EnviroInfo 2004 (Geneva)
Sh@ring – EnviroInfo 2004
 ISBN: 28 29 30 275-3

# The EDEN-IW Ontology Model for Sharing Knowledge and Water Quality Data between Heterogeneous Databases

Michael Stjernholm[1], Stefan Poslad[2], Landong Zuo[2], Ole Sortkjær[1] and Xuan Huang[2]

**Abstract**

The Environmental Data Exchange Network for Inland Water (EDEN-IW) project's main aim is to develop a system for making disparate and heterogeneous databases of Inland Water quality more accessible to users. The core technology is based upon a combination of: ontological model to represent a Semantic Web based data model for IW; software agents as an infrastructure to share and reason about the IW semantic data model and XML to make the information accessible to Web portals and mainstream Web services. This presentation focuses on the Semantic Web or Ontological model. Currently, we have successfully demonstrated the use of our systems to semantically integrate two main database resources from IOW and NERI – these are available on-line. We are in the process of adding further databases and supporting a wider variety of user queries such as Decision Support System queries.

## 1. Introduction

Environmental Data Exchange Network for Inland Water (EDEN-IW) is a project supported by the EU-IST 5th framework research program (EDEN-IW, 2004). The project's main aim is to develop a system for making disparate and heterogeneous databases of Inland Water (IW) quality more accessible to users. The core technology is based upon a combination of: ontological model to represent a semantic data model (Fensel, 2003) for IW; software agents as an infrastructure to share and reason about the IW semantic data model (Poslad, 2003); XML to make the information accessible to Web portals and mainstream Web services.

[1] National Environmental Research Institute, Department of Freshwater Ecology, P.O. Box 314, DK-8600 Silkeborg e-mail: msh@dmu.dk
[2] University of London, Queen Mary, Department of Electronic Engineering, Mile End Road London E1 4NS e-mail: Stefan.Poslad@elec.qmul.ac.uk

The EDEN-IW project focuses the development efforts within the restricted domain of surface water and more specifically stream water data, although the concepts behind EDEN-IW may be applied to other domains of water and to the environment.

## 2. Inland Water Data Application Requirements

Databases of inland water quality has been established and maintained over decades. The databases and their structure reflect the business processes of the organization that has maintained the databases. A majority of these databases have been established long before distributed services such as “public access”, Web services” and “e-government” were envisaged.

Inland water databases contain core concepts such as ”the VALUE of DETERMINANT observed at a STATION at a TIME or over a TIMEPERIOD”. The EDEN-IW project currently only handles instantaneous observations.

### 2.1 Use cases for expected data services

The EDEN-IW project has identified the following core queries:

- Concentration of substance X at monitoring station Y during the period Z.
- Stations having determinant X observed above a threshold value Y during period Z.
- Values of determinant X at station Y compared to determinant Z at station U.

Some of the use cases are fairly simple whereas others are more challenging. The third use case potentially involves data from two different databases and thus puts some requirements on having a similar understanding of the concepts involved. Further use cases have also been defined; these include the use of GIS information to locate monitoring stations and Decision Support System type queries.

### 2.2 Database organization

Data retrieval is commonly organized using relational database systems and normalized tables but meta-data, other than the primitive data types used for table columns, are often not available on-line or standardized. Meta data concepts are often better represented in non-relational type models such as object-oriented model classes with attributes, sub-classes and relationships to other classes.

Although the main concepts may be commonly accepted, local implementations can vary substantially. Similar observations may be handled differently in different database implementations. The result of the differences between databases, e.g.,

Table 1, is that a combined search and presentation data from different databases requires additional concepts and primitive classes to be defined.

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

Table 1: Different implementations of observations in a French (IOW) and a Danish (NERI) database.

## 3. EDEN-IW System Overview

There are several main techniques for integrating database resources: Web portals, syntactical integration using global schema or federated schema and Semantic integration (Zuo, 2003). The EDEN-IW System uses a semantic mediation approach to integrate disparate inland water database resources and also to integrate heterogeneous types of users and their applications. There are several advantages to using a semantic integration approach. It can support query augmentation (expansion of a user query using the context); content harmonization when information sources differ; content aggregation and content management using the semantic model to classify, (re)structure and index information.

There are several information integration projects related to environmental data. These included data capture projects such as SUMARE and SEWING, data integration and modeling projects such as CoastBase, GIMMI, MERMAID and I-MARQ and Citizen directed projects such as APNEE-TU (EU-CORDIS, 2004). EDEN-IW project is the only current environment project that focuses on inland water that adopts a semantic approach to data integration and user and application integration and is concerned with the whole data life-cycle, including data acquisition, data processing and data management.

