Using ESML in a Semantic Web Approach for Improved Earth Science Data Usability

Rahul Ramachandran, Helen Conover, Sunil Movva 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. This data format heterogeneity leads to interoperability problems when 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 interoperability without enforcing a standard format within the Earth science community. Scientists can write external files using the ESML Schema to describe the structure of the data file. 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. Furthermore, semantic elements defined in different domain ontologies can also be added to the ESML Description Files. This machine understandable description will allow development of intelligent applications that can now understand and "use" the data. This paper describes the current work on embedding semantics in ESML and its use in a prototype smart service for subsetting.

Introduction

As remote sensing technologies continue to improve and costs for online storage and network bandwidth decrease, the volumes of Earth science data easily available and accessible to researchers is increasing dramatically. However, because of the variety of formats in which these data are stored, data usability remains a major problem. In particular, 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. This means that users of Earth Science data have had to understand the intricacies of many different types of data, from different sources and in a variety of formats, in order to use them. In addition, the analysis applications used by the scientists have had to be modified every time a new data format is encountered, or the data must be translated into some other familiar format.

To address this problem, the Information Technology and Systems Center (ITSC) at the University of Alabama in Huntsville has developed an interchange technology allowing different applications to interoperate with multiple data formats. The Earth Science Markup Language (ESML) (Ramachandran et al., 2001, Ramachandran et al., 2002), which is based on the eXtensible Markup Language (XML) (Bray, T., J. Paoli, et al., Byron, P. and A. Malhotra 2001), provides a standard method for describing the structure of a data set in any of several common scientific data formats. ITSC's next phase of research in this area is to explore new ways to capture and describe semantic information about Earth science data, within the context of the Semantic Web (Berners-Lee, et al. 2001). Combining domain ontologies with ESML to describe both the structure and semantics of Earth Science data offers new possibilities for developing "smart" services, tools and applications for scientific data processing and analysis. This paper describes the ongoing effort of embedding semantics via domain ontologies with ESML and prototyping a "smart' application.. The goal of this endeavor is to demonstrate the power of coupling ontologies with ESML descriptions to build smart services.

What is ESML?

ESML is unique in that it is not another new data format, but rather a metadata based solution for decoding science data formats. The primary design goal for ESML was to provide an elegant solution allowing multiple existing data formats to interoperate with different applications. Equally important is ease of use for both the data users and the application developers. Therefore, the programming language design principles of orthogonality and generality were embraced. The three primary components in ESML that enable data and application interoperability are the ESML Schema, the ESML Description Files and the ESML Library, depicted in Figure 1.

The ESML Schema defines the grammar for generating an ESML Description File (instance document) for a given dataset in a certain data format. A GUI based Editor (ESML Editor) is also available for users to generate ESML Description Files. The ESML Library is the middleware that applications use to parse an ESML

description and retrieve the data in the associated data file(s). ESML and its associated software library allow wider interoperability of Earth Science services and tools, enabling Earth Scientists to work more easily with data in a variety of formats. ESML is being used in different applications such as collocating different satellite datasets and ingesting disparate data in numerical weather models. Source code for the ESML Schema and Library are available at the ESML web site (http://esml.itsc.uah.edu).

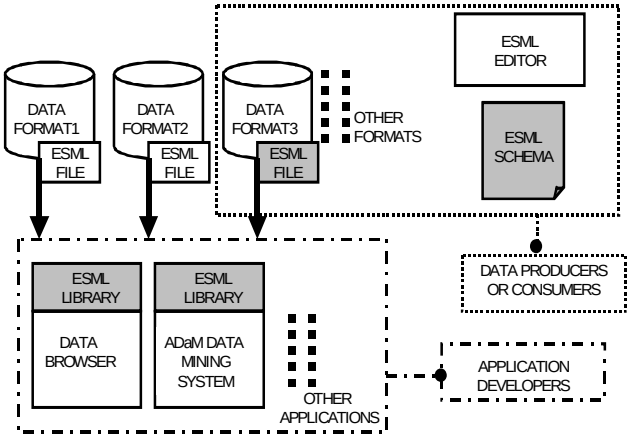

Figure 1: ESML Components

ESML Schema

The ESML Schema design comprises 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 ESML root element contains a `SyntacticMetaData` element, which in turn can contain descriptions of different data formats. A separate ESML element is defined for each individual data format. The design is scalable to allow the addition of other data format descriptions without perturbing existing modules.

