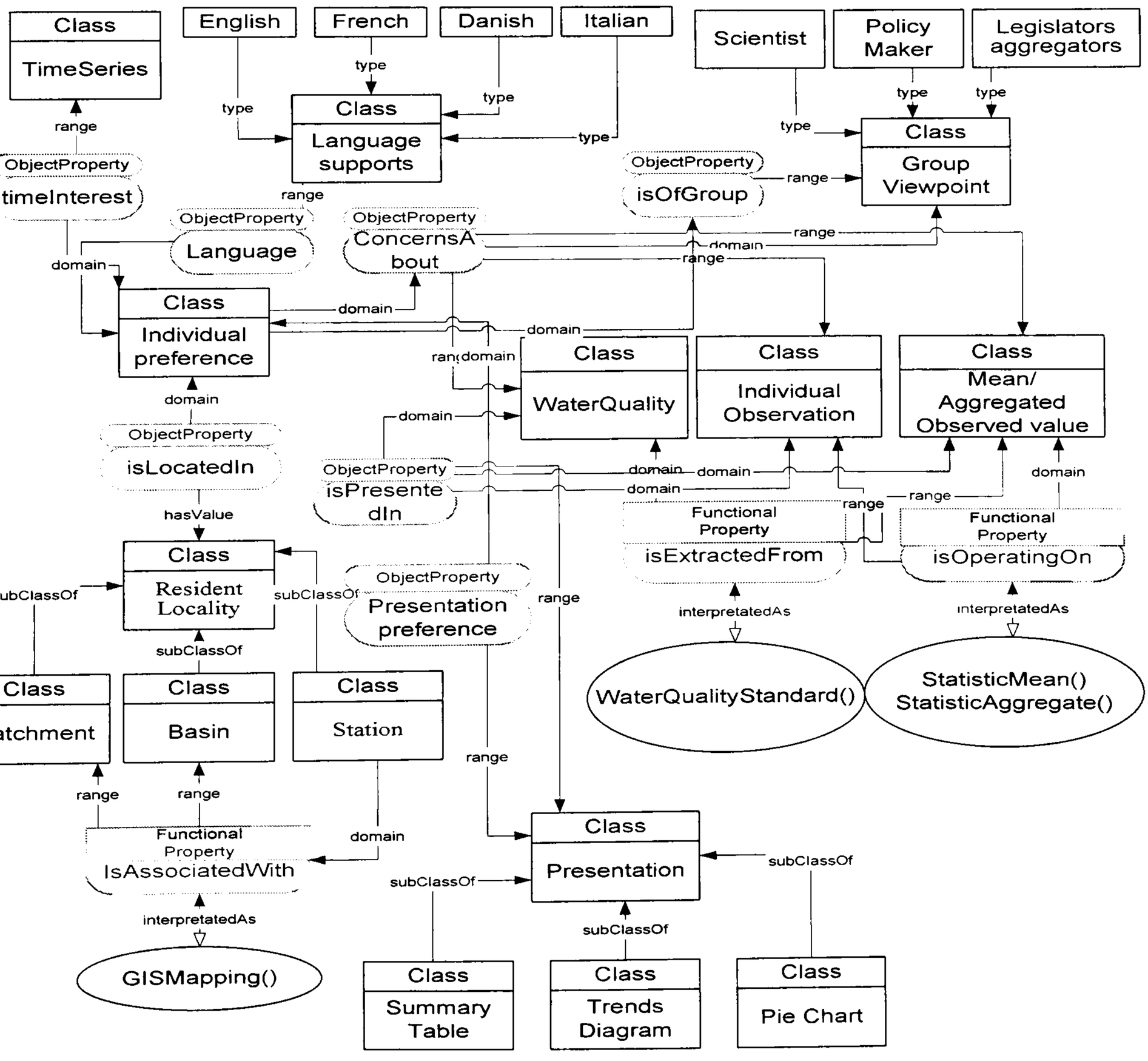

Figure 34 Conceptual model of user preference

A user is allowed to select his natural language including English, French, Danish and Italian. Language translation uses a conversion table of a multi-lingual glossary represented in XML. The role-specified rules are specified in a CLIPS-like format according to Jena standard that can be processed in a general rule engine that forms part of Jena. The rule engine can process CLIPS-like rule specification. Both forward-chaining and backward-chaining reasoning are supported, whereas only forward-chaining rules are specified in EDEN-IW system.

5.5.4 Query Answering

The process of query answering and result adjustment has been implemented in the architecture given in Figure 35, where query answering and result adjustment have been achieved in a 3-phase model including a pre-answering process, answering process and post-answering process.

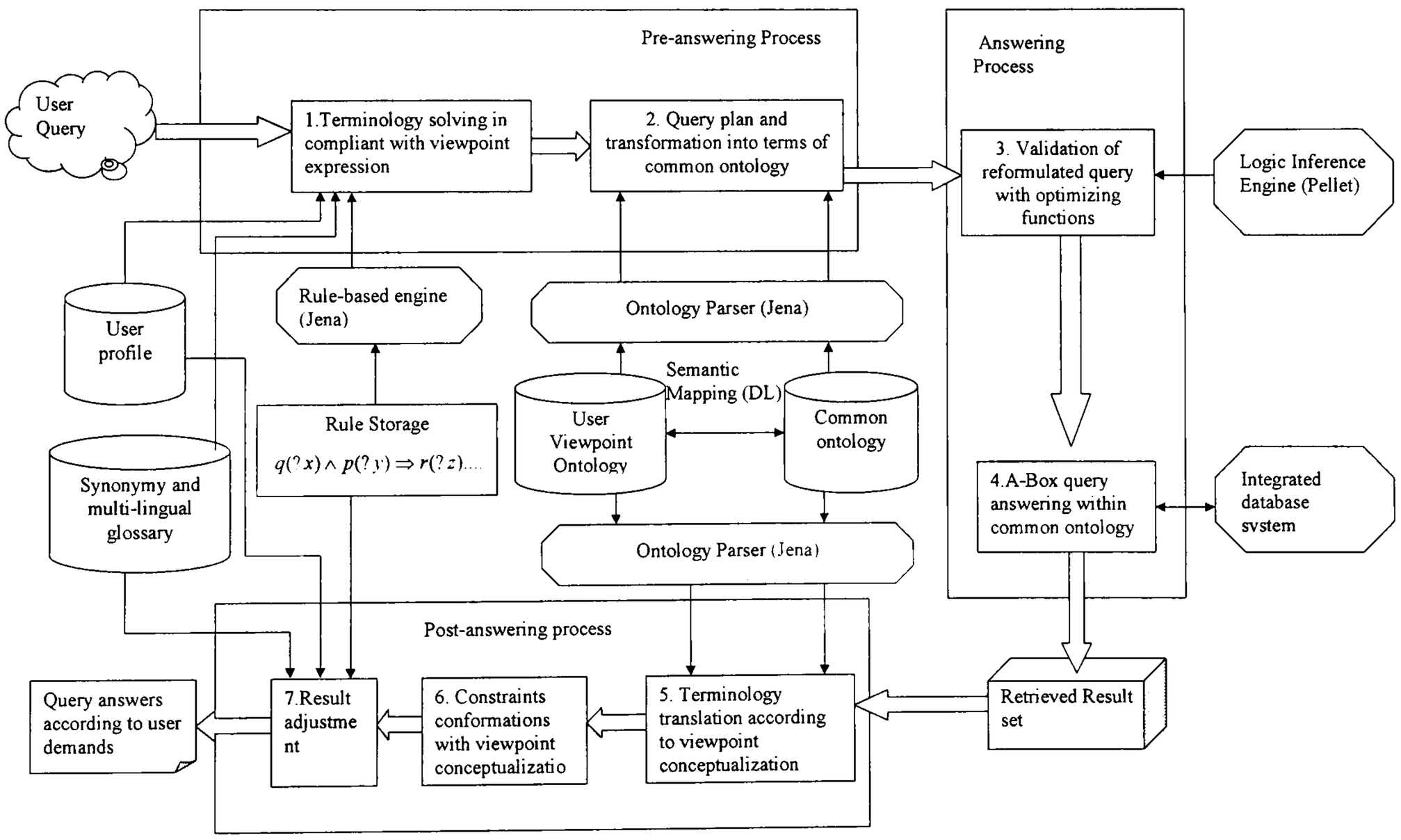

Figure 35 Architecture of the adaptive viewpoint system

5.5.4.1 Pre-answering Process

The pre-answering phase contains two sub-processes for terminology resolution and query reformulation. In the former process, a user is guided to construct query in their familiar terms that are translated to viewpoint terminology via synonym and multi-

lingual mappings defined in the glossary. The user query is constructed in a SQL-like syntax in RDF structure as described in the chapter 4. User preference and usage rules are also imported to resolve relevant terminologies into an appropriate level in compliance with semantic mapping.