## 3.1 Architecture of EDEN-IW system

EDEN-IW is a system that semantically integrates different inland water information with intelligent agents being the distributors of the semantics. The agent architecture of the EDEN-IW system is a Multi-Agent system (MAS) build on the FIPA[3] standards implemented using the JADE[4] agent tool-kit. The Jena[5] Semantic Web tool-kit is used to import, parse and export semantic messages. At a high level of abstraction, the system appears to users as two web portals: one for the end user to specify their queries into the system and the other to connect to the distributed Databases. The core agent system together with the ontology service sits in between the two portals, sharing and reasoning about the semantics, hiding the complexity from the end user, and providing a homogeneous query interface to heterogeneous databases. Table 2 summarizes the agent roles and functions.

| **Directory Agent** | Manager of meta data information from agents and resources |
|---|---|
| Function:<br>• Register, modify and deregister agent and service descriptions<br>• Search the directory for agent and service description matches. | |
| **Task Agent** | Broker between user, resources and directory |
| Function:<br>• Broker queries and schedule tasks on behalf of the user<br>• Data fusion such as combining results from multiple information resources | |
| **User Agent** | Interface between user and core agent system |
| Function:<br>• Guide user to iteratively specifying queries to databases<br>• Present results returned from the queries | |
| **Resource Agent** | Interface between core agent system and database |
| Function:<br>• Query the database and return results from the database query<br>• Provide metadata such as the database schema to the EDEN-IW directory agent<br>• Translate data between the EDEN-IW semantic layer to the EDEN-IW XML syntactical layer. | |

Table 2: Agent roles and function

[3] http://www.fipa.org/
[4] http://jade.tilab.com/
[5] http://www.hpl.hp.com/semweb/

The EDEN-IW system uses Web services for making the global view ontology and interface descriptions globally accessible. The user Web portal includes GIS functionality based on a Mapserver[6] service.

## 4. EDEN-IW Semantic Model & services

Developing and deploying a semantic approach to integrating data resources and users, whilst being very powerful, faces several key challenges: the semantic model needs to be defined, it needs to be logically and physically accessible via a variety of different viewpoints according to different application and user requirements and a life-cycle model is required to allow the semantic model to adjust to meet changing user requirements.

The core of the design is to separate the conceptual model from the application and user commitments to use the conceptual model. This has the advantage of making the conceptual model more reusable and maintainable across applications.

The core conceptual model, called the EDEN-IW global view ontology or EGV model, defines all the concepts necessary for formulating the world of inland water quality. In addition, each database resource, user and application defines a separate data model that is mapped to the global model, e.g., each of the connected databases define a local database view ontology (LDV) based on the EGV model. The classes and facets of the LDV ontology will be sub-classes and aggregations of the EGV set of classes and facets. Ontology services perform the knowledge mediation between the EGV and LDV models and between the EGV and other data models such as: multi-lingual data dictionaries, standard glossaries of terms and user preferences such as their preferred constraints for queries and metadata queries. The EGV ontology model serves several purposes:

- It provides a common data dictionary - definitions, concept names and enumeration's of e.g. determinants and units.
- It provides the basic classes (primitives) for creating local concept classes.
- It provides the schema for required information on 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 organization for common knowledge about relationships among the classes.

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. A common problem is that key information may not be expressed ex-

[6] http://mapserver.gis.umn.edu/

plicitly but remains as "implicit common knowledge" within a local group of data managers. Some of the documentation may reside in printed documentation separated from the databases. Computer processing and machine deduction requires that all knowledge about the data is specified and stored in a computer understandable way such as SQL, RDF, DAML[7] and OWL[8].

## 4.1 Global view ontology

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 current 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 organized in hierarchies, with EdenGlobalConcept as a superclass. The EGV also contain relevant instances of the defined classes.

The Inland water databases contain information of the type "the VALUE of DETERMINANT observed at a STATION at a TIME". A deeper analysis of the concept of "the VALUE of DETERMINANT" in a couple of databases has identified that the value of a determinant may actually express different types of information:

1. Instantaneous values vs. time-aggregated values.
2. The same determinant may be observed in different media and fractions of the medium.
3. The value may be expressed with different units e.g. milligram/litre or nanogram/litre.
4. The value 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 determinants that supports these design requirements, see figure 1.