ESML broadly classifies data formats into two different groups. Data files in Binary and ASCII text encoding, lacking structural metadata, form the first group. These data files do not contain sufficient metadata information to allow development of general reader software that can extract data fields from these files. Also, the same data can be structured in several different ways depending upon the data producer's preference. ESML provides for a complete structural description of these types of data files. Self describing data formats such as HDF, net-CDF, HDF-EOS constitute the second group. Files in these formats do contain structural metadata. The software libraries that accompany these formats can read the embedded metadata in the files and use that information to retrieve the data for the users. Because another key design requirement of ESML was to minimize redundancy of information, ESML's structural description for self describing formats is minimal in order to leverage the existing software library to retrieve data for the end user.

ESML Library

The ESML Library is the middleware that applications use to parse an ESML description and retrieve the data in the associated data file(s). 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 Figure 2. The core library (shown with 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 particular data formats can select only the desired modules.

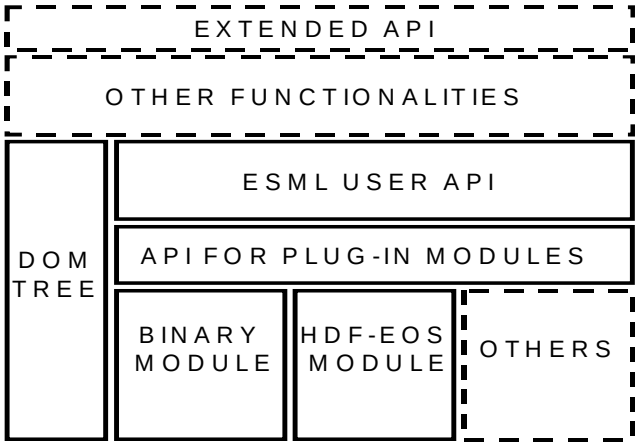

Figure 2: ESML Library architecture

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. The canonical 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.

ESML Description Files

An example ESML Description File for a simple binary data file with three fields is shown in Figure 3. 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 data 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.

```
<<a:ESML xmlns:a="ESML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="ESML.xsd" >
 <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 3: Example ESML Description File

ESML in the Semantic Web

The next logical development for ESML is to use Semantic Web technologies to foster the development of "smart" services, tools and applications for science data, which rely on the semantics of the data itself. These applications will not only read data in heterogeneous, formats but also "use" the science data intelligently for processing and analysis. The ESML design team is currently exploring mechanisms to allow embedding semantic elements for data fields in ESML Description Files, along with links to the ontologies where these terms are defined.

Semantic parsers built on top of the core ESML Library will be able to utilize the ontologies to infer information that can then be used by the applications or services. The conceptual architecture for these smart services can be seen in Figure 4.

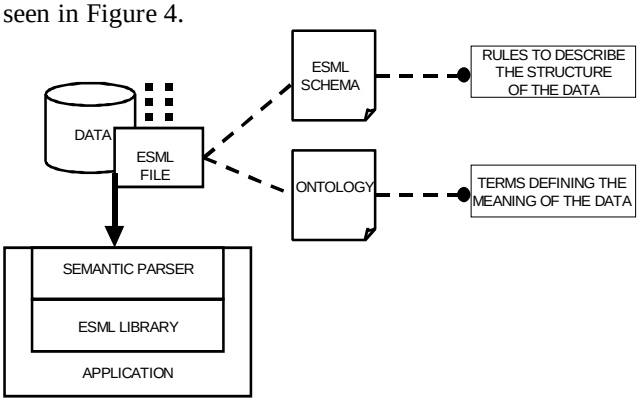

Figure 4: Architecture for building "smart" services or applications with ESML

Intelligent Data Services Prototype

ITSC is currently pursuing research to demonstrate the use of ESML with ontologies in developing "smart" services for use with various data sets. This prototype will use embedded semantic tags for data fields and links to ontologies in the ESML Description Files, as described earlier. These enhancements will be designed to have minimal impact to both the current ESML schema and the ESML Library. For this initial prototype, simple ontologies, such as those shown in Figure 5, will be designed to describe the concepts of a dataset and a data preprocessing 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.

There are reasons for selecting subsetting as prototype smart service. Because the volume of data flowing from orbiting sensors has increased dramatically, the dataset sizes have grown from megabytes to terabytes. Subsetting provides the capability to reduce the size and the complexity of the data set and is used frequently by scientists as a part of their preprocessing step. The subsetting ontology will contain 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 enhancement to the ESML Schema is shown in the new modified ESML Description File in Figure 6. This is the same ESML Description File as in Figure 3, 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.

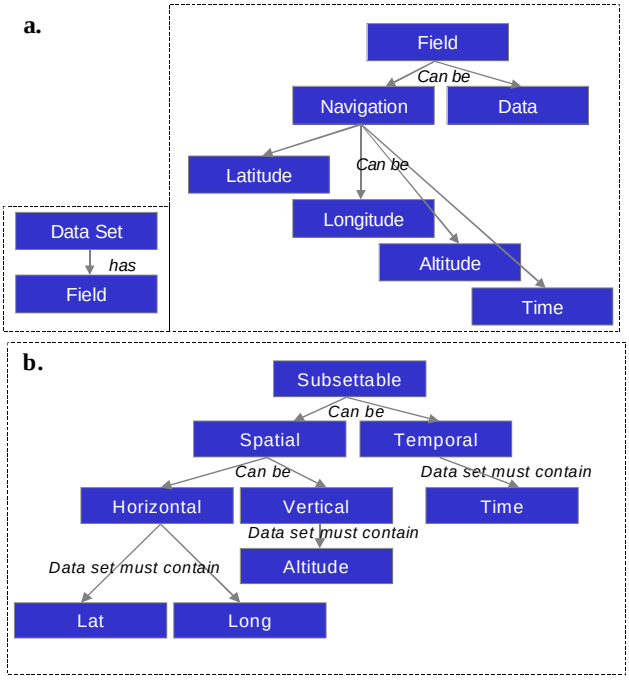

Figure 5: (a) Dataset ontology; (b) Subsetting ontology

