Semantic Integration of File-based Data for Grid Services

Andrew Woolf[1], Ray Cramer[3], Marta Gutierrez[2], Kerstin Kleese
van Dam[1], Siva Kondapalli[3], Susan Latham[2], Bryan Lawrence[2],
Roy Lowry[3], Kevin O'Neill[1]
[1] CCLRC e-Science Centre
[2] British Atmospheric Data Centre
[3] British Oceanographic Data Centre
{A.Woolf,K.D.O'Neill}@rl.ac.uk, K.Kleese@dl.ac.uk, {rnc,vpk,rkl}@bodc.ac.uk

Abstract

Data services for the Grid have focussed so far primarily on virtualising access to distributed databases, and encapsulating file location. However, orchestration of services requires richer information semantics than these mechanisms provide. Service inputs and outputs must be semantically matched, or characterised in order that sensible transformations may be performed. In many domains important information structures must be aggregated across multiple files, and numerous legacy file formats obscure the natural logical structure of information types. We present a solution for constructing semantic data services for an earth-sciences data Grid (the UK NERC DataGrid). A semantically-rich data model is developed, drawing on components from external ontologies. A 'storage descriptor' provides the mechanism for mapping legacy file-based storage onto data model instances. Finally, data services may be built on top of the data model to expose a semantic view of the data irrespective of the underlying file storage details. Our approach is similar to wrapper/mediator architectures for integrating database management systems.

1. Introduction

A fundamental goal of Grid technology is to enable scalable interoperability across heterogeneous and distributed resources [1]. The general approach is to encapsulate essential semantic behaviour within a service-oriented architecture [2].

The majority of work on data services has proceeded on two broad fronts (corresponding to the two common data storage mechanisms – database management systems (DBMS) and file systems): first the virtualisation of relational and XML databases [3]; and, second, virtualisation of file location [4]. As well, file transfer on the Grid is evolving the GridFTP and related specifications. Recently, the GGF DFDL-WG [5] has begun to develop a language for virtualisation of file format through an XML facade.

Missing from this program of work is research aimed at providing a semantically rich data representation layer for service-level interoperability in applications. In particular, current Grid data services retain the expression of legacy storage artefacts in service interfaces. The following is a short list of problems that occur across a variety of domains:

1. A number of heterogeneous data sources and file formats are in widespread use. Thus, the same *data* may be stored in a flat text file, a self-describing file (e.g. netCDF, or HDF), a spreadsheet, or relational tables, but completely different APIs are required for each.
2. Important information structures often need to be aggregated across files, or "subset" in semantically meaningful ways.
3. There is no uniform mechanism for serialisation of information in service workflows – different service interfaces are required for the same information represented in different legacy data formats.

Solutions to these are needed to facilitate sophisticated semantic service-level behaviour on the Grid.

DFDL and BinX [6] go some way to addressing the first problem, but different code may still be required to extract identical data from files in different formats. For applications with very large-volume data requirements, the DFDL program may be superfluous in any case. It is difficult to see, for example, how efficient optimisations could be made for reading gigabyte climate simulation files through a virtualised XML query interface. (On the other hand, DFDL will be a very useful tool for data preservation and long-term curation.) In addition, generic file content description languages, by themselves, will never enable data to be aggregated into domain-specific semantic information structures. Data integration must be combined with semantic modelling beyond what can be achieved with DFDL-like approaches alone. For example, DFDL aims to provide a mechanism for encapsulating the semantics of multidimensional rectangular arrays. However, more complex structures are required in the earth sciences. The international standard GRIB format used by meteorological agencies, for instance, supports data on non-rectangular 'reduced grids' (fewer gridpoints per parallel of latitude towards the Earth's poles). Furthermore, GRIB packs data values into a compact bit-level representation[1]. It will be difficult for DFDL to be able to represent the spectrum of domain-specific semantic structures in files across a variety of disciplines. Different file formats, while abundant, are at least well-known and generally have efficient APIs.

The method proposed here reverses the approach of supplementing files with content descriptions, instead placing primacy on semantic information models themselves; file-based data are regarded merely as the building blocks. It is explicitly recognised that in practice file formats are well-known community schemas with efficient APIs.

The remainder of this paper is structured as follows: section 2 motivates the current work in the earth science domain; section 3 reviews data modelling and integration from the database literature; section 4 presents an initial solution being developed in the NERC DataGrid (NDG) project; and conclusions are presented in section 5.