An individual user profile is associated with a specific group viewpoint, where user terms can be mapped appropriately using synonym relations. Identification of business role of user is linked to rule specifications in rule storage, where underlying knowledge of specific business roles is explicitly specified. The role specific rules mainly resolve the homonym problems in which the same term is interpreted differently according to business role. The rule specifications are loaded into rule engine in Jena, which is configured in a Forward-Chaining model. The terms in a query expression are validated against rule conditions. If any rule trigger satisfies the corresponding terms in a query expression, it is expanded and replaced using the appropriate literals defined in rules. The condition validation and rule firing is conducted iteratively. The expanded and replaced query is checked against rule conditions repeatedly until no new satisfactory rule is found. The output of the terminology resolution is forwarded to the next sub-process of query reformulation.

Query reformulation is conducted in two steps: sub-query planning and mapping unfolding. As the viewpoint conceptualisation is expressed in a hierarchy structure, the production of rule-based terminology resolution is not sufficient to be mapped into a common Ontology terms in a straightforward manner. A reasoning approach is required to cover the distance and decompose query into sub-queries. The reasoning focuses on the hierarchical operations defined in section 5.4.4.2 reduces the query granularity into an appropriate level where all terms can be directly mapped to a common Ontology. The term unfolding uses a GAV like query-rewriting approach regarding the soundness and completeness of views for the semantic mapping of the viewpoint terminology to expressions in the common Ontology. The relevant terms in sub-queries are substituted by its semantic correspondences.

5.5.4.2 Answering Process

The reformulated user query is further validated and optimised using logic-based query optimising functions to remove the redundant sub-goal expression for reformulated query. The optimised query is forwarded to a virtual database for result answering when all constraints of common Ontology are satisfied. The virtual database is realised

in EDEN-IW as an integrated database system with the EGV Ontology. The process of information retrieval from underlying data source has been described in chapter 4. The result set of query in RDF structure is returned to the post-answering process for result adaptation.

5.5.4.3 Post-answering Process

The results set returned from the answering process needs to be processed in compliance with user specified representation of the conceptual model and terminologies defined in the individual user profile. The results set in the instance table is reformulated into viewpoint structures by going through an adaptation path where conceptual operations have been defined in relevant semantic mappings, see section5.3.2. The reformulated results must reflect an ABox set of viewpoint conceptualisation in its OWL representation. The transformed ABox set assertion is evaluated with a TBox definition and constraints. Any part of the unsatisfied results is filtered out. The constraints integrity of viewpoint conceptualisation is guaranteed. The result of a successful evaluation is further adjusted according to the user preference and user roles. The adjustment operations including synonym translation, rule-specified terms resolution and result presentation.

5.5.5 Validation

The multiple viewpoint system is validated by posing user queries at specified viewpoints with respect to a particular user profile and role specification. For example, the following information has been taken into account to answer a query in policy maker's viewpoint such as "What's the quality status of river Thames in 1980?":

- User role: UK environmental policy maker
- Applied Standard: UK General Quality Assessment
- Associated viewpoint: policy maker's viewpoint

In a traditional IR system, answering such a question is not possible, because the query specifies information that is too vogue to be retrieved from the database directly. In a multiple viewpoint system, the original query is able to be expanded and adapted using semantic analysis according to a viewpoint conceptualisation. In this case, sub-queries would be generated for UK standards with respect to Nutrients and Total Phosphorus with restricted time and geographical scope. The retrieved result from an integrated database system is further adjusted according to an average function and grading

standards for corresponding parameters. Test cases have been made for each viewpoint to validate its flexibility to handle queries with different conceptualisation and different user roles.

Table 19 Validation of viewpoint system via test case

Group viewpoint	***Query Expression***	***Traditional IR***	***Conceptual IR with viewpoint***
Scientist	*Concentration of Nitrite in river Y at time T?*	*Possible*	√
Legislator and aggregator	*Monthly mean concentration of total Nitrite in basin X of river Y?*	*Aids of expertise*	√
Policy Maker	*What's the most polluted area in river X at time Y?*	*Aids of expertise*	√

Table 19 shows that high-level queries posed on a group viewpoint with the underlying conceptualisation are successfully handled via a multiple viewpoint framework in conjunction with the conceptual IR system. The answering of such queries in a traditional IR system requires more human support for information processing and conceptual interpretation. The multiple-viewpoint framework automates the processes via an explicit definition of domain conceptualisation. The ability to adapt information to multiple viewpoints provides more flexible usage.

The following parts of the multiple view framework have been fully implemented: user profile, group viewpoint model, viewpoint development process, query augmentation, query answering and result adjustment. Snapshot pictures are shown as follows for different presentation modes of the query result. Figure 36 shows the Trends Diagram of query result on scientist viewpoint. The query asks for the Nitrate value in three different stations

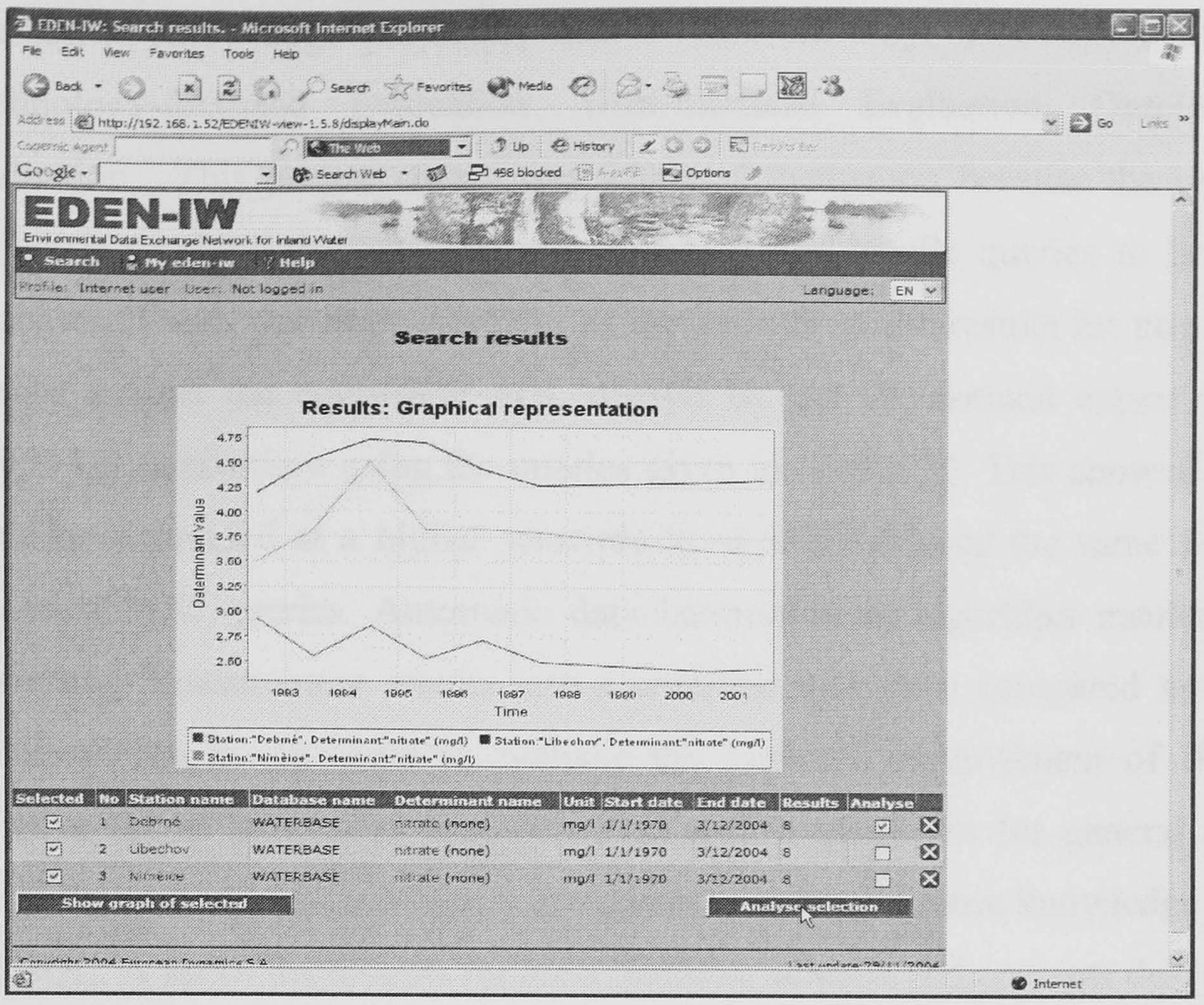

Figure 36 Trends Diagram of Query Result

The same result can be represented in different representation model such as summary table as shown in Figure 37 according to the user preferences defined.

