Report on XEWA-00: The XML Enabled Wide-Area Searches for Bioinformatics Workshop

Terence Critchlow
critchlow@llnl.gov
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory (LLNL)

Abstract

The XEWA-00 workshop, held in December 2000 and sponsored by the IEEE Computer Society, was organized to bring together members of the bioinformatics community to determine if XML could simplify accessing large, heterogeneous, distributed collections of web-based data sources. The starting point for a series of breakout and group discussions was a proposed strawman of a grammar that described how to query a data source through its web interface. As a result of these discussions, the approach was validated, the strawman was refined, and several reference implementations are being generated as part of an ongoing effort. This article contains an overview of the workshop, including the proposed approach and a description of the strawman.

1. Introduction

The XEWA workshop provided a forum for the bioinformatics community to begin addressing a problem of increasing importance — how to access the large number of disparate genomics data sources distributed over the web. To improve the likelihood of producing generally applicable results, we encouraged participation from several communities. Due to significant interest in this problem, we were able to attract participants representing academia, industry, and the public sector from the biology, bioinformatics, and computer science perspectives. This mix provided the basis for the lively and productive discussions that took place.

The workshop began with two presentations aimed at establishing a common terminology for the participants to build on. A strawman proposal was presented and later refined during the following series of breakout and group discussions. Finally, a plan was developed for continuing the work started here. The remainder of this article provides an overview of these presentations and discussions. For additional information, the interested reader is referred to the XEWA web pages at www-casc.llnl.gov/xewa.

2. Establishing a Common Terminology

Tom Slezak (LLNL) started off the workshop with a presentation about "What XEWA means to a Genomics System Builder." This talk focused on the day-to-day challenges faced by the people who regularly provide bioinformatics solutions to genomics researchers, and helped to ensure that the discussions remained strongly grounded in reality. The reality of bioinformatics, at least from a CS perspective, is that it is ugly. Historically, biologists have refused to release control of their data, or to expend significant resources to manage it. As a result, there is a proliferation of custom interfaces that provide limited access to collections of data without any standard representations. In this environment, providing access to more than a handful of sources is impossible. Furthermore, even accessing data contained within a single source can be challenging if it requires performing queries unanticipated by the interface designer. For example, answering what seems like a simple request — identify the genes from Chromosome 19 that have been entered into GenBank — requires significant effort because the only available interface capable of answering this query is a simple keyword search in which the data is viewed as a collection of flat files (note that dates are also part of the record, so almost every file contains the string "19"). In addition, there is no standard representation of "Chromosome 19" — 5 different versions were identified while attempting to answer this query, including one which used roman numerals. While XEWA is not attempting to solve the data integration problem

```
Page_interface    ::=    page_url script_url thesaurus_url query
Query             ::=    input_param+ output constraint*
Input_param       ::=    script_param thesaurus_params [confidence] [transformation]
Ouput             ::=    No_data | Error | (base_location [delay?] output_mapping*) *
Output_param      ::=    result_params thesaurus_param [confidence] [transformation]
Parameter         ::=    name [datatype] [param_loc] [required?] [value]
Param_loc         ::=    Parser | ParamName | AnchorString |
                             Start_line [end_line] [start_column [end_column]]
```

Figure 1: Low-level View of Interface Interactions

(including the formatting issues highlighted by the example), if we help system builders to access a larger number of sources with the same resources, they will be much better off than they are now.

Carole Goble (Univ. Manchester) gave the second presentation, "A Knowledge Representation Briefing," which described categories of data representations, from user documentation to well-defined ontologies. She also highlighted the need for semantics to be presented in machine-understandable formats since, until knowledge can be represented in a format that applications can use and be exchanged between applications, the vast amounts of information stored in computers will not be fully utilized. Thus, the goal of moving from human-understandable information to machine-understandable information is central to the ontology community. While this is an extremely hard problem, the definition of semantically powerful formats, such as the Ontology Inference Layer (OIL) [1], which build on less rigorous formats, is the first step in allowing meaning to be transferred between applications. The lowest level of the OIL hierarchy is XML, which, while a standard for data interchange, lacks the ability to represent semantic information. The Resource Description Framework (RDF) and RDF Schema (RDFS) formats [2] are built on XML, providing the basic mechanisms for attribute type definitions, class structure definitions, and inheritance, which are important in defining and understanding the semantics of the underlying data. Finally, the OIL, which is being developed by the W3C Semantic Web Activity Group, builds on RDF to provide the additional semantic information required to describe rich knowledge representations such as frames. In addition to simply representing this information, OIL provides mechanisms for translating it between alternative representations, thereby allowing it to migrate across applications.

3. An Initial Framework

To provide focus to the discussions, a strawman definition of two grammars was presented. These grammars, summarized in Figure 1 and Figure 2, were initial attempts at describing how to interact with a web interface and represent two views of the interaction between the interface and the data. The grammar in Figure 1 describes a low-level view of the interface, including details such as the location of the page being described, the input variables, and the expected output. The grammar in Figure 2 provides a higher-level view that reflects the different types of interfaces typically encountered in a domain (e.g., keyword search, similarity search) complete with examples of the tags used to describe the parameters and values that typically return results when passed to the appropriate interface. This grammar can also be viewed as a thesaurus, since an informal, semantic specification of the interfaces and attributes

```
Service_class     ::=    name required_param * optional_param *
                                negative_example* return_param* constraint*
Parameter         ::=    name datatype [description] [parent_parameter]
                                example_tag* example_value*
```

Figure 2: High-level View of Interface Interactions

may be provided in the description attribute. Since a low-level view of an interface may reference the high-level view of the corresponding service class to provide semantic information about the input and output attributes, the interface cannot be defined before the service class. This precedence dependence may be avoided by omitting the references to the thesaurus if the semantic information is either not available or not wanted.

