TOWARDS A DISTRIBUTED INFORMATION ARCHITECTURE FOR AVIONICS DATA

Chris Mattmann
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
M/S 171-264
Chris.Mattmann@jpl.nasa.gov

Dana Freeborn
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
M/S 171-264
Dana.Freeborn@jpl.nasa.gov

Dan Crichton
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
M/S 306-438
Dan.Crichton@jpl.nasa.gov

ABSTRACT

Avionics data at the National Aeronautics and Space Administration's (NASA) Jet Propulsion Laboratory (JPL) consists of distributed, unmanaged, and heterogeneous information that is hard for flight system design engineers to find and use on new NASA/JPL missions. The development of a systematic approach for capturing, accessing and sharing avionics data critical to the support of NASA/JPL missions and projects is required.

We propose a general information architecture for managing the existing distributed avionics data sources and a method for querying and retrieving avionics data using the Object Oriented Data Technology (OODT) [3] framework. OODT uses an XML [4] messaging infrastructure that profiles data products and their locations using the ISO-11179 [2] data model for describing data products. Queries against a common *data dictionary* (which implements the ISO model) are translated to domain dependent source data models, and distributed data products are returned asynchronously through the OODT middleware. Further work will include the ability to "plug and play" new manufacturer data sources, which are distributed at avionics component manufacturer locations throughout the United States.

KEYWORDS

Information Architecture, OODT, Distributed Data Management.

1. INTRODUCTION

In October 2003, the Center for Advanced Avionics (CAA) at the Jet Propulsion Laboratory funded the State of the Art Survey Task that seeks to bring component manufacturers such as Boeing, Ball Aerospace, and Honeywell to JPL in order to survey state of the art products being developed in industry for possible use on future JPL missions. Manufacturers provide product specifications in the form of presentation slides, data sheets, and documents which describe their state of the art products. These product specification documents are stored in Docushare [6], a web-based content retrieval system. Though Docushare provides a means of defining a standard set of metadata for describing a document, it is too limited in scope to allow true search and retrieval of the avionics components at the detailed, domain-specific level required by flight engineers for system design.

An existing approach at JPL for managing this survey data as well as other sources of avionics product information is in the form of a set of HTML web pages. HTML data tables are used to store component metadata and links to the component data product sheets; however, these tables provide no underlying method for search and retrieval other than the primitive *find on page* methods of the internet browsers.

Consequently, while the State of the Art Survey Task has been successful in gathering data specifications for new flight technology, this information is still not accessible through simple search and retrieval mechanisms by flight system design engineers. Furthermore, the ability to compare the survey data with other existing avionics data sources is not currently possible as these distributed data sources suffer from common data management problems such as domain dependent data models and heterogeneous data sources.

Our solution is two-fold. The first step is to design and implement a database for storing avionics component specifications that are currently not managed in a way that makes them accessible, searchable and comparable. The second step is to employ the Object Oriented Data Technology (OODT) framework [3] in order to create an information architecture for the distributed, heterogeneous data sources. We start by creating a ***Data Dictionary*** for Avionics flight component data, using the ISO-11179 model for describing data resources. The data dictionary provides a standard data model for an XML web service providing an online data management system to describe and manage the survey data, the State of the Art Database (SOADB). Next we *plug in* the SOADB, the HTML data sources, and the Avionics Docushare data sources to the OODT framework, which allows distributed clients to access the distributed component data in a common XML format, which is compliant with the given data dictionary. We create a sample Query client as a proof of concept of this system.

Our proposal for follow-on work in the next fiscal year includes *plugging in* manufacturer data sources at their physical source, distributed across the country, thus making their component data "visible" through the distributed query infrastructure that the OODT middleware layer provides. This essentially eliminates the need for manufacturers to distribute their data product specifications during survey presentations to the CAA, and saves the cost and management of the survey data at JPL, which is a long-term goal for the center.