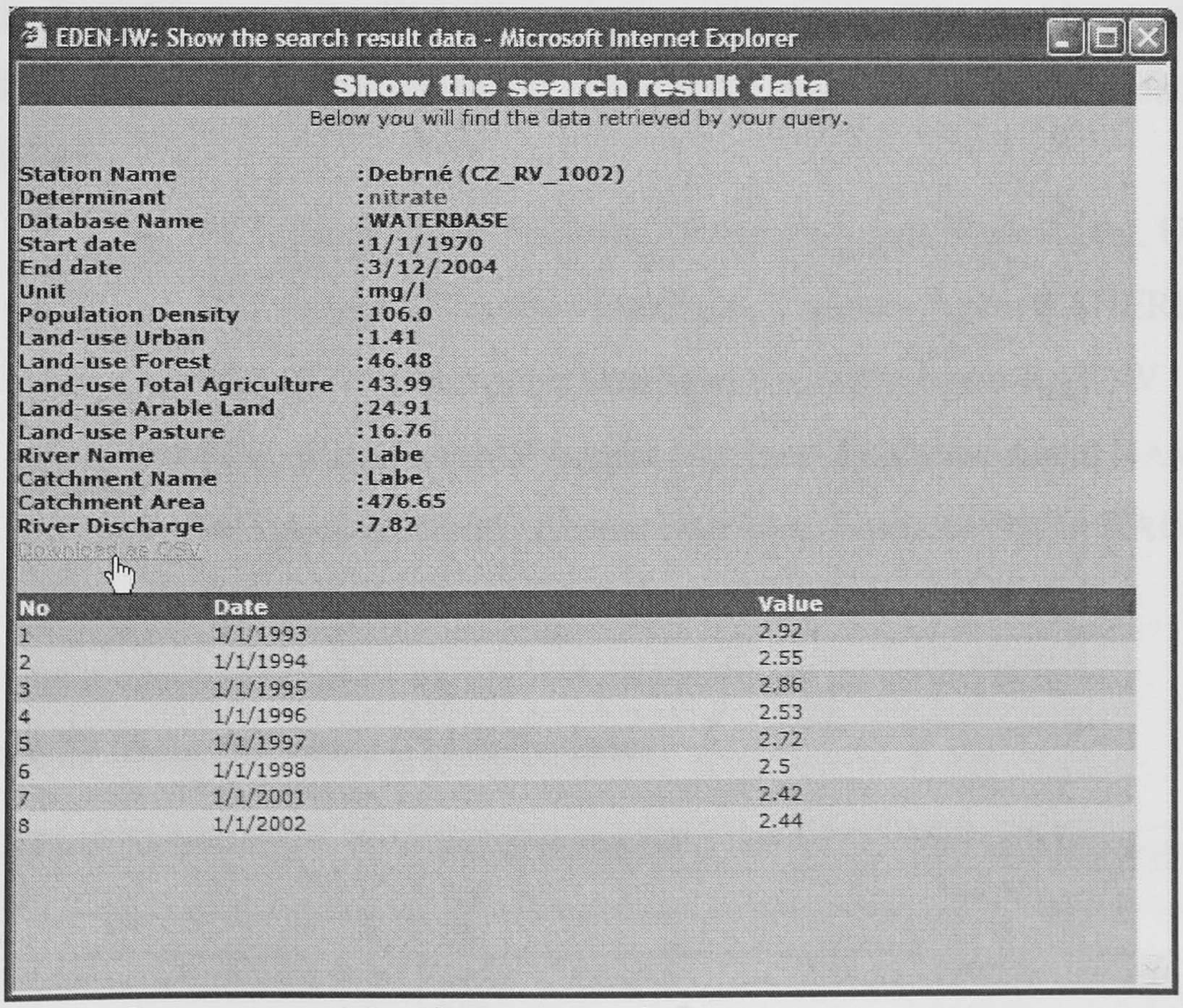

Figure 37 Summary table of query result

The On-to-Knowledge project[6] lists 3 different ways to evaluate an Ontology: Technology-focussed Evaluation, User-focussed Evaluation, Ontology-focussed Evaluation. This project uses a user-focussed evaluation because the motive of the project is to show that an Ontology approach can handle queries to heterogeneous databases. There was limited testing of the validity of the results for posing semantic queries against the equivalent SQL queries by the IW domain expert users in the EDEN-IW consortium using the queries given in Table 22. This showed that queries could be expressed at a higher semantic level and produced the same results as the low-level SQL queries. Automatic data harmonisation algorithm results, e.g., from converting measurement results into a common unit were compared with manually calculated results. In order to evaluate the usability improvement of IR system, a selective set of beta-users were provided with a query list for general concerns of environmental information. Users are assumed to have a basic knowledge of database and SQL. A set of documents for local database schema and contents description were provided. Users were instructed to answer specified queries using both direct access of SQL query posed on local database and via the EDEN-IW interface. Timing measurements for answering queries were also recorded in order to analyse the time taken by sub-processes. The setup of the evaluation system consisted of 6 PCs (2.8 GHz Pentium IV with Windows XP), where the following agents and databases were installed:

- PC1: Main JADE Platform, Directory Agent, Task Agent, User Agent
- PC2: NERI MS Access Database, Resource Agent RANERI
- PC3: IOW MS Access Database, Resource Agent RAIOW
- PC4: Waterbase MS Access Database, Resource Agent RAWB
- PC5: UK-HMS MS Access Database, Resource Agent RAUK
- PC6: User Interface (Web Browser)

The main characteristics of the applied databases are given in Table 1.

Table 20 Main Database Characteristics

Database	No. of Stations	No. of Measurement Records
NERI	534	346380
IOW	30	92278
Waterbase	3438	189253
UK-HMS	277	565225

Table 21 Example of time caculation of query answering

Time (hh:mm:ss.ssssss	From->To	No. of bytes	P:Process/T:Transmission	Time (h:mm:ss.sss)
16:18:55.997248	UI->UA			
16:18:56.215905			P: UA+TA	0:00:00.022
16:18:56.237517	TA->RANERI			
16:18:56.287748		13280	T:TA -> RANERI	0:00:00.050
16:18:56.238352	TA->RAUK			
16:18:56.260753		13280	T:TA -> RAUK	0:00:00.023
16:18:56.242430	TA->RAWB			
16:18:56.327146		13276	T:TA -> RAWB	0:00:00.085
16:18:56.258284	TA->RAIOW			
16:18:56.274199		13278	T:TA -> RAIOW	0:00:00.016
			P:RAWB	0:00:58.826
			P:RANERI	0:00:27.772
			P:RAIOW	0:00:01.166
			P:RAUK	0:00:18.072
16:18:57.439884	RAIOW->TA			
16:18:57.488303		46740	T:RAIOW -> TA	0:00:00.048
16:19:14.332825	RAUK->TA			
16:19:14.863515		410039	T:RAUK -> TA	0:00:00.531
16:19:24.059525	RANERI->TA			
16:19:24.818783		749229	T:RANERI -> TA	0:00:00.759
16:19:55.153025	RAWB->TA			
16:19:59.105614		3821189	T:RAWB -> TA	0:00:03.953
			P:TA+UA	0:00:12.122
16:20:11.228115	UA->UI			
16:20:11.668152			T:UA -> UI	0:00:00.440
			TOTAL	**0:01:15.671**

For the monitoring of the network traffic a 7th PC (with Linux) was also connected to the network. The PCs were all connected via a wired intra-network with a standard 10 mbps hub. The output of monitoring PC is a dump file for all TCP/IP message transactions across the network. By sorting the dump file contents with specified IP address, the timing of particular query transaction can be calculated as shown in Table 21. Table 21 shows the results of performance measurements in terms of measured processing times or transmission times for verified use-case 1 semantic queries , see

section 4.2.3 . Each user is given a query list consisting of 6 queries. The testing queries contain the real concerns from different user viewpoint as table below.

Query No.	*Viewpoint*	*General Queries*
1	Scientist	What is the concentration of determinand X in river , e.g., What is the concentration of Nitrite in river CEINE during 1980?
2	Legislator and aggregator	What is the monthly averages value of determinant X at station Y in year Z?, e.g., What is the monthly average value of PH in station Rodemark during 1980?
3	Policy Maker	What is quality indicator W for River Y in year Z, e.g., What is the Nutrients GQA grade for river Thames in 2000-01?
4	Scientist	Which station has data on determinand x? e.g., Which station has data on PH?
5	Legislator and aggregator	Which station has mean value of determinand X above threshold V. Which station has mean value of total Phosphorus above threshold 0.2 mg/l?
6	Policy Maker	What station Y has the quality indicator W between years Z1-Z2, e.g., Which station had chemistry GQA grade C in river Thames from 1980 to 2000?

Table 22: Test queries for the user viewpoint evaluation

The results of testing query are compared in the following table. All results in the table are estimated using average value across testing group.

Query No.	*Num of relevant databases*	*Num of Result records*	*Times of query estimation (Direct Access)*	*Times of query estimation (EDEN-IW)*
1	1	34	120s	20s
2	1	1	420s	25s
3	1	1	600s	35s
4	4	3376	720s	74s
5	4	75	1080s	15s
6	1	2	300s	43s

Table 23: Comparison of direct-access SQL to EDEN-IW

The comparison tables show that timings increase significantly when users need to directly submit SQL queries to more than one relevant data sources. Users need to understand the different types of information heterogeneity amongst the data sources. In contrast, the time taken to answer queries at the EDEN-IW UI varies by tens of seconds. The supporting semantic-rich metadata services guide the generation of local SQL queries and the any post-processing of the results such as to support measurement unit harmonisation. Processing time is a major element of the information retrieval process, e.g. the processing time/total time ratio is about 90%, see Table 22. Practical

experience has also shown that the size of result record can also affects the post-processing time.