The attribute definitions contained in these grammars are similar to those found in many other languages (e.g., CORBA IDL, Java Beans). This new definition is not an attempt at reinventing the wheel, but rather at defining a representation capable of including definitions from these varied languages. It should also be noted that these grammars contain information, such as translations, mappings, and confidences, which is not present in the others. As a result of a desire to reuse existing work, the grammars contained some elements that were intentionally left undefined and will be filled in as appropriate formats from external communities are identified — avoiding reinvention of formats where standards already exist. For example, the formats of translations and constraints were left unspecified in the hope that we could leverage work from the relevant research communities. Several discussions clarified that implementations of the grammar should be flexible enough to include the alternative representations, allowing simple translations between representations to be defined. Breakout sessions addressed issues surrounding adding semantics to instances of the grammar and refining the grammar and representation of the strawman.

4. Group Discussions

The semantics breakout session began by trying to limit the scope of their discussions to a practical subset of the overall problem. In particular, they raised the questions: How do you describe the types of knowledge contained within the data instances provided by a source? What are the semantics of the queries supported by an interface? What is the meaning of executing an application as part of a query (i.e., running a visualization tool)? These questions play an important role in understanding an interface, and the answers impact an application's ability to chain queries across interfaces (pass the output of one to the input of another). This session also identified two general categories of queries: exploratory queries, where you want to find everything related to a given starting point; and focused queries, where you are looking for something specific. Often, answering focused queries requires using significant semantic information to identify and utilize the relationships between query results. Answering these queries requires different meta-data, and allowing both to be asked simultaneously is not easy. The conclusions from this session were that we needed to reduce the scope of the problem in order to keep it manageable. Thus, while we do not need to provide sufficient semantics to allow chaining of queries, we should concentrate on defining only the information required to perform exploratory queries. These decisions reduce the complexity of the problem, allowing us to manage the semantics without having to solve the entire problem up front.

The format breakout session began by discussing the appropriate interface view to focus on. A focus on the low-level view would provide a format capable of automatically generating data source wrappers. A focus on the high-level view would provide a format describing how to interact with the set of wrapper for a service class. The motivation behind separating these views, and initially focusing on only one, was to make the resulting grammar both more understandable and more general. For example, separating the example tags and values from the canonical interfaces simplifies the high-level grammar and makes it easier to understand; including an explicit protocol attribute in the low-level grammar removes the implicit assumption that all connections are through http, and generalizes it. This session also discussed which data format should be used to represent the meta-data. Alternatives considered include XML, RDF, OIL, and CORBA IDL. The conclusions of this session were that we should focus on the high-level view, and that the implementation formats are not important, since as long as they conform to the grammar transforming between them, they should be relatively simple. As reference implementations and service descriptions are created, we will encourage the authors to disseminate them to the community at large.

As the final exercise of the workshop, the group investigated several existing blast [3] interfaces and

```
<service_class>  <name> Blast Service </name>
      <attribute> <required /> <name> sequence </name>
                        <type> <union> <type> <class> GenericSequence</class> </type>
                                          <type><class> Annotationnumber</class> </type>
                                          <type><class> FileName</class></type>
                        </union> </type></attribute>
      <attribute> <required/> <name> database </name><type> String </type> </attribute>
      <attribute> <required/> <name> outputFormat </name>
              <type> <enum><item> email </item> <item> http </item></enum>
              </type></attribute>
      <attribute> <optional/> <name> filter </name>
              <type> <boolean /> </type> </attribute>…….</service>
```

Figure 3: Example Service Class

partially defined a service class representing those interfaces using an XML format and a slightly modified version of the strawman's high-level grammar (a summary of this definition is shown in Figure 3). While this definition was sufficient to describe how to interact with a wrapper of the target sites, it contained several compromises. In particular, the constraint format was not formalized and is currently assumed to be English. This prevents developing applications capable of using these constraints, for example, to restrict the valid values for one parameter depending on the value of a different parameter. While this was not seen as a major limitation, the hope is to identify an appropriate constraint representation language in the future.

5. Conclusion

The workshop wrapped up with a discussion on how to best proceed, and tasks were identified. First, the workshop web page is being updated to include copies of the presentations and of the breakout session slides (without added commentary), and, to facilitate communication, a mailing list is being created. Volunteers were found to develop reference models for the meta-data formats using OIL and XML. A repository will be created to allow for easy dissemination of both reference models and service class definitions to the community. Volunteers will also present the dual-level meta-data approach espoused by the group to some of the major data source providers, with the hope of having them describe their interfaces using this format. Finally, we anticipate holding a second workshop, co-located with ISMB 2002, where we will demonstrate how our approach simplifies accessing multiple distributed data sources. There is a lot of work to be accomplished before the bioinformatics community can reap the benefits of integrated access to hundreds of data sources, but this workshop provided a significant step in the right direction.

Acknowledgments

The author would like to thank all of the participants of the XEWA workshop, but especially Tom Slezak, Carole Goble, and Bob Coyne, without whom the workshop would not have been as productive.

References

[1] Fensel, D., et al., "OIL in a Nutshell," in: Knowledge Acquisition, Modeling, and Management, Proceedings of the *European Knowledge Acquisition Conference* (EKAW-2000), R. Dieng et al. (eds.), Lecture Notes in Artificial Intelligence, LNAI, Springer-Verlag, Oct. 2000.

[2] Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 19990222. February 1999. www.w3c.org Technical Report.

[3] Madden, T.L., Tatusov, R.L. & Zhang, J. (1996) "Applications of Network BLAST Server," *Meth. Enzymol.* 266:131-141.

This work was performed under the auspices of the U.S. Department of Energy by the University of California Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.1. UCRL-JC-141925