Semantics and the Earth Science Markup Language

Rahul Ramachandran, Sunil Movva, Helen Conover and Sara Graves
Information Technology and Systems Center
University of Alabama in Huntsville
Huntsville, AL 35899
rramachandran@itsc.uah.edu

***Abstract*- Earth science data is archived and distributed in many different formats. These formats vary from character format, packed binary, and "standard" scientific formats to self-describing formats. This heterogeneity is partly due to historical and partly practical reasons. Since there is not a single standard format that meets the needs of the entire Earth Science community, most scientists choose the most familiar data format for their use. Interoperability problems arise when these scientists try to use unfamiliar data with their applications. The Earth Science Markup Language (ESML) is designed as an elegant solution to this problem. ESML is an interchange technology that enables data structural interoperability without enforcing a standard format within the Earth science community. Scientists can write external metadata files using the ESML Schema to describe the structure of their data files. Applications can then utilize the ESML Library to parse this ESML Description file and decode the data format. Software developers can now build data format independent scientific applications utilizing the ESML Library. The ESML design team is now focusing on semantic interoperability for data processing automation via ESML. Semantic tags defined in various domain ontologies can be added to the ESML Description files. This complete machine understandable description will allow development of intelligent applications that can understand and "use" the data. This paper will focus on semantics in ESML and its use in a prototype "intelligent" application for subsetting.**

I. INTRODUCTION

Earth Science data are archived in many different digital forms. These forms vary from character based text format, and packed binary to different "standard" scientific data formats including various self-describing data formats. Some examples of commonly used scientific data formats are GRid In Binary (GRIB), Binary Universal Format Representation (BUFR), Common Data Format (CDF), network CDF (netCDF), Hierarchical Data Format (HDF) and HDF for NASA's Earth Observing System (HDF-EOS). There are historical and practical reasons for having all these different standard data formats. Many of these formats were developed by different agencies for their own use and needs. Some were developed by different communities within Earth Science for specific needs such as compactness, or portability. However, as the Earth Science community has become more interconnected and the science problems have gotten more complex, the cross collaborations between different agencies and scientists have fostered data sharing. Users of Earth Science data now have to understand the intricacies of these different data formats in order to use them. In addition, the analysis applications used by the scientists must be modified every time a new data format is encountered, or the data must be translated into some other familiar format.

The Earth Science Markup Language (ESML) [1,2] provides a solution to this data/application interoperability problem, allowing multiple existing data formats to interoperate with different applications. The next logical development in ESML is to allow the semantic meaning of the data itself to foster the development of "smart" services, tools and applications for science data. This would allow applications to not only read heterogeneous data formats but also to "use" the science data intelligently for processing and analysis.

II. SEMANTICS

Semantics define the meaning of a word or term. Tim Berners-Lee's vision for a semantic web is an extension of the current world-wide web, where people and computers work in cooperation and machines are better able to process and "understand" the data [3]. One of the traditional methods to provide semantic information in Machine Learning, Artificial Intelligence and Intelligent Systems is by using an ontology, which is a formal, explicit, specification of a shared conceptualization [4]. In an ontology, the type of concept and constraints on its use are explicitly defined. It is formal and shared because it has to be machine parsable and must capture consensual knowledge.

An ontology can be used in a wide variety of ways, from simplest uses, as a catalog for controlled vocabulary, to complex, where it is used by agents to determine general logical constraints. The DARPA Agent Markup Language (DAML) is one of the more widely used ontology languages, especially for use in the US military in areas of command and control, as well as in military intelligence such as integration of information from many sources. DAML has also been used to describe web services to allow automated web service composition and interoperability [5, 6] and for describing grid services for Grid Infrastructure [7]. These domain ontologies provide semantic mediation between database schema, applications' input/output, and workflow work items with rules for constraining the parameters of machines/algorithms and inferring allowable configurations. In addition, they can be used in checking for semantic validity of a workflow composition. Thus, the use of such ontolgies allows users to chain services automatically and to query distributed databases.

Combining domain ontologies with ESML to describe both the structure and semantic meaning of Earth Science data offers a new possibility of developing "smart" services, tools and applications for ***scientific data processing*** and analysis. Before the details of a "smart" ESML based subsetter prototype are described, an overview of the ESML design is required to highlight the scalable features that make these semantic additions easy.

III. ESML

The primary design goal for ESML is to provide an elegant solution allowing multiple existing data formats to interoperate with different applications. The three primary components in ESML that enable data/application interoperability are the ESML Schema, the ESML Description files and the ESML software Library (See Fig. 1). The ESML Schema defines the grammar for generating an ESML Description file (instance document) for a given dataset in a certain data format. The ESML Library is the middleware that applications use to parse an ESML description and retrieve the data in the associated data file(s).