5.6 Summary

The use of a semantic global view and conceptualisation that is a higher level of abstraction and closer to the related physical world conceptualisation than the stored data model conceptualisation can enable query transparency and data result harmonisation in the face of data heterogeneities to be supported. However, the conceptualisation may still be too complex for less knowledgeable end-users to work with. A more flexible approach is needed to adapt information to data views targeted at specific kinds of usages. Such an approach has been developed in this chapter as an extension to the semantic framework described in the previous chapter, however, this introduces further complexity and additional data heterogeneities to be handled.

To make the processing of user views of the stored data to be more computationally tractable, user views are restricted to be one of a fixed set of constrained derivations of the global conceptual model. Information adaptation to user views can take place along the dimensions of: coverage of subsets of the concepts in the global conceptual model; granularity split into conceptual and processing granularity; perspective split into terminological conceptualisations, presentation styles and multi-lingual preferences.

This chapter has described an extension to the system to integrate information retrieval databases to support multiple user viewpoints. Information usability is improved by providing information adaptation to cope with different user groups. A general framework supporting IR with multiple viewpoints is able to handle the conceptual adaptation of different user viewpoints at different perspectives including viewpoint development and management, semantic mapping, query transformation, query answering and result adaptation. General IR requirements such as query augmentation, multi-lingual synonyms, customised information retrieval and customised presentation are satisfied.

Ontology-based viewpoint modelling and adaptation has effectively guided the IR interaction between user and data sources with an expressive description of conceptualisation in a hierarchical and logic-based structure. User preference and the underlying knowledge are processed in an explicit manner that can be reused in multiple applications. The successful cases have demonstrated the utility of using a

Semantic Web user-centred approach for information modelling and extraction. The framework has been implemented as an extension to the EDEN-IW project system to support information retrieval and analysis of quality status for inland water with different levels of knowledge understanding. It enables the information demands of different users to be more effectively handled.

Chapter 6 Discussion, Further Work and Main Conclusion

6.1 Discussion

The research work presented in this thesis presents a semantic based computation framework to support information retrieval (IR) involving data aggregation from multiple distributed heterogeneous databases in the inland water (IW) domain. The semantic framework has been used for two separate purposes: firstly, to wrap and integrate relational data base information sources and secondly, as an access mediator to present multiple views of the data to different users. Two sets of associated issues have been analysed in depth: wrapping and integrating multiple relational database model into a semantic model, and information tailoring so that the domain knowledge can adapt to the user's background knowledge and presentation preferences.

6.1.1 A Semantic Approach to Database Integration

A Semantic approach has been adopted to support heterogeneous information interoperability, supporting a rich machine-understandable metadata conceptual model for the stored data. The conceptual model is independent of the application logic data models and the stored data structures thus promoting reuse and openness across applications and data resources. The semantic representation of the metadata about the stored database data firstly, makes this explicit model accessible in a machine readable form. Secondly, the semantic metadata model is expressive enough to capture the constrains expressed in data design models such as E-R models that relate to the physical world conceptualisation and that become omitted in the SQL model. Information interoperability can be achieved through the interlinking of semantic conceptualisations.

An ontology supports the formal representation of domain knowledge in a machine readable manner with logic and semantic structures that are understandable by general users without prior expertise of the underlying knowledge. With respect to the semantic web, the advantage concerns extensibility and flexibility when the integrated system is applied in a wider scope including different type data content with different data models. The ontology-based approach can be more complicated to maintain than

the straightforward solution such as schema mapping regarding database integration in a concrete application domain. However, an information system presented and integrated in semantic model is more powerful when the system has to handle more flexible, dynamic and heterogeneous data sources and applications and become part of global information repositories. In such a case, the ontology is regarded as a generic solution to wrap underlying knowledge and to support interoperability and reusability. A machine-processable and formally represented knowledge model supports powerful reasoning algorithms according to formal logic structures. In addition, the extensibility of the information system is improved by using ontology-based services. New data sources can be added into system without change existing applications. More domain knowledge and related can be modelled and queried in a more generic way where particular semantic relations and constraints can be traced and analysed separately from the knowledge domain. The Semantic Web promotes the general use of domain knowledge without consideration of the concrete application details. The process of development, management and reuse of domain knowledge can be formally defined. The reapplying of ontology model into different application domain can be achieved easily with less impact to existing models.

There are two main ways to do this: semantic interlinking via conceptual alignment or via conceptual merging. These are distinguished by whether or not a common or global view is used as an intermediary – yes for the merging approach and no for the alignment approach. The merged or global conceptualisation was used and its advantage is that it reduces the number of potential mappings to be a factor of the number of different Ontologies present. The merged Ontology approach is more useful when different Ontologies tend to have large similarities. It is also useful when multiple user views are supported over multiple data resources views as the use of a global view as an intermediary reduces a many-to-many mapping problem to a more computation tractable many-to-one and one-to-many mapping problem. The mapping process can be static, done at design time, e.g., when a new user view is derived from an existing view. It can also be dynamic, for example, when a query expressed using the user view model conceptualisation needs to be transformed via the global view into local data views.

A semantic approach promotes access and location transparency to hide the heterogeneities of multiple distributed databases within an application domain. e.g., users do not need to understand SQL to access the data, nor the data schema of the

stored data, nor the location of the data, nor the type of RDBMS. The semantic metadata model can also support metadata queries so that data sources can be filtered to select the ones that have the appropriate data rather than sending queries to each data source including ones that often may not contain relevant data to the query. In addition, a semantic approach is more than being a metadata about the stored data model – it can provide a higher fidelity, common, or global, conceptual model of a part of the physical world that relates to the domain, rather than projecting the database storage view to users that is an artificial view of data structures, more optimised for data storage but difficult for end users to understand. Thus it can be used to better inform of users of the meaning of the stored data and to help explain the results or why a modified data query may lead to more useful related results. More specifically a Local-as-View or LAV approach is used for the data resources to interlink to terms in a semantic model of the global view. It makes the system more extensible to accept new database connections.

A general middleware framework has been developed to support a partitioned multi-lateral Ontology model to support multi-database storage and multiple user views of the data. It is partitioned because it is easier to reuse to support an open IR system that allows its data resources and user views to be changed dynamically and it does not require each data source or data user component in the IR system to be re-structured into a common semantic structure. The partitioned semantic model requires the use of Ontology mediation to support data mappings between different parts of the Ontology model that are interlinked to a common Ontology. A multilingual canonical thesaurus, derived by an international standards committee has been linked to common Ontology so that data and metadata provenance is supported. Vocabulary mapping issues such as synonyms are resolved separately from the semantic representations.

A series of data mappings are needed to transform the query expressed in global view terms into queries that can be issued to databases. A series of data mappings is also needed to support data harmonisation so that data results can be more correctly combined, e.g., values are converted to the same units for a more accurate comparison, rather than merely returning the results to the user as a list of results without any post-processing data harmonisation. A graph-based semantic routing algorithm traverses interlinked Ontology models where a semantic network is represented using RDF predicates in order to determine local SQL relations containing a similar semantic meaning. The graph-based model is flavoured by attribute weighting through all

relation candidates to find the best match. The algorithm is suited to an integrated database information system that may lack support for formal logic but can nevertheless support data constraints and processing about these. A routing algorithm with customised attribute weights is useful for local databases so that they can benefit from accurate data retrieval and optimised query expressions[8]. The routing algorithm is interlinked into a local database model but iterative updates to it may need to be re-evaluated and adjusted, affecting the attributes weights in the corresponding routing algorithm. When a general routing algorithm is applied with different local databases, the accuracy and performance is often reduced.

Many research projects on semantic interoperability focus on issues of determining mappings between different Ontology views as the end-point rather than on using this as the means to an end to better handle the interoperability between non-semantic data instances, e.g., database extensional data. In this case, the research and development must also deal with the issues of generating database queries and in handling post-processing of the result sets returned. This is not only far more complex because it must also deal with issues in forming satisfiable database queries to return non-empty data sets or explain why they do not.

A computational framework has been built as the author's contribution to an EU project, EDEN-IW, to demonstrate a working semantic framework to support interoperability between heterogeneous databases. In the demonstrated system, the mappings firstly between the global semantic schema and the local semantic schema and then secondly, to the local syntactical schema, e.g., SQL schema of the stored data, needed to be constructed and processed at run-time. These mappings can be simple vocabulary (direct concept to equivalent concept) mappings but they can also be more complex syntactic and semantic compositional (single concept to multiple concept) mappings and they may need to take into account constraints and rules that constrain the mappings. In the EDEN-IW system, these mappings and the associated reasoning about mappings were shown to retrieve the expected results during validation, but the mappings were largely ad hoc and hard-coded into the database Ontology wrapper software.