2. CURRENT PROGRESS

2.1 SOADB Software Architecture

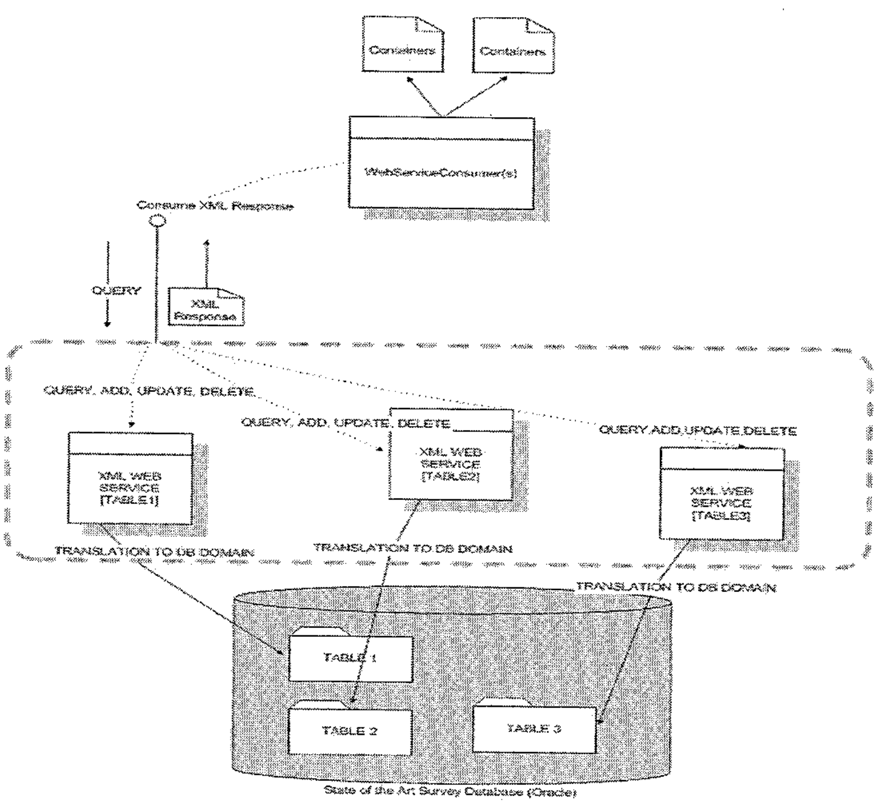

Figure 1. Implemented XML web service software architecture of State of the Art Survey Database (SOADB). *Consumers* consume *XMLResponse* objects, which are generated by the SOADB web api (the blue dashed box in the figure). XMLResponse objects are generated by *XML Web Services* , which service each relational database table within

the physical Oracle data store. Consumers generate *Containers*, which are "container" classes for Avionics data products that provide methods to access the actual data product information.

Our current implementation of the State of the Art Survey Database (SOADB) uses xml web services (**SOADB Web API**) that provide functionality through HTTP POST Requests. Requests allow the user to *ADD, UPDATE, and DELETE* from the actual Oracle database we have implemented that stores the flight component data. An XML response is generated and encoded into a SOAP Envelope [6] to be served back to the requestor, and allows the user to verify that the requested database operation was performed. If any error occurred, it is encoded in the *status* portion of the XML response.

2.2 Distributed Query Infrastructure

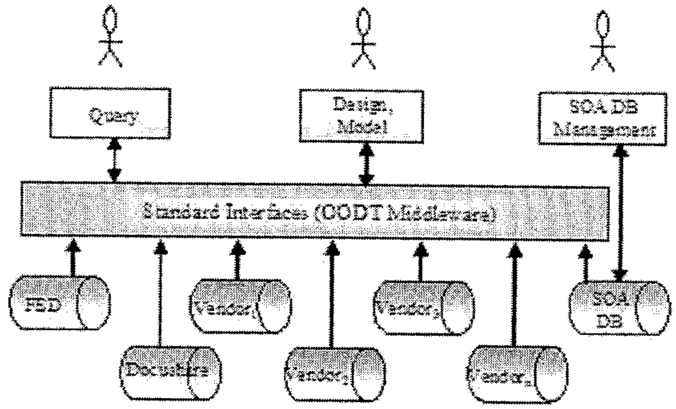

Figure 4. Proposed information architecture implementation.