2. Earth Science Use Case

In many scientific domains, including the earth sciences, data stores are commonly file-based rather than held in database management systems. In addition, it is a characteristic of many scientific Grid applications that data volumes are giga- or tera-scale. It is necessary to provide an abstraction layer to these information stores. A mechanism is needed similar to the integration approaches applied in the databases domain [7]. The large data volumes rule out data warehousing approaches. Wrapper/mediator approaches and exchange standards are more suitable but have not been adequately developed for large file stores in Grid applications. Indeed semantic integration is a current topic of research even for small-scale file stores [8]. While DFDL is developing a mechanism for encapsulating limited syntax of files, the integration of higher-level domain-specific semantic structures remains a problem.

The need for semantics in an integration framework may be seen by considering three examples from the earth sciences. It is very common for data having a temporal component to be split across multiple files (e.g. tide-gauge measurements over separate years in separate files, or daily weather forecasts in separate files). However, these should be exposed as single logical data sources with a continuous temporal dimension. Similarly, while data over the entire globe may be stored in a file, an application may require only a semantic subset (e.g. 'temperature' over 'Europe', or within some 'latitude-longitude' bounding box). The third motivator for this work is the need, in a data Grid, to provide an abstraction layer across heterogeneous data sources. The NERC DataGrid project is building an infrastructure to provide uniform access to a wide range of environmental data in UK managed repositories, with an initial focus on the curated stores of the British Atmospheric and Oceanographic Data Centres (BADC [9] and BODC [10]). These data centres have very different legacy file-based data management infrastructures, and so a uniform semantic data model must be combined with integrating services.

Such problems have been identified in the past, e.g. [11]:

... Fortran codes usually use files to read input data and often use files as output. Most Grid platforms resolve this problem by providing some kind of file transfer system that enables users to copy input and output files to and from the particular resource where they are required.

There are a number of difficulties with this approach. First, transferring whole files across a network is wasteful since usually only a small part of the transferred data changes between requests. ... Results are hard to analyse in file format since data are rarely organised in a way that optimises searches. The distribution of relevant data across files makes aggregation and comparison of results inflexible. Applications that scientists write to analyse their results are not easy to reuse since they are tied too closely to particular file formats. Finally data are not represented in a consistent way with each application having a slightly different format for the same data structures

By way of specific example, consider a workflow utilising output from a climate simulation model as illustrated in Figure 1.

[1] A data value, Y, in GRIB takes the value: $Y=R+(X\times2^E)/10^D$, where R, E and D are reference parameters and X is a datum of arbitrary (but fixed) bitsize, http://www.wmo.ch/web/www/WDM/Guides/Guide-binary-2.html

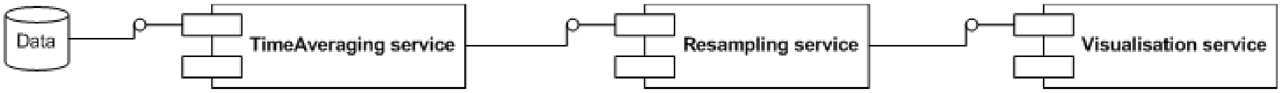

Figure 1: Processing workflow for climate model data

The model output consists of a timeseries of global snapshots of climate parameters (temperature, winds, etc) discretised on a curvilinear grid over the Earth's surface. A TimeAveraging service calculates the annual mean of these fields, while a Resampling service interpolates the fields onto a uniform longitude-latitude spatial grid for rendering with a Visualisation service. In practice, climate model output is produced normally as a series of flat files, one per timestep, in any of a variety of formats. Considerable metadata (often in the researcher's head) is required to identify the semantic contents of a file, even in the case of “self-describing” formats such as netCDF or HDF. A large amount of intervention is required at each step of the workflow to ensure that service inputs and outputs are matched and accompanied by necessary semantic metadata.

Simply put, Grid workflows require a representation of information and data semantics, not files. Clearly, request and query interfaces for data access services on the Grid must be meaningful with respect to the semantics of the data they encapsulate – a request for data from the North Atlantic is appropriate for geo-referenced environmental data, but not for molecular chemistry simulation data. While there is a rich history of semantic data modelling for database management systems [12,13], its use for legacy file encapsulation is considerably less mature.