---

[8] Optimised query expression here indicates sophisticated reasoning over database model to reduce unnecessary SQL operations during the database access, e.g. SQL expression with key or non-key constraints may make huge difference to query execution when less intermediary data is used. This work is considered as part of further work.

Table 24 Summary of EDEN-IW solution for information integration

Requirements	***Solutions***
Query Transparency	Conceptual wrapping of data sources. Semantic mapping to generate SQL queries. Reasoning model to cover structural heterogeneity of different SQL tables due to mismatched integrity constraints and to cover conceptual heterogeneity. See section 4.36 and 4.3.7 for each.
Query internationalisation	Domain concepts are defined in several languages such as French, Danish, English and Italian in an online XML structured table compiled by an international domain standards organisation that is accessed when presenting information to the user
Query Augmentation	Reasoning about the semantic relations , viewpoint representation, user preference
Data harmonisation	Explicitness of metadata representation and reasoning functions.
Data Aggregation/presentation	The user viewpoint ontologies are linked to the core ontology model via mapping relations to support conceptual adaptation of data aggregation and presentation.
Application and Storage independence	Semantic model is partitioned to support this, see sections 4.3.8.4 and 5.5.2.
Metadata provenance	Concepts in the query can be related to terms defined by the international domain standards organisation online glossary.
Metadata structuring	There is an explicit process to handle metadata updates
Query flexibility to handle semiotic transparencies	Viewpoint projections from the common Ontology, user preference and terminology rules are used, see chapter 5.

The contribution of author's research deals with a more comprehensive resolution of information heterogeneities within a single knowledge domain, e.g. Inland Water. The proposed solution has fulfilled all IR requirements described in 4.2.4. The summarised features have been given in Table 24 that has illustrated the EDEN-IW solution fulfilling each requirement. The conceptual framework wraps, reconstructs and integrates heterogeneous data into semantic whole to facilitate user's understanding. Whereas other integration approaches focus on narrowed facets of data transformation or information mediation. For example, vocabulary issues of synonym problem were the major concern of integration projects such as Carnot [30], DOME [31], knowledge shifter [54], IF-MAP [50], and InfoSleuth [38],[13]. Semantic resolution in Observer

[64] and ONION [67] is restricted with pre-defined operations between concepts, TSIMMIS [28] identifies the meaning of concept using catenation names according to type hierarchies. Other database-based approach such as Xu and Embley [97] focus more on data transformation and less about orienting information modelling to users. The integration framework is regarded as a more holistic solution to harmonise data heterogeneities in practice.

6.1.2 A Semantic Approach to Support Multiple User Viewpoints

A multiple viewpoint model has been developed to enhance the data integration system to support information adaptation with respect to user groups or user stereotypes, and to individual preferences. This allows users to retrieve data in a conceptualisation that is better suited to them rather than to have to understand the entire detailed global view conceptualisation. Users can for example view data retrieval using more coarsely grained views or using a different coverage of concepts.

Two types of approaches have been proposed for viewpoint derivation, either an independent Ontology goes through adaptation process to become consistent with common Ontology, or it can be derived from existing viewpoint model. The advantage of the latter approach used in this PhD is that it reduces the number of potential conceptualisations and associated mapping to be more manageable. An extended Global-as-View or GAV approach has been applied to handle query reformulation over user views of the stored data.

Whereas an ad hoc framework based upon conventional distributed programming language and a rule framework could be used to support user views and adaptation to user views, a more formal framework has the benefit in that it can support reasoning about the consistency, equivalence, containment and conflict resolution when traversing data models. A semantic model that also supports a more expressive formal logic reasoning than that about the equivalent semantic conceptualisation of SQL model, which only supports Boolean logic operations, could be used to detect illogical data values if defined, e.g., pH for water should not normally be less than 5 or greater than 9; to detect properties that violate class and property constraints; to detect if constraints for non-class related relationships are violated, e.g., can't measure discharge in a fish. A semantic model for IR retrieval must also handle the fundamental differences between how a semantic model and how a database model views the world. One important difference is that the database model operates in a

closed world assumption and uses different kinds of integrity constrains to maintain data consistency whereas, semantic models often operate under an open world assumption, and consistency is managed differently. One simple way to avoid this is to reason about the semantic (open world) conceptualisation of the stored data model rather than to reason about the (closed world) stored data model itself

There are several possible designs for a formal model but a preliminary formulation of the formal model is based upon extending a Datalog type algebra with hierarchical, attribute and instance value operators. These operators can be applied to support compositional mapping and consistency checking of data views. The multiple viewpoint system was implemented as a Java-based application consisting of two sub-systems, one for viewpoint adaptation and management, the other for query processing and query result adjustment. The OWL Ontology is parsed and loaded into internal structure for operational processing using Jena, a Java-based Ontology parser with a limited inference capability for reasoning about OWL-Lite. More complicated logic inference can occur via in an external inference engine, Pellet.

6.2 Further work

As discussed earlier, by default queries answered in practice are based on a closed-world-assumption that is inherent in the relational data query language this. By default, reasoning about the semantic wrapper for the data source is based upon an open-world-assumption. The combination of closed and open world can introduced conflicts that can for example lead to inconsistencies in the close world. The case may typically happens when IR system try to answer a metadata query with implicit knowledge that maps to a database tuple set. A reasoning approach could be applied in the semantic model by introducing additional constraints into the semantic model to restrict the information retrieval in compliance with "close-world-assumption" This converts implicit incompleteness to explicit negation, such that a reconciliation between the closed world and open world semantics can be achieved.

Knowledge representation associated with mismatched data models may cause some information loss during data mappings used for data query process, e.g. synonym equivalence and containment may not be exact. A reasoning approach can analyse the differences between semantic correspondences and evaluate information loss during the query evaluation in the light of different data models. The computational model developed mainly considers terminology differences in both the linguistics synonym

and homonym naming and semantics. The similarity of non-exactly equivalent terminologies could be measured according to adjacent and hierarchy relations, in relation to a structured standard thesaurus. The semantics difference can be measured within the context of intensional and extensional information in the knowledge domain w.r.t. different structures and categories in terms of real instance sets. The accuracy can be evaluated using common IR factors such as precision and recall [36].

The graph-base routing algorithm can be further improved to fit into a distributed semantic model as an addition to logic framework, in order to solve non-logic reasoning about mismatched constraints. Such a routing algorithm can be used to better understand the implicit data retrieval semantics for the database model concerning the performance of query execution. An explicit expression of such information can help general use of such routing algorithm with different data model to give a better performance.

6.3 Main Conclusions

Information retrieval or IR from autonomous distributed databases within a domain, such as the inland water environmental quality domain, poses three complex IR problems: firstly, a complex data access problem; secondly a complex data interoperability problem and thirdly, a complex data modelling problem. Access is complex because the normal user access interface requires the user to understand both the structure and naming of the detailed stored data. This can vary for each different database in that domain, and the use of a data query interface such as SQL can vary. Interoperability is complex because there exists a range of heterogeneities in the stored data and a range of heterogeneous users and applications may access the data. In addition, an integrated data store such as a data warehouse is not feasible as the data collectors and owners do not want to give up control and maintenance of their data by giving wholesale copies of parts of their database to third-parties. Data modelling is a complex challenge not just because of the data, user and application heterogeneities but also because it is often difficult to construct a suitable single unified homogeneous data model in practice.

Use of an Ontological conceptualisation of metadata about the stored data, users and applications, a so called semantic approach to database interoperability is a powerful solution to handle these three problems. It can provide a higher-level, commonly understood or global conceptualisation of the stored data in relation to its associated

conceptualisation in the physical world. It thus supports access transparency by hiding the low-level access details to the individual database resources such as their data storage syntactic structure, and the identification process to locate a database and a data item within a database. It can also support different conceptualisations of the data that are adapted more to the conceptualisation understood by human users and the different conceptualisations used in application data processing.

Research into the development and application of a semantic model is very challenging and involves far more than simply creating a global view semantic model of a data domain and sharing this to support database, user and application interoperability. This is mainly because the elements of IR system can't often be reengineered to use a single global semantic model, instead a global semantic model is used as a mediating model to interoperate with multiple heterogeneous database models, user and application data models. This introduces complex data mapping problems when data queries are posed by different users and applications to combinations of database resources. A suitable semantic based framework solution has been shown to incorporate the following list of useful design features:

- A mediating Ontology model that is partitioned into a global domain conceptualisation model can interlink to:
    - Multiple database resource wrapper models that support conventional data query interfaces such as SQL and that can handle a range of data heterogeneities at the vocabulary, syntactic and semantic levels;
    - Multiple user models that better adapt to a user's presentation and internationalisation preferences, and a user's conceptualisation with respect to the coverage, granularity and the usage perspective.
- A Multi-agent framework to support the distribution, processing and reasoning about the uses of the semantic model, that handles different mappings within this partitioned Ontology model.
- An Ontology model that supports rule-processing and reasoning about data mappings to aid consistency and containment checking and conflict resolution and that can deal with not just direct synonym type data mappings but also indirect mappings that involve data instances and compositions as intermediaries.