```
<a:ESML xmlns:a="ESML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="ESML"
xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
xmlns="http://www.itsc.uah.edu/esml-ex#">
  <SemanticMetaData>
    <Latitude rdf:ID="DimX"/>
    <Longitude rdf:ID="DimY"/>
    <DataField rdf:ID="UWind"/>
    <DataSet rdf:ID="SampleSet">
      <hasField rdf:resource="#DimX"/>
      <hasField rdf:resource="#DimY"/>
      <hasField rdf:resource="#UWind"/>
    </DataSet>
  </SemanticMetaData>
  <SyntacticMetaData>
    <Binary>
      <Structure instances="1" name="SampleSet">
          ……
      </Structure>
    </Binary>
  </SyntacticMetaData>
</a:ESML>
```

Figure 6: ESML Description File with semantics

The example described here is an extremely simplified case to demonstrate the coupling of semantic information using an ontology with the structural description in an ESML Description File. The actual subsetting ontology being designed is much more complicated, containing concepts such as projections, units, scaling, direction, etc and each concept having a complex relationships with the other.

The new ontology-driven services such as the subsetting prototype will be able to guide users through data preprocessing based on their output requirements. The major benefit of this research is to seamlessly integrate data into applications or numerical models by automating data preprocessing steps.

Conclusion

ESML is already being used to access and interpret Earth Science data in its varied formats. By combining the ideas and applications of ESML with those of the Semantic Web, the data can be easily interpreted both in format and in meaning. The prototype described in this paper will enhance the basic functionality of ESML by linking it with domain ontologies. The smart service developed for this prototype, will provide for automated preprocessing, thus relieving the user of much of the tedium involved in data preparation. With these capabilities, the science researchers will not only be able to find and access many forms of existing data, but will also able to exploit it more easily and effectively.

Acknowledgements

This work was funded by NASA Earth Science Technology Office under grants 5-22104 and 5-22461. The authors would also like thank Karen Moe for her support and guidance in this work.

References

Berners-Lee, T., Hendler, J. and Lassila, O., 2001. The Semantic Web. Scientific American, 284(May): 34-43.

Bray, T., Paoli, J. and Sperberg-McQueen, C.M., XML 1.0 Specification, W3C.

Byron, P. and Malhotra, A., 2001. XML Schema Part 2: Datatypes, World Wide Web Consortium.

Ramachandran, R. et al., 2001. Earth Science Markup Language, 17th Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, 81st American Meteorological Society (AMS) Annual Meeting, Albuquerque, NM.

Ramachandran, R. et al., 2002. Interchange Technology for Applications to facilitate generic access to heterogenous data formats, 2002 IEEE International Geoscience and Remote Sensing Symposium and the 24th Canadian Symposium on Remote Sensing, Toronto, Canada.