Currently used approaches include the DODS/OPeNDAP [14] protocol for subsetting and delivering binary data via the web and the OpenGIS Consortium (OGC) web services for delivering maps and geo-data. None of these currently provide SOAP web service bindings. OPeNDAP is intended to have broad applicability, and so incorporates no specific mechanism for representing data semantics (such as geo-referencing information) [15]. Amongst other problems, the OGC services provide only limited support for the vertical dimension. To retain compatibility with community access mechanisms, however NERC DataGrid will support data delivery via both OPeNDAP and OGC services.

3. Database Integration and Data Modelling

There is a considerable history in database research of semantic data modelling and data integration techniques – both being dimensions of the problem outlined above for Grid data services required in the earth sciences.

Data modelling has evolved from Codd's relational model [16] through the ER model of Chen [17] to semantic [18,19] and fully object-oriented models incorporating inheritance, aggregation, and behaviour [20]. It is perhaps surprising that while such advances have been made in data modelling for DBMS, semantic modelling for access to file-based storage is far less developed. It is well-known in the climate sciences, for instance, that graduate students spend a considerable amount of time (re)developing code for low level data handling (file format translation and manipulation, archival etc.).

The databases literature also contains a considerable history of data integration methods. These have been developed for a rich range of problems including reverse- and re-engineering, schema translation, and database integration (see, for instance, [21,22,23,7,24,25,26] and references therein). Proven approaches for DBMS integration that might be examined for their applicability in a file-based data Grid include: data warehousing where data is imported *en-masse* from legacy databases and transformed into a common data model, and wrapper/mediator architectures where heterogeneous local sources are mapped to a global schema and integrated through middleware. Distributed/federated databases that require legacy systems to migrate to a common schema involve considerable effort and cost. For integration of heterogeneous file-stores in a data Grid, the warehousing and federation models are impractical. Instead, a wrapper/mediator approach is required, with a common data model exposed through semantic data services. Parent and Spaccapietra note [7]:

> *Two levels of complexity may be separated in addressing interoperability. The more complex case is when information involves data sources that are not in a database format ... In this case, understanding of ... data calls for sophisticated mechanisms for extraction of semantics.*

The requirement for data integration on the Grid has led to a significant amount of activity in the GGF DAIS-WG, with specifications developed for relational [27] and XML [28] Grid Database Services. Preliminary work extending OGSA-DAI for file access [29] has prompted a focus on encapsulating file formats through the DFDL (Data Format Description Language) Working Group of GGF. As mentioned earlier, encapsulating file format is often not the main concern in scientific data Grids. More important is the layering of rich domain-specific information semantics over file stores. In many cases, file formats are well-known community schemas for which there exist efficient and standard APIs, but which of themselves carry little in the way of information semantics. For example, the popular netCDF file format has standard API array subsetting methods (e.g. extract subarray `A(1:30,125:150)` from the full array `A(1:360,1:180)`). On the other hand, what is really needed in a data Grid is a semantic data integration framework that allows the request on this global geographic dataset for data from the Greenwich meridian to 30ºE and from 35ºN to 60ºN. OGC web services provide this type of interface, and, as a result, may be "cascaded" into value-added processing chains [30]. Pushed down to the file level, a request should use optimised native file format-specific I/O APIs.

4. NERC DataGrid Solution

NDG has adopted an approach that emphasises the semantics of data themselves, rather than the structure or abstract data models of files.

First, object models representing different data types at a conceptual level are constructed, quite independently of any file-based storage concerns. The ISO standard Geography Markup Language (GML, [31]) XML schema[2] is used for the representation of these models, which are known as 'feature types' – GML includes a large toolkit of conceptual models for spatial and temporal reference systems, geometry, topology, etc.

Second, feature instances are built by mapping file-based data onto the models, using a 'storage descriptor' wrapper mechanism.

Finally, services are implemented over the instances. Such services may be as straightforward conceptually as the OGC Web Feature Service[3] which is a web service for requesting feature instances serialised as GML. Or more complex services may be imagined, for example serialising feature instances into files of specified formats (of course this assumes conventions for such serialisation, e.g. the CF conventions [40]). The essential element is that the semantic data models (feature types) themselves are the fundamental unit of interest; file-based data are cast onto, and exposed as, feature instances. The process is inevitably 'lossy' – not everything in a file is necessarily represented. However, semantic unification is more important than completeness of representation. An initial semantic decoration of data in this manner adds context to an entire chain of processing, thus eliminating the required intervention at each step of a workflow alluded to earlier in the motivating use case.

