GEDDM: Grid Based Conversion of Unstructured Data using a Common Semantic Model

Karen Loughran
Paul Donachy
Terrence J harmer
Ron H Perrott
Mark Prentice
Belfast e-Science Centre
www.qub.ac.uk/escience

Sarah Bearder
Jens Rasch
Datactics Ltd
www.datactics.co.uk

Abstract

Managing unstructured data is a problem that has been around for as long as people have been using computers to electronically store and retrieve information. As commercial and social demands for data collection increases so also does the number of formats and structures in which it is stored.

Additionally, the sheer volume of data presents challenges for access and conversion in a timely manner. To further compound this problem it is expected that the size of datasets will increase exponentially in the near future with ever increasing demands for information. There is therefore a need to access and convert large quantities of data from a variety of formats in a common, parallel and structured manner.

GEDDM is a collaborative industrial e-Science project in conjunction with BeSC and industrial partners Datactics Ltd. GEDDM has defined a Common Semantic Model (CSM) to assist with the representation and conversion of data from various sources. This model will facilitate the conversion of data residing in a range of formats including email, PDF, web log and various database formats into a common format for subsequent data mining operations. The project exposes such CSM based conversion capabilities via a suite of Grid Services called Data Conversion Services (DCS).

1 Introduction

As information increasingly plays a crucial role in today's economy, the quality of such data is of paramount importance. Demand is constantly increasing for availability of quality data from an ever increasing range of sources. Data is collected about unlimited subject matters and stored in vastly different formats. The extents to which these sources are structured vary immensely. At one end of the scale we have structured data such as databases where columns and fields make individual units of data clearly identifiable. At the other end of the scale we have unstructured data typically in the form of plain text documents containing descriptive text, where little is known about the semantic content or how to identify an independent self-contained piece of information. The level of structure will dictate how and to what extent a source can be queried to extract useful information. At the unstructured end of the scale queries may be simple string searches. At the structured end, where small indivisible units of data are identifiable, more complex queries will isolate or combine multiple entities to perform more intelligent data mining searches.

This paper presents the background and architecture of the CSM, outlining the commercial motivation for the work and identifying shortcomings of existing models. The paper presents issues and experiences of developing DCS using a grid based services architecture along with initial results from conversion of

unstructured real world sample data sources. Finally, it will provide a roadmap for implementing the model under an OGSA-DAI Grid based framework with a view to supporting access to and integration of a wider range of data sources.

2 Current Practice

The biggest challenge for representation and conversion of unstructured data is from data sources which do not fall within the category of traditional databases. For example, email, web log, PDF and Word report documents. These formats traditionally store little or no information about the structure or semantic content of data within them. They are essentially text-like Flat File Formats (FFF) with no means of identifying independent self-contained pieces of information. Additional to these formats, custom database formats often do not have ODBC type connectors, making direct access by external data mining applications impossible. In these circumstances database contents are often dumped to plain text files (FFF) using arbitrary print formats.

FFFs present many challenges for data conversion. These arise from the sheer irregularity of how data in FFF files can appear. In current practices, typically a new program must be designed and implemented for conversion of *each* new FFF data source to a structured format which can be subsequently mined. This is a costly and time consuming operation which must be performed for each business request to data mine a FFF. Of late, the Industrial partner has had requests from major customers in the US to interrogate data sources in numerous structures and formats. These business opportunities all present a similar technical problem, in that interfacing to such information in a common structured approach across such disparate structure and sources was a bottle neck and problematic. (E.g. one legal customer held over 45Tb of data in various formats including email, PDF, web logs, various RDBMS).

3 Architecture

In the overall architecture of the Common Semantic Model a client will first retrieve a remote sample of a FFF data source via a **getSample** service. Based on this sample the client will make a first attempt at describing the semantics of the FFF. The client will then undertake an iterative process of performing remote sample conversion on the FFF (via a **convertSample** service) with each newly revised version of the data source description. The process is repeated until the client is happy that the output from conversion accurately represents a formatted version of the FFF data source. The **convertSample** might typically convert only the first 50 records of a data source. Eventually the client may perform a full conversion of the whole FFF data source with a **convertFull** service. Results of each sample or full conversion will be displayed locally on the client in response to a **getResult** service call. Figure 1 demonstrates an architecture whereby a thin client performs such services.

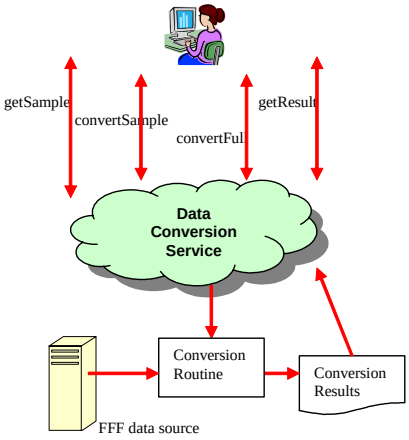

Figure 1

4 Data Conversion Services

Core to the design of the Common Semantic Model is the necessity to concisely and accurately describe the format and structure of FFF data sources. The Data Conversion Service will rely heavily on this description to convert these data sources into a coherent common text format.

In GEDDM a description of a FFF data source is contained within a Semantic Configuration File (SCF). The SCF describes what is known about the format and structure of the data source. The Semantic Configuration Language (SCL) is expressed in XML to make it easy to build supporting tools and libraries. A Schema Definition File (SDF) defines a set of rules by which a description of a FFF data source in an SCF must comply.