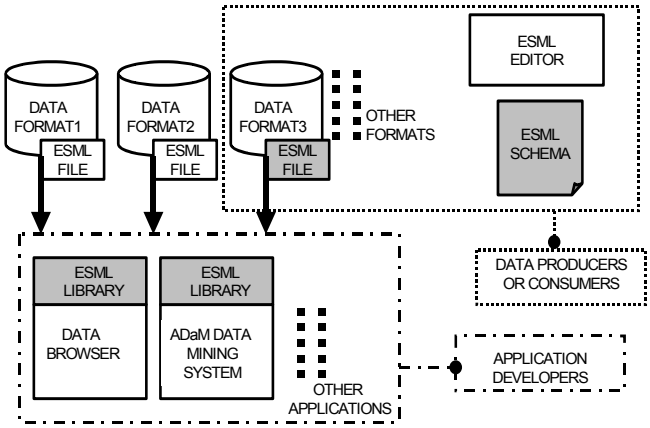

Figure 1: ESML Components

1. ESML Schema

Fig. 2 depicts the scalable ESML Schema design. The schema design embraces orthogonality and generality principles. The schema design contains a few basic building blocks that can describe different data structures. These basic building blocks are separately understandable and free from interactions when combined. The design is also scalable to allow the addition of other data format descriptions without perturbing existing data format modules. The ESML root element contains a `SyntacticMetaData` element, which in turn can contain descriptions of different data formats. The current schema supports descriptions for unstructured data formats such as ASCII and Binary, as well as self-describing data formats such as HDF-EOS and GRIB. A separate ESML element is defined for each individual data format. Descriptions for additional data formats can be added to the schema.

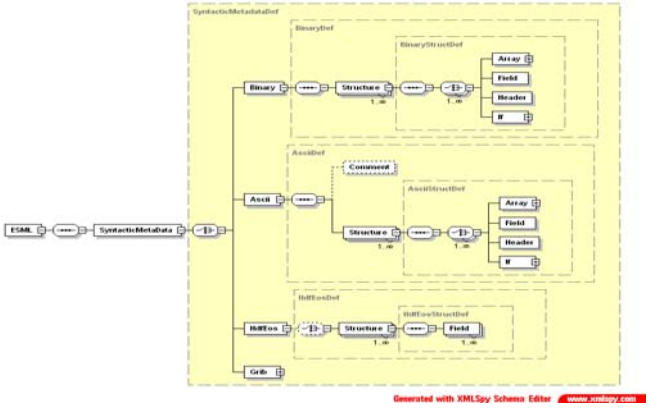

Figure 2: ESML Schema

2. ESML Library

The ESML Library has been recently redesigned with a layered architecture to allow more scalable functionality. The conceptual architecture of the ESML Library is depicted as a block diagram in Fig. 3. The core library (depicted in solid lines) consists of a User API, a Document Object Model (DOM) tree, plug-in modules for different data formats and an API for adding new modules.

This architecture design allows both horizontal and vertical functionality and extensibility. First, different functions such as Subsetting, Remote File Access, etc., can be added on top of the core library by extending the User API. The basic functionality of the core library can be increased horizontally by adding new plug-in modules for data formats not yet included in ESML. The use of separate modules for different formats also allows ease in packaging the library. Users interested in certain data formats can select only the desired modules.

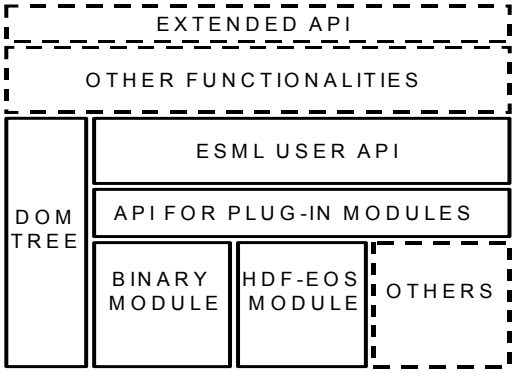

Figure 3: ESML Library Architecture

The core library provides the basic functionality of reading the structural metadata from the ESML Description file and retrieving the data from the data file. A canonical Document Object Model (DOM) tree structure stores the metadata as it is parsed from the ESML Description file. For self-describing formats such as HDF-EOS, the data file is queried

for additional metadata and the retrieved values are used to populate the DOM tree. The DOM tree is accessible at different functional levels of the library. The process of accessing the data objects from the file via the User API is designed to mimic directory traversal.

3. ESML Description File

An example ESML Description file for a simple data file in Binary format with three fields is shown in Fig. 4. The data fields are two dimensional arrays. After declaring the format, the subsequent sequence of declarations follow the data structure of the file. The entire data file is enclosed within a single `Structure` element. The fields themselves are nested in two `Array` elements with size specified for each dimension. Data values in the array are described by the `Field` element, with a name (Uwind, DimX, DimY, respectively) and a data `type` definition to read integer data in base ten.