We now consider these elements in turn.

The NDG project has developed the Climate Science Modelling Language (CSML) [32] as a first attempt to define a semantic data model with a limited number of classes ('feature types' [33]) across the atmospheric and oceanographic domain. This provides a semantic abstraction layer across the BADC and BODC file stores (Figure 2). The model is object-based, and similar to the object wrapping technique ('Semantic Transformation Protocol') applied by Papazoglou and van den Heuvel [22] to relational databases. It provides a common conceptual schema that avoids M*N mappings from M data sources to N data access mechanisms. NDG feature types are distinguished primarily on the basis of geometric and topologic structure, and draw on a number of emerging geographic information standards for conceptual models of spatial [34] and temporal [35] referencing, geometry [36], etc. These standards provide a powerful *implicit ontology* [37] for geographic data, independent of the underlying file contents. (Typically, files contain uninterpreted numerical data, little metadata, and rarely references to knowledge bases, controlled vocabularies, or ontologies.) The feature types are listed in Table 1 and illustrated in Figure 5. Controlled vocabularies may be referenced, e.g. for parameter-type semantics and units of measure, providing ontological commitment in the sense of Gruber [37].

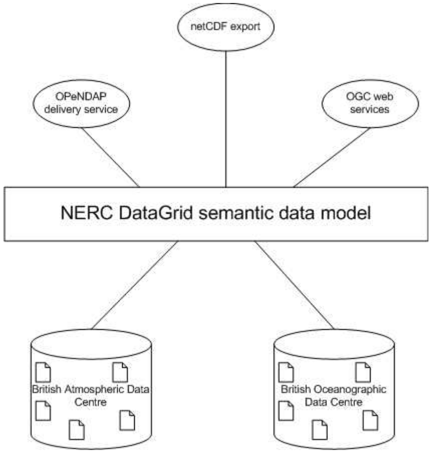
netCDF export
OPeNDAP delivery service
OGC web services
NERC DataGrid semantic data model
British Atmospheric Data Centre
British Oceanographic Data Centre

Figure 2: NERC DataGrid data model as abstraction layer

As well as the data classes themselves, a wrapper mechanism is needed for encapsulating file-based data in feature instances. Such a mechanism was employed by Abiteboul *et. al.* [8] to map structured files onto a database. They proposed a 'structuring schema' composed of two elements: a database schema, and an annotated grammar. The grammar is a generator for file instances with actions mapping non-terminals to elements of the database schema. The NDG mechanism

[2] While GML v.1 was released as an RDF schema, it is now issued as an XML schema.

[3] https://portal.opengeospatial.org/files/?artifact_id=7176

does not attempt to provide a complete generative grammar for files, relying instead on the fact that in practice stored files exist in a finite number of well-known community formats. Rules for extracting subsets of a file's contents are easily parameterised [32]. For instance, the abstract data model of a netCDF file consists of named rectangular numerical arrays called ‘variables’, with associated named vectors (‘dimension variables’) for each array axis. A given numerical extract, therefore, may be referenced simply by the corresponding netCDF variable name. Similarly, the WMO standard GRIB format consists of a series of records containing packed arrays of data. Individual records may be referenced by an associated ‘parameter code’ (for the physical parameter being represented in the record – temperature, wind, humidity, etc.), or by the record’s byte position in the file. In the rare case of non-standard file formats, a format description language like DFDL may be used to provide an XML encapsulation of the contents, but this is likely to prove far less efficient for reading than native file APIs. NDG’s wrapper mechanism is called an Array Descriptor, and provides the means to parameterise an extract of a file’s numerical contents. The Array Descriptor class, in turn, may substitute for any numerical content in the CSML data model. There is by design, therefore, a loose relationship between the contents of any given file and the object instance to which the contents are being mapped. Thus data in a netCDF file or a GRIB file may be exposed as the same semantic feature type. The Array Descriptor also provides for inline XML numerical content, limited implied transformations, and employs the ‘composite’ pattern [38] to enable aggregation of numerical content across files. A UML conceptual model of the Array Descriptor is illustrated in Figure 4 and an example of its use is shown in Figure 3.