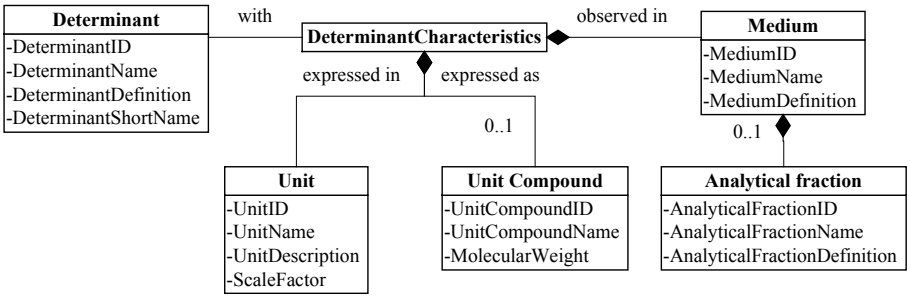


Figure 9: EGV representation of determinants and associated classes

[7] http://www.daml.org/

[8] http://www.w3.org/TR/owl-ref/

The use of any representation of a data model necessitates conforming to restrictions of the constraints in expressivity of that particular data representation, e.g., the Web Ontology Language (OWL) does not allow instances to have specific relationships to other instances. Since the class definitions in OWL are templates for instances, OWL does not allow classes to have a property set a given value, though the class may be defined by the fact that a property has a certain value. This has led to a design of the EGV model where determinants can have facets both as a class and as an instance. An example of this can be taken from the domain of Determinants and seen in figure 2.

```
<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>
    <Ideterminants rdf:ID="nitrate">
     <DeterminantID>19</DeterminantID>
     <DeterminantDef>Nitrogen in the form of NO3- </DeterminantDef>
     <EDENTermID>1014</EDENTermID>
```

1. Figure 2: OWL representation of Nitrate as a class defined by the Ideterminant instance “nitrate“

The class “Ideterminants” has an instance “nitrate” with a set of properties:(formula, definition etc). “Nitrate” is a subclass of “Nitrogens_Oxided” and defined by the property “hasIdeterminant” having exactly the value of the instance “nitrate”.

OWL allows a certain class to be a sub-class of several different classes. The class-hierarchy is thus not exclusive. In the example above the class “Nitrate” may also be a subclass of the “DeterminantList” subclass “Nutrients”.

Additional design requirements relate to the ability to convert between different units. One problem with units in the relation 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 SI 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 type they are instances of. “ConcentrationUnits” is a subclass of “FractionUnits”, which is specified to have both a numerator and a divisor.

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.

## 4.2 Ontology services

The current version of ontology service was designed to parse the EDEN-IW multi-lateral ontology knowledge model in order to deduce the logic relations and retrieve the domain information. The ontology service is a collection of Java applications that were developed on the top of the ontology parser implanted using Jena - a Java framework for building Semantic Web applications. Jena 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 as this was the most mature semantic model. As the project progressed, support for OWL became more mature, the project focused more on OWL.

The objective of building the ontology service is to improve the reusability and efficiency of the ontology access from different users. Although, the initial focus was on content harmonization. Further developments have focused on use of the ontology mode via multi-lateral user and multiple application viewpoints. Using the EGV concepts and relations, EDEN-IW users can easily build the user query according to the specific interests of a information fraction in the knowledge domain. EGV instances were created in order to maintain the value mapping within the particular EGV class, for example mapping between determinantID to determinantName. Additionally, aggregation relations were introduced into EGV model to give the explicit definition of the particular domain logic. For example Observation is an aggregation of several other classes such as Determinant, Medium, and AnalyticalFraction.

### 4.2.1 Local database view ontology and services

The LDV model consists of database schema and local logic concepts, derived from the EGV concepts so that the LDV can be mapped to EGV. The representation of the LDV model in DAML adopts several generic rules to ensure the scalability of LDV model and reusability of ontology service. Each table in a database is abstracted into a subclass of the common super-class concept of Table. Each column is abstracted into a sub-property of the common super-property field. A Foreign Key relation is presented using the DAML tag of SamePropertyAs. The Primary Key is

marked using the DAML tag of UnambiguousProperty. By browsing the DAML tags, the ontology services can easily inference about the data model.

Mapping relationships need to be set up between the EGV and LDV data views. SamePropertyAs was used to mark the semantic equivalent properties crossing the two models. The mapping includes three categories:

- *Direct mapping from database column*: The direct mapping is applied in the condition that the EGV property has a direct equivalent column in database schema, no processing is needed to proceed the logic or mathematic conversion.
- *Value conversion*: Value conversion is applied when the EGV property has the same semantic meaning as LDV property, but the direct mapping could not be established due to the problem of different coding format and value representation between EGV and LDV terms. In this case, a local logic concept was introduced to inherit the EGV concept and provide the value mapping or processing. Direct mapping is needed to map logic property to the database column. For example, due to the different ID coding of EGV determinant and IOW determinant, IOWDeterminant is created in IOW LDV as the subclass of Determinant in EGV, IOWDetermint consists of DetermintID, which comes from EGV value and IOWDeterminantID.
- *Logic conversion*: Value conversion does not resolve all mapping problems. Logic conversion are required in the case of more complex logic mapping between LDV and EGV concepts. The complex LDV concept has different semantic meanings that can be mapped to a EGV concept directly. Normally, it can be represented as the logic combination consisting of several EGV concepts. For example, in NERI LDV, a local logic concept NERIObservationCharacteristic was used to abstract the concept of NERI Inland Water Observation that contains the value combination from several EGV concepts such as Determinant, Medium, AnalyticalFraction and Unit.