The framework has been applied in part during the EU EDEN-IW, information retrieval and pollution monitoring for inland water, project. The system prototype consisted of 4 different national databases with more than 2 million records.

.

Bibliography

1. Foundation for Intelligent Physical Agents standards activity, Home Page, http://standards.computer.org/fipa/, accessed on 2005-12.
2. JADE (Java Agent Development Environment) Home page for an open-source FIPA MAS, http://sharon.cselt.it/projects/jade/, accessed on 2005-12.
3. Java agent Template "light", JATLite, Home page, accessed from 2003-05, http://java.stanford.edu/java_agent.html.
4. Jena- A Semantic Web Framework for Java, Home page accessed from 2003-09, http://jena.sourceforge.net/.
5. OKBC, Open Knowledge Base Connectivity, Home page accessed from 2005-12, http://www.ksl.stanford.edu/software/OKBC/,
6. On-To-Knowledge: Content-driven Knowledge-Management through Evolving Ontologies, Home page accessed from 2005-12 http://www.ontoknowledge.org/ .
7. Pellet,an open-source Java based OWL DL reasoner, Home page, http://www.mindswap.org/2003/pellet/index.shtml,accessed on 2006-01.
8. Protégé,A free, open source ontology editor and knowledge base framework, Home page, http://protege.stanford.edu, accessed from 2004-01, Stanford Medical Informatics.
9. Adnani, M.E., Ygtongnon, K. and Benslimane, D., A Multiple Layered Functional Data Model to Support Multiple Representations and Interoperability of GIS: Application to Urban Management Systems. in *Proceedings of the 9th ACM international symposium on Advances in geographic information systems*, (Atlanta, Georgia, USA), 2001, 70-75.
10. Alonso, G., Casati, F., Kuno, H. and Machiraju, V. *Web Services, Concept, Architectures and Applications*. Springer, 1998,
11. Baader, F. and Nutt, W. *Basic Description Logics*. Cambridge University Press, 2003,
12. Baldwin, D. Applying Multiple Views to Information Systems: A Preliminary Framework. *ACM SIGMIS Database*, *24* (4), 15-30.
13. Bayardo, R.J. and Bohrer, W., InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments. in *ACM SIGMOD International Conference on Management of Data (SIGMOD 1997)*, (Tucson, Arizona, USA), ACM, 1997, 195-206.
14. Bechhofer, S., The DIG Description Logic Interface: DIG/1.0, University of Manchester,Manchester, U.K.,http://dl-web.man.ac.uk/dig/2002/10/interface.pdf
15. Bechhofer, S., Harmelen, F.v., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A. and Olin, F.W., OWL Web Ontology Language Reference, W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-ref-20040210/
16. Bechhofer, S., Harmelen, F.v., Hendler, J., Horrocks, I., McGuinness, D.L.. Patel-Schneider, P.F., Stein, L.A. and Olin, F.W. OWL Web Ontology Language Reference,W3C Recommendation 10 February 2004, http://www.w3.org/TR/owl-ref/#Sublanguages.

17. Bechhofer, S. and Ng, G. OILEd, DAML+OIL editor, Home page. accessed from 2003-09, http://oiled.man.ac.uk/. University of Manchester, Manchester.
18. Benchikha, F. and Boufaida, M. Roles, Views and Viewpoints in Object-Oriented Databases *Workshop at European Conference on Object-Oriented Pragamming (ECOOP 2005)*, Glasgow, UK., 2005.
19. Berners-Lee, T. WWW past & future, 2003. Accessed from 2006-05, http://www.w3.org/2003/Talks/0922-rsoc-tbl/.
20. Berners-Lee, T., Hendler, J. and Lassila, O. The Semantic Web. *Scientific American, 2001 May*.
21. Bertino, E., Catania, B. and Zarri, G.P. *Intelligent Database Systems*. Addison-Wesley, 2001,170-178.,
22. Bouquet, P., Ehrig, M., Euzenat, J., Franconi, E., Hitzler, P., Krötzsch, M., Serafini, L., Stamou, G., Sure, Y. and Tessaris, S., Knowledge Web Project Deliverable Version 2.2.1, Specification of a common framework for characterizing alignment, Home page, accessed from 2005-02, http://knowledgeweb.semanticweb.org
23. Boyd, M., Sasivimol, Kittivoravitkul, Lazanitis, C., McBrien, P. and Rizopoulos, N., AutoMed: A BAV Data Integration System for Heterogeneous Data Sources. in *the 16th International Conference on Advanced Information Systems Engineering*, (Riga, Latvia), Spring-Verlag, 2004, 82-97.
24. Brachman R.J., Borgida A., McGuinness D.L., Patel-Schneider P.F. and Resnick L. Alperin The CLASSIC Knowledge Representation System, or KLONE, The Next Generation. *Special issue on implemented knowledge representation and reasoning systems*, *2* (3), 45-56.
25. Bradshaw J.M., Dutfield S. and Benoit P. KAoS: towards an industrial strength open agent architecture. *Software agents*, AAAI Press, 375-418.
26. Calvanese, D., Giacomo, G.D. and Lenzerini, M., A Framework for Ontology Integration. in *Proceedings of the 2001 Int. Semantic Web Working Symposium*,2001, 303-316.
27. Calvanese, D., Lembo, D. and Lenerini., M., Survey on Methods for Query Rewriting and Query Answering Using Views, University of Rome,Roma,http://citeseer.ist.psu.edu/cache/papers/cs/31839/http:zSzzSzwww.dis.uniroma1.itzSz~lembozSzD2IzSzProdottizSz.zSzdeliverablezSzD1.R5.pdf/calvanese01survey.pdf
28. Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J. and Widom, J., The TSIMMIS Project: Integration of Heterogeneous Information Sources. in *Proceedings the 10th Meeting of the Information Processing Society of Japan*, (Tokyo, Japen), 1994, 7-18.
29. Codd, E.F. A Relational Model for large data banks. *Communications of the ACM, 13* (6), 377-387.
30. Collet, C., Huhns, M.N. and Shen, W. Resource Integration Using a Large Knowledge Base in Carnot. *IEEE Computer*, *24* (12), 55-62.
31. Cui Z., Jones, D. and O`Brien, P. Semantic B2B Integration: Issues in Ontology-based Approaches. *ACM SIGMOD,SPECIAL ISSUE: Data management issues in electronic commerce table of contents*, *31* (1), 43-48.
32. Damasio, C.V., Analyti, A., Antoniou, G. and Wagner, G., Supporting Open and Closed World Reasoning on the Web. in *Proceedings of Principles and Practice of Semantic Web Reasoning (PPSWR06)*, (LNCS), Springer, 2005, 21-36.
33. Das, S. *Deductive Databases and Logic Programming*. Addison-Wesley, 1992.

34. De Bruijn, J., Eiter, T., Polleres and A. Tompits H. On Representational Issues about Combinations of Classical Theories with Nonmonotonic rules. DERI Technical Report, available from http://www.deri.at/digital-library/browse/technical-reports/.
35. Ferber J. *Multi-Agent Systems*. Addison-Wesley, 1999,
36. Fernández-López, M. and Gómez-Pérez, A., Ontoweb Deliverable D1.4: A survey on methodologies for developing, maintaining, evaluating and re-engineering ontologies., http://ontoweb.org/About/Deliverables/D1.4-v1.0.pdf
37. Foster, I., Kesselman, C. and Tuecke, S. The Anatomy of the Grid: Enabling Scalable. *Virtual Organizations. Int. J. High Performance Computing Applications*, *15* (3), 200-222.
38. Fowlery, J., Nodine, M., Perryy, B. and Bargmeyerz, B. Agent-Based Semantic Interoperability in InfoSleuth. *ACM SIGMOD,SPECIAL ISSUE: Data management issues in electronic commerce table of contents*, *28* (1), 60-67.
39. Garlan D. and Shaw M., An Introduction to software architecture. in *Advances in Software Engineering and Knowledge Engineering Review*, (Singapore), World Scientific Publishing Company, 1993, 1-39.
40. Genesereth, M.R. and Fikes, R.E., et al., Knowledge Interchange Format Version 3 Reference Manual, Logic-92-1, Stanford University Logic Group,http://logic.stanford.edu/kif/Hypertext/kif-manual.html
41. Haastrup, P. and Wuertz, J., DELIVERABLE D16: Conceptual Design of Software Agents, The Environmental Data Exchange Network for Inland Water (EDEN-IW) project, available from http://www.eden-iw.org,
42. Hendler, J. Agents and the Semantic Web. *IEEE Intelligent Systems Journal*, *16* (2), 30-37.
43. Horrocks, I., Parsia, B., Patel-Schneider, P. and James Hendler, Semantic web architecture: Stack or two towers? in *Principles and Practice of Semantic Web Reasoning (PPSWR 2005)*, (SV), 2005, 37-41.
44. Horrocks, I. and Patel-Schneider, P.F. Reducing OWL Entailment to Description Logic Satisfiability. *Journal of Web Semantics*, *1* (4), 10.
45. Horrocks, I. and Sattler, U., A tableaux decision procedure for SHOIQ. in *Proceedings of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*,2005, 448-453.
46. Imhoff, C., Galemmo, N. and Geiger, J.G. *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. John Wiley & Sons Inc, 2003,
47. Jennings, N.R., Sycara, K. and Wooldridge, M. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems Journal*, *1* (1), Kluwer Academic Publishers, 7-38.
48. Jon, B. and Seligman, J. *Information Flow: The Logic of Distributed Systems*. Cambridge University Press, 1997,
49. Jung, E.-C., Sato, K., Chen, Y., He, X., MacTavish, T. and Cracchiolo, D., DIF Knowledge Management System: Bridging Viewpoints for Interactive System Design. in *Proceeding 11th Human Computer Interaction International Conference*, (Las Vegas, Nevada USA), 2005.
50. Kalfoglou, Y. and Schorlemmer, M. IF-Map: An Ontology-Mapping Method based on Information-Flow Theory. *Journal of Data Semantics*, *1* (1).
51. Kalfoglou, Y. and Schorlemmer, M. Ontology mapping: the state of the art. *Knowledge Engineering Review*, *18* (1), Cambridge University Press, 1-31.