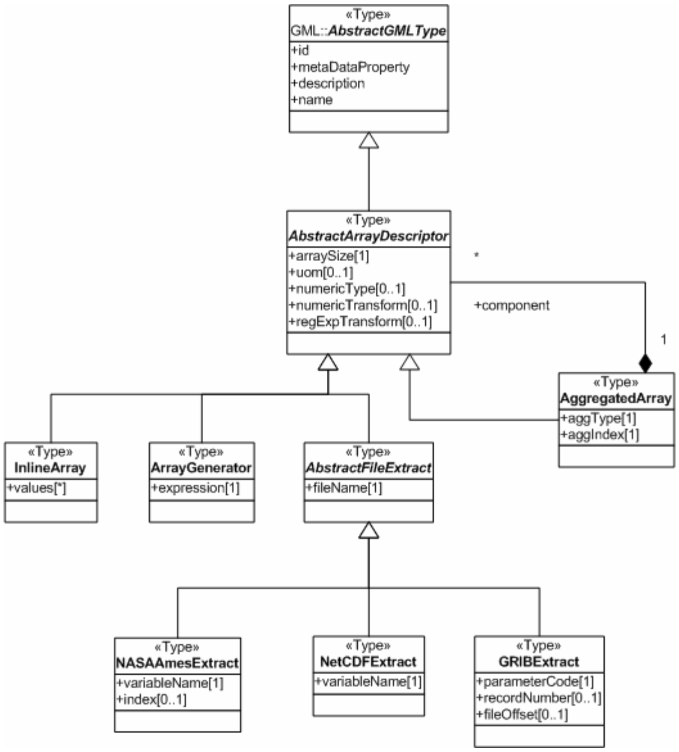

Figure 4: UML conceptual model for Array Descriptor wrapper.

```
<AggregatedArray gml:id="globaltemperature">
  <arraySize>180 360</arraySize>
  <aggType>existing</aggType>
  <aggIndex>1</aggIndex>
    <component>
      <NetCDFExtract>
        <arraySize>90 360</arraySize>
        <fileName>northern_hemisphere.nc</fileName>
        <variableName>TMP</variableName>
      </NetCDFExtract>
    </component>
    <component>
      <NetCDFExtract>
        <arraySize>90 360</arraySize>
        <fileName>southern_hemisphere.nc</fileName>
        <variableName>TMP</variableName>
      </NetCDFExtract>
    </component>
</NetCDFExtract>
```

Figure 3: Example encoding of aggregated netCDF extracts for Array Descriptor.

Given a semantic representation of data, and a wrapper mechanism for encapsulating file-based storage, a variety of data services may be implemented. For instance, an early prototype of our system [39] employed a convention for serialising feature instances into a netCDF file. Thus, a canonicalised file-instance with semantically meaningful contents could be produced regardless of the underlying legacy file store. By using such a canonical serialisation, a uniform representation of semantics is realised, and may be exploited for interoperable service workflows (Figure 1). Services would not need to be specialised for data in different file formats, nor would intervention be required at service inputs and outputs. Such semantic data services are needed for the full promise of Grid computing to be realised. As well, data services may provide semantically meaningful methods – for instance a service could subset a GridSeries feature instance (Table 1) by selection in time (“provide data for years 2002-2004”).

Table 1: Object classes ('feature types') in NDG semantic data model

CSML feature type	*Description*	*Examples*
TrajectoryFeature	Discrete path in time and space of a platform or instrument.	ship's cruise track, aircraft's flight path
PointFeature	Single point measurement.	raingauge measurement
ProfileFeature	Single 'profile' of some parameter along a directed line in space.	wind sounding, XBT, CTD, radiosonde
GridFeature	Single time-snapshot of a gridded field.	gridded analysis field, e.g. from a numerical simulation
PointSeriesFeature	Series of single datum measurements.	tidegauge, rainfall timeseries
ProfileSeriesFeature	Series of profile-type measurements.	vertical or scanning radar, shipborne ADCP, thermistor chain timeseries
GridSeriesFeature	Timeseries of gridded parameter fields.	numerical weather prediction model, ocean general circulation model