There are many similarities in principle between the design of the SCL within this Common Semantic Model and the design of Data Format Description Language (DFDL) [4] by an OGSA-DAI work group [1]. In evaluating suitability of SCL and DFDL for use within the CSM it is important to highlight the main difference between them. An FFF description in SCL will be considerably more restrictive and less extensible than one in DFDL. SCL focuses primarily on the description of the 4 broad categories of FFF formats in the industrial client's business domain (email, web log, report and database dump). SCL is sufficient for our needs and relatively straightforward to implement. DFDL processing has not yet been implemented under OGSA-DAI and it awaits GGF approval before progress is made in this area. But one compelling argument for the simpler approach with SCL is that for our purposes we treat the files as character and string data as we are simply representing and converting a *text* format to a common *text* format for *subsequent* mining. We do not need to represent and manipulate pieces of data as dates, times, integers, floats etc. We are only concerned with the preparation of data resources before data mining by Datactics core data mining engines.

Figure 2 below shows an example of a FFF data source generated from a text file dump of a database. Figure 3 shows how it is described in SCL and figure 4 gives the resulting Common Output Format File (COFF) after conversion which clearly records format and structure within the data. As an example of just one type of irregularity which may existing within an FFF, the text of one column (the address field) is continued on the following line *after* the ending new line of the record. This is recorded in the description of the data source with a "multiline" attribute within the "pfield" element.

App	Account	Address	Balance
IMP	343818	Dede H Smith 181 Glen Rd Earls Court, London	8600.76
IMP	565777	Annie Saunders 60 Newhaven St Edinburgh, Scotland	9905.50

Figure 2 – FFF data source

As there is no clear delimitation between fields in this data source it must be described with positional information within the SCF using "pfield" elements:

```
<datasource>
 <database>
 <header><headertext>App  Account  Address
 Balance</headertext></header>
 <rectype eorecord='\n'>
  <pfield name="App" pos=1 length=3/>
  <pfield name="Account" pos=10 length=6/>
  <pfield name="Address" pos=24 length=23
                                  multiline="yes"/>
  <pfield name="Balance" pos=49 length=8/>
 </rectype>
 </database>
</datasource>
```

Figure 3 – SCF description of FFF

```
<STARTSOURCE>
<TEXT> App  Account  Address
 Balance </TEXT>
<STARTRECORD>
IMP@343818@Dede H Smith, 181 Glen Rd, Earls
Court, London@8600.76
<ENDRECORD>
<STARTRECORD>
IMP@565777@Annie Saunders,  60 Newhaven St,
Edinburgh, Scotland@ 9905.50
<ENDRECORD>
<ENDSOURCE>
```

Figure 4 – COFF output from conversion

Each separate indivisible unit of data as described in the SCF is clearly delimited from other data in a common manner.

For the actual conversion of a data source (described by an SCF) an object oriented hierarchy of components is defined which together collectively represents a complete data source. From these, a direct mapping can be made between XML components within the SCF and OO class components which internally represent data units of the type described by the XML element. The implementation performs a SAX parse of the XML description held in the SCF, and creates and inserts objects of appropriate types into the correct location within the new overall OO hierarchical representation of a data source.

Each object in this hierarchy encapsulates information relating to the data unit it describes in its attributes. It will also encapsulate an all important "parse" method which will contain code for parsing a component of its type from a FFF based on its attributes. Once the SAX parse phase has generated an internal representation of the source in the form of objects, a call is made to the "parse" method of the highest parent object, i.e. the "data source" object, which will result in a "parse" operation on each sub-component in turn. As each object is parsed, the resulting output is produced to COFF with clear delimiters inserted.

5 FFF Data Access and Integration

The conversion service capabilities described in the previous section are exposed via a suite of Grid Services called Data Conversion Services (DCS). This is based on the Open Grid Services Architecture (OGSA) model [2].

The DCS is implemented under an OGSA-DAI [1] Grid based framework which provides an extension to the OGSA specifications [2,3] to allow data resources such as databases to be incorporated within an OGSA framework. This supports access to and integration of a wide range of data sources. OGSA-DAI is extended to represent, convert and access data held in FFF such as those identified in this paper. This is achieved by providing additional functionality to four key extensibility points within DAI's architecture. The provision of:

1. A Connectivity Driver class which invokes conversion operations on FFF data sources.
2. Activity definitions and class mappings to allow the client to interact with a FFF data source.
3. Design of perform documents to express data conversion operations.
4. A configuration file to specify the data source exposed and point to details of the connectivity driver, activity & class mappings and perform document designs active for the FFF data source.

6 Summary

Grid technology presents a framework that aims to provide access to heterogeneous resources in a secure, reliable and scalable manner across various administrative boundaries. Conversion of unstructured data sources is an ideal candidate to exploit the benefits of such a framework.

Once a global standard has been set for describing the format and structure of FFF data sources it is expected that OGSA-DAI[1] will be extended to provide access to such FFFs. This paper showed how a restricted variant of such a standard can be defined and implemented for the business requirements of a data mining company and how conversion services can be implemented to generate a Common Output Format File of any FFF based on this description. It provided an insight into how OGSA-DAI can be extended to allow for access to FFF data sources to be incorporated within an OGSA framework.

The design of the SCL language to describe a FFF in this application is suitably restrictive and simplistic for our purposes; we were only concerned with the conversion of text *prior* to subsequent data mining. Other data mining applications may benefit from the full power and complexity of DFDL which provides the ability to manipulate data from FFFs as dates, times, integers etc. It would therefore provide an application with the ability to perform proper data mining operations on FFF.

References

[1] OGSA-DAI
http://www.ogsadai.org

[2] OGSI
http://www.gridforum.org/ogsi-wg/

[3] OGSA
http://www.globus.org/ogsa/

[4] DFDL
http://forge.gridforum.org/projects/dfdl-wg/

[5] Grid Enabled Distributed Data Mining and Conversion of Unstructured Data
http://www.qub.ac.uk/escience/projects/geddm/