From the Cardinality view, we can mark direct mapping, value mapping as a 1-1 mapping between two ontology views, then logic conversion mapping involves more complex relations of one-to-many mappings.

### 4.2.2 Content harmonization

The multi-lateral ontology allows the disparate syntactic representation and semantic interpretation of domain knowledge in different lateral ontologies. The different lateral ontologies can be translated because of the semantic mapping of each lateral ontology into a common EGV. The Ontology harmonization service translates the context between the different lateral ontologies, e.g., the translation between user query and SQL database query. The application builds the SQL statement according to the input XML user query, an example is shown in figure 3.

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

Figure 3: Example of query in XML

The query statement gives an SQL-like query structure that consists of two subsets for the query arguments and the constraints statement. The former is represented as a XML tag column and the later one as a constraint. This sort of XML representation hard-codes the semantic logic of user query into a specific structure. Each user query asks for the value of one or more properties or columns defining the constraints. The mapping SQL statement is of the form:

```
SELECT DISTINCT param.code_param,stations.localisation
    FROM parametres, stations, mesures
        WHERE ((param.CODE_PARAM=mesures.CODE_PARAM)
            AND (mesures.CODE_STATION=stations.CODE_STATION)
            AND (param.CODE_PARAM=1311 ))
```

We can relate the SQL statement to the XML query structure as follows:

1. The column set of XML is semantically equivalent to the Select in SQL.
2. The constraints set of XML is semantically equivalent to part of information in WHERE section of SQL.
3. Two more things are required to fill up the SQL statement, the table names that columns belong to and the relations to join those tables.

In points 1 and 2, the EGV terms and values in the XML query can be translated to the LDV terms and values directly using the ontology service. Browsing the LDV model, the SQL building service can find the related table name for the particular columns. Then the only question is how to join these tables together and form the WHERE section in SQL. A graph algorithm helps to calculate a join between tables. We can imagine that each table is an individual node in a graph, and each foreign key is the arc to link different nodes together, then the calculation of the join become the determination of a path between given nodes.

## 5. Discussion

International data sharing programs will necessarily be based on agreements of delivery. Due to cost restrictions and differences in monitoring traditions, such agreements will tend to focus on "lowest common denominator" approaches. The EDEN-IW approach opens up the opportunity to exploit the union of knowledge and data rather than the minimum agreed set. The EDEN-IW system is able to do this by using a semantic model that is open and scalable in several directions:

- New determinants and concepts can be added to the EGV ontology and is then available to all connected agents - the simple addition of a new determinant or database does not require any changes in user interfaces or other parts of the agent system;
- New application areas can rely on the already developed ontology elements and core agents and thus focus their efforts on developing their domain specific concepts and applications.
- New specialized user interfaces and processing agents can be developed by anyone, as long as their interfaces conform to the external EDEN-IW system interfaces including the EGV model.

In addition, the design of the EDEN-IW ontology model has the means to isolate the development of the software of the information processing system from the domain specific knowledge.

The EGV model and associated application commitment models are represented using the W3C RDF, DAML and OWL based models. A set of intelligent software agents has been created within the EDEN-IW project to allow users to make rich set of data and metadata queries and to support semantic data exchange and reasoning between heterogeneous database resources, metadata repositories and user applications. These agents are implemented using the FIPA agent forum standards implemented in a Java agent tool-kit called JADE and a Java based framework called Jena that allows agents to parse RDF, DAML and OWL messages. Queries are expressed in the EGV ontology. These are mapped into the LDV models of the corresponding database resources and then to SQL queries. The same agent services will also translate the SQL responses from the database back into to the common language (EGV) terms. The system has demonstrated the integration of two heterogeneous IW databases with a third in development.

## Acknowledgements

The research described in this paper is partly supported by the EC project EDEN-IW (IST-2000-28385). The opinions expressed in this paper are those of the authors. The authors acknowledge the input from all members of the project consortium and its corresponding user community.

## References

EDEN-IW (2004):Project Home Page: http://www.eden-iw.org. Available August 2004.

EU Cordis (2004): Project Database. http://dbs.cordis.lu/search/en/simple/EN_PROJ_simple.html. Available August 2004

Fensel D., Brodie M. L. (2003): Ontologies. ISBN: 3540003029. Springer-verlag 2003.

Poslad S., Willmott S. (2003): Modeling Agent Services for Open Environments, parts 1 & 2. 5th EASSS, European Agent Systems Summer School Tutorial, Barcelona, Spain, Feb. 2003

Zuo L., Poslad S. (2003): Supporting multi-lateral semantic information viewpoints when accessing heterogeneous distributed environmental information. EU Multi-Agent System (EUMAS) Conf., Oxford, UK, December 2003.