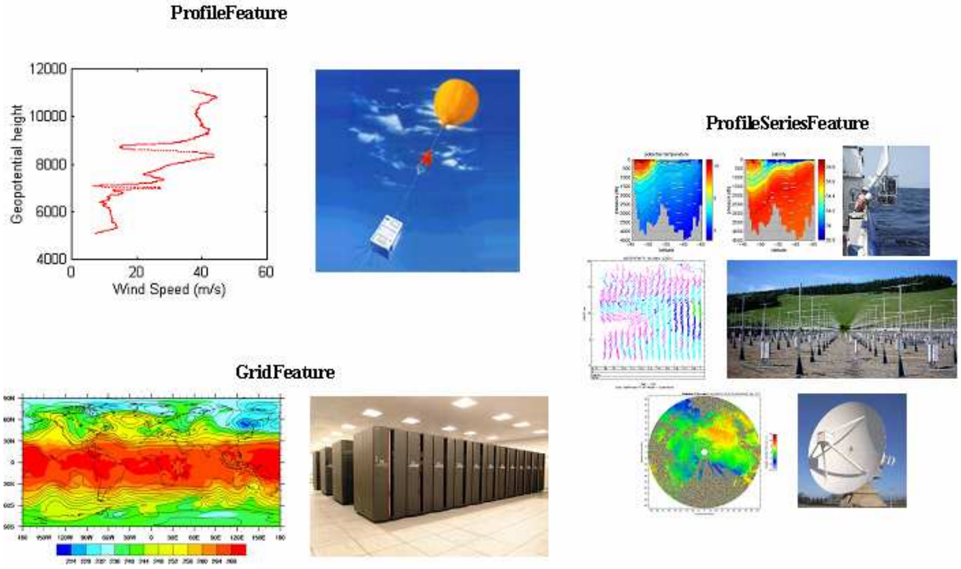

Figure 5: Visualisation of several object classes in NDG data model

5. Conclusions

The imperative for semantics on the Grid is even greater than the web. Workflows in a loosely-coupled service-oriented architecture must be accompanied by sophisticated semantic characterisations of data inputs and outputs. However, data services for the Grid have focussed so far primarily on encapsulating data syntax (distributed relational databases, file format and location).

We have presented a mechanism for constructing semantic data services being implemented in the NERC DataGrid project. A data model has been developed for representing data semantics. The range of data extends across the holdings of the British Atmospheric Data Centre and the British Oceanographic Data Centre, and includes both observational and numerical simulation data. A storage descriptor has been used for mapping legacy file-based storage onto data object instances. High-level services may be built on top of the data model, for semantic operations such as coordinate transformations, unit conversions, etc. A first demonstration service has been implemented enabling a data selection to be exported in a file conformant to the CF conventions [40] irrespective of the underlying storage details.

A generalisation of our approach can be postulated. The elements of a generic framework would include: (a) a meta-model for constructing semantically-rich domain specific data models independent of storage concerns, (b) a data storage description language for describing the construction of semantic data object instances from legacy storage artefacts, and (c) a canonical process for serialising semantic data instances in service workflows. Both implicit and explicit knowledge-bases or ontologies are supported by the general framework. Finally, this general framework has connections with both the GGF DFDL-WG (and BinX) and the DAIS and OGSA-DAI efforts. The relationship between these might be characterised as follows:

1. The framework discussed here could incorporate DFDL/BinX in the data storage description language
2. Data access services may be built on top of a data model constructed according to the framework. These could be exposed through Activity extensions in OGSA-DAI

Acknowledgements: This work was funded under the UK e-Science program through grant NER/T/S/2002/00091 from the Natural Environment Research Council.

[1] Foster, I., C. Kesselman and S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”, *Int. J. HPC Apps.* **15**(3) (2001) 200-222

[2] Foster, I., C. Kesselman, J. Nick, S. Tuecke, “The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration”, Global Grid Forum, OGSI-WG.

[3] Open Grid Services Architecture Data Access and Integration, http://www.ogsadai.org.uk

[4] See, for instance, the Storage Resource Broker (http://www.npaci.edu/DICE/SRB/), and Replica Location Service (http://www.globus.org/rls/).

[5] Data Format Description Language, http://forge.gridforum.org/projects/dfdl-wg/.

[6] Baxter, R. *et. al.*, “BinX – A tool for retrieving, searching, and transforming structured binary files”, UK e-Science All Hands Meeting, Nottingham

[7] Parent, C. and S. Spaccapietra, “Database Integration: The Key to Data Interoperability”, In *Advances in Object-Oriented Data Modeling*, ed. M.P. Papazoglou *et. al.*, The MIT Press (2000).

[8] Abiteboul, S. *et. al.*, “Querying the File”, In *Fully Integrated Data Environments*, ed. M.P. Atkinson and R. Welland. Springer-Verlag (2000).