Source code for the ESML Schema and the Library are available from the ESML web site (`http://esml.itsc.uah.edu`).

```
<SyntacticMetaData>
 <Binary>
  <Structure instances="1" name="SampleSet">
   <Array occurs="100">
    <Array occurs="100">
     <Field name="UWind" type="Int32" order="LittleEndian"/>
    </Array>
   </Array>
   <Array occurs="100">
    <Array occurs="100">
     <Field name="DimX" type="Int32" order="LittleEndian"/>
    </Array>
   </Array>
   <Array occurs="100">
    <Array occurs="100">
     <Field name="DimY" type="Int32" order="LittleEndian"/>
    </Array>
   </Array>
  </Structure>
 </Binary>
</SyntacticMetaData>
</a:ESML>
```

Figure 4: Example ESML Description File

IV. Semantics and ESML: Future Work

A prototype is being developed to demonstrate the use of ESML with ontologies in developing 'smart' services for use with various data sets. An important objective of this prototype is to design a mechanism to embed semantic tags for data fields in the ESML Description files, along with links to the ontologies where these terms are defined. However, the design should have minimal impact to both the ESML Schema and the ESML Library. For this initial prototype, simple ontologies will be designed to describe the scientific domain of Earth Science, as well as the concepts of a dataset and a service, such as subsetting. The dataset ontology will encapsulate the concept that a dataset must contain fields, and these fields can be of different types, such as navigation or data fields. The subsetting ontology will conceptualize rules for dataset subsettability. For example, the rule to be described in the ontology for this prototype could state that a file is subsettable if it contains navigation fields such as latitude, longitude and time. The dataset and subsetting ontologies can be used by an inference/reasoning engine to infer the various possible subsetting options for a particular data file.

This proposed design is shown in the new modified ESML Description file in Fig. 5. This is the same ESML Description file as in Fig. 4 except it now contains additional semantic information, namely a link to the Dataset/Subsetting Ontology (*data-set*). In addition to the structural metadata, it contains a `SemanticMetadata` element, enclosing semantic terms and their mapping to the structure and data fields described in the `SyntacticMetadata` element. Thus, the ESML Description file now provides complete structural metadata to read the data values from the file and semantic metadata to allow inference and intelligent use.

```
<a:ESML xmlns:a="ESML" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="ESML.xsd" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"  xmlns:ds="http://www.itsc.uah.edu/data-set#" >
 <SemanticMetaData>
          <ds:Latitude rdf:ID="DimX"/>
          <ds:Longitude rdf:ID="DimY"/>
          <ds:DataField rdf:ID="UWind"/>
          <ds:DataSet rdf:ID="SampleSet">
               <ds:hasField rdf:resource="#DimX"/>
               <ds:hasField rdf:resource="#DimY"/>
               <ds:hasField rdf:resource="#UWind"/>
          </ds:DataSet>
 </SemanticMetaData>
 <SyntactiMetaData>
  <Binary>
       <Structure instances="1" name="SampleSet">
            . . . . . . . . . . .
       </Structure>
  </Binary>
 </SyntactiMetaData>
</a:ESML>
```

Figure 5: ESML Description File with Semantics

V. Discussion

This prototype will clearly demonstrate the power of combining semantics with ESML via ontologies for developing "smart" services and applications for science data. This approach will allow domain experts such as scientists to develop their own ontologies and provide semantic definitions in the ESML Description file. It also allows software developers to design and build services/applications that can exploit this complete metadata description. Thus, ESML will provide a versatile solution for both structural and semantic interoperability between services/applications and data.

Acknowledgements

This research work has been partially funded by NASA's Earth Science Technology Office. The authors would like to acknowledge Bruce Beaumont, Andrew McDowell, Matt Smith and Xiang Li for their contributions to this work.

REFERENCES

[1] R. Ramachandran, M. Alshayeb, B. Beaumont, H. Conover, S. J. Graves, N. Hanish, X. Li, S. Movva, A. McDowell, and M. Smith, "Earth Science Markup Language," presented at 17th Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, 81st American Meteorological Society (AMS) Annual Meeting, Albuquerque, NM, 2001.

[2] R. Ramachandran, H. Conover, S. Graves, and S. Christopher, "EARTH SCIENCE MARKUP LANGUAGE: A Solution to the Earth Science Data Format Heterogeneity Problem," presented at American Meteorological Society's (AMS) 19th International Conference on Interactive Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, Long Beach, CA, 2003.

[3] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, vol. 284, pp. 34-43, 2001.

[4] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," Knowledge Acquisition,, vol. 5, pp. 199-220, 1993.

[5] J. Hendler, "Agents and Semantic Web," IEEE Intelligent Systems, vol. 16, pp. 30-37, 2001.

[6] S. A. McIlraith, T. C. Son, and H. Zeng, "Semantic Web Services," IEEE Intelligent Systems, vol. 16, pp. 46-53, 2001.

[7] C. Goble and D. D. Roure, "The Grid: An Application of the Semantic Web," SIGMOD Record, vol. 31, pp. 65-70, 2002.