52. Kashyap V. and Sheth A.P., Semantics-based Information Brokering. in *Proceedings the 3rd International Conference on Information and Knowledge Management (CIKM)*,1994.
53. Kashyap V. and Sheth A.P., Semantics-based Information Brokering. in *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM)*,1994.
54. Kerschberg, L., Chowdhury, M., Damiano, A., Jeong, H., Mitchell, S., Si, J. and Smith, S., Knowledge Sifter: Agent-Based Ontology-Driven Search over Heterogeneous Databases using Semantic Web Services. in *International Conference on Semantics of a Networked World*, (Paris, France), 2004, 278-295.
55. Klein, M., Fensel, D., Kiryakov, A. and Ognyanov, D., Ontology Versioning and Change Detection on the Web. in *Proceedings of the 13th European Conference on Knowledge Engineering and Management, EKAW-2002*, (Madrid, Spain), Springer, 2002, 197-212.
56. Léger, A., Yannick Bouillon, Ecoublet, P., Dieng, R., Persidis, A., Sure, Y., Gomez, A., López, F. and Ding, Y., ONTOWEB project deliverable D2.2-Successful scenarios for ontology-based applications, http://ontoweb.org/About/Deliverables//D22-final_final.pdf
57. Lenzerini, M., Data Integration: A Theoretical Perspective. in *the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS02)*,ACM Press, 2002, 233--246.
58. Li, W.-S. and Clifton, C. SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data & Knowledge Engineering*, *33* (1), 49-84.
59. Lord, P., Stevens, R.D., Goble, C.A. and Horrocks., I. *Description Logics: OWL and DAML+OIL, In Genetics, Genomics, Proteomics, and Bioinformatics.* Wiley, 2004.,
60. Madhavan, J., Bernstein, P.A. and Rahm, E., Generic Schema Matching with Cupid. in *Proceedings of the 27th VLDB Conference*, (Roma, Italy), 2001, 48-58.
61. Martin D.l., Cheyer A.J. and Moran D.B., Building distributed software systems with the open agent architecture. in *Proceedings of Conference of The Practical Application of Intelligent Agents and Multi-Agents (PAAM98)*,1998, 355-376.
62. McBrien, P. and Poulovassilis, A., Data Integration by Bi-Directional Schema Transformation Rules. in *19th International Conference on Data Engineering (ICDE'03)*,2003, 227.
63. Mena, E. and Illarramendi, A. *Ontology-Based Query Processing for Global Information Systems*. Springer, 2001,
64. Mena, E., Illarramendi, A., Kashyap, V. and Sheth, A.P. An Approach for Query Processing in Global Information systems Based on Interoperation Across Pre-Existing Ontologies. *International Journal Distributed and Parallel Databases (DAPD)*, *8* (2), 223-271.
65. Meyden, R.v.d. *Logical approaches to incomplete information*. Kluwer Academic Publisher, 1998, 307-356
66. Michael Stonebraker, Paul Brown and Martine Herbach Interoperability, Distributed Applications and Distributed Databases: The Virtual Table Interface. *IEEE Data Eng. Bull.*, *21* (3), 25-33.

67. Mitra, P., Wiederhold, G. and Decker, S., A Scalable Framework for the Interoperation of Information Sources. in *the 1st International Semantic Web Working Symposium (SWWS `01)*, (Stanford University, Stanford, CA), 2001.
68. N. Noy and M. Musen, PROMPT: Algorithm and tool for automated ontology merging and alignment. in *the 17th National Conference on Artificial Intelligence (AAAI'00)*, (Austin, Texas, USA), 2000.
69. Ng., G., Open vs Closed world, Rules vs Queries: Use cases from Industry. in *OWL experiences and directions workshop*, (Galway, Ireland), 2005.
70. Nodine, M. and Fowler, J., On the Impact of Ontological Commitment. in *Proceedings of 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, (Bologna, Italy), 2002, 21.
71. Nodine M., Fowler J. and Perry B., Active information gathering in InfoSleuth. in *Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications CODAS*,1999, 15-26.
72. Nodine M. and Unruh A., Facilitating open communication in agent systems: the InfoSleuth Infrastructure. in *Processing of Intelligent Agents Conference IV*, (ATAL), 1998, 281-296.
73. Noy, N.F. and Klein, M. Ontology Evolution: Not the Same as Schema Evolution. Knowledge. *Information Systems*, *6* (4), 428-440.
74. Pan, Z. and Heflin, J., DLDB:Extending Relational Databases to Support Semantic Web Queries. Technical Report LU-CSE-04-006, Dept. of Computer Science and Engineering, Lehigh University,http://www.cse.lehigh.edu/~heflin/pubs/psss03-poster.pdf
75. Patel-Schneider, P.F. and Horrocks I., A Comparison of Two Modelling Paradigms in the Semantic Web. in *Proceedings of the Fifteenth International World Wide Web Conference (WWW 2006)*,ACM, 2006, 3-12.
76. Poole J., Chjang D., Tolbert D. and Mellor D. *Common Warehouse Metamodel.* John Wiley & Sons, Inc., New York, 2002,
77. Poslad S. and Charlton P. Standardizing agent interoperability: the FIPA approach. *Multi-Agent Systems and Applications*, Lecture Notes CS, 98-117.
78. Pottinger, R. and Halevy, A. MiniCon: A scalable algorithm for answering queries using views. *The Very Large DataBase Journal*, *10*, 182-198.
79. Raskin J. *The Humane Interface: New Directions for Designing Interactive Systems.1st edition.* Addison-Wesley Pub Co; ISBN: 0201379376, 2000,
80. Ribière, M. and Dieng-Kuntz, R., A Viewpoint Model for Cooperative Building of an Ontology. in *Proceeding of 10th International Conference on Conceptual Structures (ICCS 2002)*, (Borovets, Bulgaria), Springer Berlin / Heidelberg, 2002, 220-234.
81. Russell, S. and Norvig, P. *Artificial Intelligence: a modern approach*. Pearson Education International, New Jersey, 2003,
82. Searle, J.R. *Speech Acts.* Cambridge University Press, 1969.,
83. Semeraro, G., Degemmis, M., Lops, P. and Palmisano, I., WordNet-based User Profiles for Semantic Personalization. in *Proceedings of the Workshop on New Technologies for Personalized Information Access (PIA 2005), part of the 10th Int. Conf. on User Modeling (UM'05)*, (Edinburgh, Scotland, UK), 2005. 74-83.
84. Shadbolt, N., Berners-Lee, T. and Hall, W. The Semantic Web Revisited. *IEEE Intelligent Systems*, *21* (3), 96-101.
85. Sheth, A. and Larson, J. Federated Database Systems for Managing Distributed. Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*. *22* (3), 183-230.