[9] http://badc.nerc.ac.uk

[10] http://www.bodc.ac.uk

[11] Butchart, B., C. Chapman and W. Emmerich, “OGSA First Impressions – A Case Study Re-engineering a Scientific Application with the Open Grid Services Architecture”. UK e-Science All Hands Meeting, Nottingham

[12] Borkin, S.A., *Data Models: A Semantic Approach for Database Systems*. The MIT Press (1980).

[13] Gray, P.M.D., K.G. Kulkarni and N.W. Paton, *Object-Oriented Databases: A Semantic Data Model Approach.* Prentice Hall (1992).

[14] http://www.opendap.org

[15] Woolf, A., K. Haines and C. Liu, “A Web Service Model for Climate Data Access on the Grid”. *Int. J. HPC Apps.* **17**(3) (2003) 281-295

[16] Codd, E.F., “A Relational Model of Data for Large Shared Data Banks”. *Comm. ACM* **13**(6) (1970) 377-387.

[17] Chen, P.P., “The Entity-Relationship Model – Towards a Unified View of Data”, *ACM Trans. Database Systems* **1**(1) (1976) 9-36.

[18] Hull, R., and R. King, “Semantic data modeling: survey, applications and research issues”. *ACM Computing Surveys* **19**(3) (1987) 201-260.

[19] Peckham, J., and F. Maryanski, “Semantic Data Models”. *ACM Computing Surveys* **20**(3) (1988) 153-189.

[20] Norrie, M., “Advances in Object-Oriented Data Modeling”. In *Advances in Object-Oriented Data Modeling*, ed. M.P. Papazoglou et. al. The MIT Press (2000).

[21] Missaoui, R. *et. al.*, “Mapping an Extended Entity-Relationship into a Schema of Complex Objects”. In *Advances in Object-Oriented Data Modeling*, ed. M.P. Papazoglou et. al. The MIT Press (2000).

[22] Papazoglou, M.P. and W.J. van den Heuvel, “Leveraging Relational Data Assets”. In *Advances in Object-Oriented Data Modeling*, ed. M.P. Papazoglou et. al. The MIT Press (2000).

[23] Duwairi, R.M., N.J. Fiddian and W.A. Gray, “Schema Integration Meta-Knowledge Classification and Reuse”. In *Advances in Databases*, Proceedings of the 14th British National Conference on Databases, Springer LNCS 1094 (1996).

[24] Papakonstantinou, Y., H. Garcia-Molina and J. Widom, “Object Exchange Across Heterogeneous Information Sources”. In *Proc. IEEE Int. Conf. Data Eng.* (1995).

[25] Risch, T. and V. Josifovski, “Distributed data integration by object-oriented mediator servers”. *Concurrency Computat.: Pract. Exper.* **13** (2001) 933-953.

[26] Roth, M.T. and P. Schwarz, “A Wrapper Architecture for Legacy Data Sources”. IBM Technical Report RJ10077 (1997).

[27] Antonioletti, M. *et. al.*, “Grid Data Service Specification: The Relational Realisation”. DAIS-WG Informational Draft.

[28] Krause, A. *et. al.*, “Grid Data Service Specification: The XML Realisation”. DAIS-WG Informational Draft.

[29] Westhead, M, “Data Access and Integration Services – File Access”. DAIS-WG Informational Draft (2003).

[30] ISO/DIS 19119, “Geographic information – Services”.

[31] ISO/CD 19136, “Geographic information – Geography Markup Language”.

[32] Woolf *et. al.*, “Climate Science Modelling Language: Standards-based markup for metocean data", 85th meeting of American Meteorological Society, San Diego, Jan 2005.

[33] ISO 19101, “Geographic information – Reference model”

[34] ISO 19111, “Geographic information – Spatial referencing by coordinates”

[35] ISO 19108, “Geographic information – Temporal schema”

[36] ISO 19107, “Geographic information – Spatial schema”

[37] Gruber, T.R., “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”. Technical Report KSL 93-04 (1993). Knowledge Systems Laboratory, Stanford University.

[38] Gamma *et. al.*, “Design Patterns. Elements of Reusable Object-Oriented Software”, Addison-Wesley (1995).

[39] Woolf, A. *et. al.*, “Data Virtualisation in the NERC DataGrid”, *Proceedings of UK e-Science All Hands Meeting 2003*, Nottingham, UK, ISBN 1-904425-11-9.

[40] http://www.cgd.ucar.edu/cms/eaton/cf-metadata