86. Sheth, A.P. *Changing Focus on Interoperability In Information Systems: from System, Syntax, Structure to Semantics.* Kluwer Academic Publishers, Dordrecht, Netherlands, 1998, 5-29
87. Singh, M.P. and Huhns M.N. *Service oriented Computing*. Wiley, 2005,
88. Sowa, J.S., Building, Sharing and Merging Ontologies, Accessed from 2005-12, http://www.jfsowa.com/ontology/ontoshar.htm,
89. Spaccapietra, S., Parent, C. and Vangenot, C., GIS Databases: From Multiscale to MultiRepresentation. in *Proceedings 4th InternationalSymposium, SARA-2000*, (Horseshoe Bay, Texas, USA), 2000, 57-70.
90. Stjernholm, M., Preux, D., Sortkjaer, O. and Zuo, L., DILIVERABLE 17,Structured list integration into IW Distributed Semantic Hybrid Agents application,The Environmental Data Exchange Network for Inland Water (EDEN-IW) project, available from http://www.eden-iw.org,
91. Tsichritzis, C. and Klug, A.e. The ANSI/X3/SPARC DBMS Framework: Report of the Study Group on Data Base Management Systems. *Information Systems*, *3*, 173-191.
92. UK Enviornmental Agency. GQA methodologies for the classification of river and estuary quality. Available from http://www.environment-agency.gov.uk., 2000.
93. Visser, U. and Schuster, G., Finding and Integration of Information- A Practical Solution for the SemanticWeb. in *Proceedings of ECAI 02, Workshop on Ontologies and Semantic Interoperability*, (Lyon, France), 2002, 73-78.
94. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. and Hübner, S., Ontology-Based Integration of Information -A Survey of Existing Approaches. in *Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing*, (Seattle, WA), 2001, 108-117.
95. Weiss G. (Ed.) *Multi-Agent Systems: A Modern Approach to Artificial Intelligence*. MIT Press, 2000,
96. Wiesman, F. and Roos, N., Domain independent learning of ontology mappings. Proc. of *the Third International Joint Conference on Autonomous Agents and Mutli-Agent Systems (AAMAS 2004)*,2004.
97. Xu, L. and Embley, D.W., Combining the Best of Global-as-View and Local-as-View for Data Integration. in *Proceedings of ISTA 2004: 3rd International Conference on Information Systems Technology and its Applications*, (Salt Lake City, Utah, USA), 2004, 123-136.
98. Xu, L. and Embley, D.W., Discovering Direct and Indirect Matches for Schema Elements. in *Proceedings of the Eighth International Conference on Database Systems for Advanced Applications*, IEEE Computer Society, 2003, 39.
99. Zaniolo, C., The Logical Data Language (LDL): An Integrated Approach to Logic and Databases", MCC Technical Report STP-LD-328-91,
100. Zuo L, P.S., Supporting multi-lateral semantic information viewpoints when accessing heterogeneous distributed environmental information. in *1st European workshop on multi-agent systems, EUMAS*, 2003.

Appendix I Author's Publications

1. Poslad S, Tan JJ, Zuo L, and Huang X. Middleware for semantic based security and safety management for Web applications. Int. Journal of Web and Grid Services, Vol. 1, No. 3/4, pp 305 – 327, 2005.
2. Stjernholm M., Poslad S., Zuo L., Sortkjær O., Huang X.. The EDEN-IW Ontology model for sharing knowledge and water quality data between heterogeneous databases. 18th conf. EnviroInfo 2004 of German Informatics Society (GI), Geneva, Switzerland, 21st-23rd October, 2004.
3. Zuo L, Poslad S. Supporting multi-lateral semantic information viewpoints when accessing heterogeneous distributed environmental information. EUMAS-2003, EU Multi-Agent System Conf., Oxford, UK, December 2003.
4. (Accepted 2006)Zuo L, Poslad S., and Huang X. Agent-based Information Sharing and Semantic Information Retrieval from Heterogeneous Databases, International Conference of Business Knowledge Management, Macao, 26th-28th October, 2006.
5. (Accepted 2006)Zuo L, Poslad S., A Dynamic Semantic Framework to Support Multiple User Viewpoints during Information Retrieval, 1st International Workshop on Semantic Media Adaptation and Personalization (SMAP 2006), Athens, Greece, 4th-5th December 2006.
6. (Accepted 2005) Poslad, S., Zuo, L Huang, X. Agent Technology in the Environmental Data Exchange Network for Inland Water project. In: Environmental Data Exchange Network for Inland Water, Haastrup P. , Wurtz J. (Eds), Elsevier, 2006.
7. (Accepted 2005) Stjernholm M, Poslad, S., Zuo, L, Sortkjær O, Huang, X. An Ontology Based Approach for Enhancing Inland Water Information Retrieval from Heterogeneous Databases. In: Environmental Data Exchange Network for Inland Water, Haastrup P. , Wurtz J. (Eds), Elsevier, 2006
8. (Accepted 2005) Poslad, S., Stjernholm M, Zuo, L Huang, X. Review of models and technologies for database integration. In: Environmental Data Exchange Network for Inland Water, Haastrup P. , Wurtz J. (Eds), Elsevier, 2006.
9. Stjernholm, M., Preux, D., Sortkjaer, O. and Zuo, L. DELIVERABLE 12, Distributed Semantic Hybrid Agents application, The Environmental Data Exchange Network for Inland Water (EDEN-IW) project, available from http://www.eden-iw.org, 2006
10. Stjernholm M., Zuo L. and Poslad S. DELIVERABLE 18, Ontology Files, The Environmental Data Exchange Network for Inland Water (EDEN-IW) project, available from http://www.eden-iw.org, 2006
11. Poslad S., Zuo L. and Huang X., DELIVERABLE 23, Final Report for WP1, The Environmental Data Exchange Network for Inland Water (EDEN-IW) project, available from http://www.eden-iw.org, 2006

Appendix II The RDF schema for SQL query representation

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/TR/2004/REC-rdf-schema-20040210/
      xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">
 <rdfs:Class rdf:ID="Query">
  <rdfs:label xml:lang="en">Query</rdfs:label>
  <rdfs:label xml:lang="fr">Query</rdfs:label>
  <rdfs:subClassOf rdf:resource=
   "http://www.w3.org/TR/2004/REC-rdf-schema-20040210#Statement"/>
  <rdfs:comment>This describes the set of queries</rdfs:comment>
 </rdfs:Class>
<rdfs:ConstraintProperty rdf:ID="ConversationID">
  <rdfs:label xml:lang="en"> ConversationID </rdfs:label>
  <rdfs:domain rdf:resource="#Query"/>
  <rdfs:range rdf:resource=
   "http://www.w3.org/TR/2004/REC-rdf-schema-20040210#Literal"/>
</rdfs:ConstraintProperty>
<rdfs:ConstraintProperty rdf:ID="Status">
 <rdfs:label xml:lang="en">status</rdfs:label>
 <rdfs:domain rdf:resource="#Query"/>
 <rdfs:range rdf:resource=
  "http://www.w3.org/TR/2004/REC-rdf-schema-20040210#Literal"/>
</rdfs:ConstraintProperty>
<rdfs:Class rdf:ID="Action">
 <rdfs:label xml:lang="en">action</rdfs:label>
 <rdfs:label xml:lang="fr">action</rdfs:label>
 <rdfs:subClassOf rdf:resource=
  "http://www.w3.org/TR/2004/REC-rdf-schema-20040210#Resource"/>
 <rdfs:comment>This describes the set of actions</rdfs:comment>
</rdfs:Class>
 <rdfs:ConstraintProperty rdf:ID="act">
 <rdfs:label xml:lang="en">act</rdfs:label>
```

```
  <rdfs:label xml:lang="fr">acte</rdfs:label>
  <rdfs:domain rdf:resource="#Action"/>
</rdfs:ConstraintProperty>
 <rdfs:ConstraintProperty rdf:ID="actor">
  <rdfs:label xml:lang="en">actor</rdfs:label>
  <rdfs:label xml:lang="fr">acteur</rdfs:label>
  <rdfs:domain rdf:resource="#Action"/>
</rdfs:ConstraintProperty>
 <rdfs:ConstraintProperty rdf:ID="argument">
  <rdfs:label xml:lang="en">argument</rdfs:label>
  <rdfs:label xml:lang="fr">argument</rdfs:label>
  <rdfs:domain rdf:resource="#Action"/>
</rdfs:ConstraintProperty>
 <rdfs:ConstraintProperty rdf:ID="done">
  <rdfs:label xml:lang="en">done</rdfs:label>
  <rdfs:label xml:lang="fr">fini</rdfs:label>
  <rdfs:domain rdf:resource="#Action"/>
 </rdfs:ConstraintProperty>
 <rdfs:Class rdf:ID="result">
  <rdfs:label xml:lang="en">result</rdfs:label>
  <rdfs:label xml:lang="fr">resultat</rdfs:label>
  <rdfs:domain rdf:resource="#Query"/>
  <rdfs:subPropertyOf rdf:resource=
    "http://www.w3.org/TR/2004/REC-rdf-schema-20040210#Seq"/>
  <rdfs:comment>This describes the set of query result</rdfs:comment>
</rdfs:Class>
 <rdfs:Class rdf:ID="Rule">
   <rdfs:label xml:lang="en">rule</rdfs:label>
   <rdfs:label xml:lang="fr">regle</rdfs:label>
</rdfs:Class>
 <rdfs:ConstraintProperty rdf:ID="selection">
  <rdfs:comment>The selection part </rdfs:comment>
  <rdfs:domain rdf:resource="Rule"/>
</rdfs:ConstraintProperty>
```

```
 <rdfs:ConstraintProperty rdf:ID="selection-result">
<rdfs:comment>
  Identifies the container filled with selection results
</rdfs:comment>
<rdfs:domain rdf:resource="Result"/>
<rdfs:range rdf:resource=
  "http://www.w3c.org/TR/1999/PR-rdf-schema-19990303#Bag"/>
</rdfs:ConstraintProperty>
</rdf:RDF>
```