Current work includes applying the OODT framework to integrate the SOADB (through its web service API), the HTML Data Source (labeled as **FED for the Flight Equipment Database** in Figure 4), and Docushare into a common data view defined by the data dictionary. OODT provides a DIS-style [3] querying mechanism that mediates source data elements from domain dependent terminology to more generalized data dictionary terminology (e.g., examining 2 domain dependent attributes from 2 different data sources, "computers" and "computer systems" and realizing that both attributes refer to the same common term, "computer"). To employ OODT, first a user creates a *Data Dictionary*, [1] to query against. The data dictionary serves as a reference by which to translate common data elements (**CDE**s) to domain dependent source data elements (**SDE**s) [1]. After the query is formulated and issued, it is forwarded asynchronously to *profile servers*, which profile the locations of *data resources* using the ISO-11179 [2] model, and the Dublin Core set of data elements [9]. These resources can be *product servers*, which access heterogeneous data sources and deliver products back to the requester, *data resource locations*, which is anything that can be referenced by a URI (uniform resource locator) [8], or *other profile servers* themselves which are managing another set of data resources. Profile servers enable the intelligent routing of queries so that queries are only set to product servers that relevant to the query. Once the profile server has found a set of *Product Servers*, the query is sent to the product servers to retrieve the product. Queries to the product server are translated into each local data domain (**CDE to SDE translation**) for resolution. Data products are then packaged and returned using a commn data structure, **XMLQuery**, that maps back to the underlying CDEs (**SDE to CDE translation**) of the data dictionary.

In Figure 4, we present our proposed information architecture for the Center of Advanced Avionics at JPL.

3. CONCLUSION AND FUTURE WORK

We have presented an overview of an effort to establish an information architecture for Avionics flight component data at the Jet Propulsion Laboratory. Our information architecture leverages the OODT framework to correlate distributed, heterogeneous avionics data sources, and induce a common data model, or data dictionary, which we use to create our information architecture. The common data model then serves as a basis for search and retrieval of avionics component data products and avionics manufacturer information. In the next fiscal year, further work includes *plugging in* manufacturer data sources at their physical source, distributed across the country, thus making their component data available for search and retrieval as well as other types of processing in the common data view that OODT provides. This eliminates the need for manufacturers to distribute their data products during survey presentations to the CAA, and saves the cost and management of the survey data at JPL, which is a long-term goal for the center.

ACKNOWLEDGEMENT

Chris Mattmann is a Co-Task Lead, and Software Architect on the State of the Art Survey for Avionics Task, in the Earth Science Data Systems Section at JPL. He is also a PhD Candidate in the Department of Computer Science at the University of Southern California, where he is currently researching software architectures under Dr. Neno Medvidovic. His interests are in software architectures, machine learning, and distributed computing. He holds a B.S. and M.S. in Computer Science from the University of Southern California. He can be reached at Chris.Mattmann@jpl.nasa.gov.

Daniel Crichton is a Project Element Manager at JPL and the principal investigator for the Object Oriented Data Technology task where he is leading a research effort developing distributed frameworks for integrating science data management and archiving systems. He also currently serves as the implementation manager and architect of a JPL initiative to build an enterprise data architecture. His interests are distributed architectures, enterprise and Internet technologies, and database systems. He holds a B.S. and M.S. in Computer Science. He can be reached at Daniel.Crichton@jpl.nasa.gov.

Dana Freeborn is the Technical Group Supervisor of the Data Management Systems and Technologies Group at JPL. The focus of her group is on providing data management solutions for science missions, enterprise applications and archive systems at a low cost with low risk. Dana is also a science data processing Cognizant Design Engineer for the earth orbiting Sea Winds scatterometer missions currently flying and Co-Task lead on the State of the Art Survey Task for the Center for Advanced Avionics at JPL. Her interests are in data modeling, database design, distributed architectures and the development and use of common software frameworks for data management systems. She holds a B.S in Mathematics and can be reached at Dana.Freeborn@jpl.nasa.gov.

The work described was performed at the Jet Propulsion Laboratory, California Institute of Technology under contract with the National Aeronautics and Space Administration.

REFERENCES

[1] D.J. Crichton et al., 2001, A Distributed Component Framework for Science Data Product Interoperability. *The 17th International Conference on Scientific and Technical Data.* http://oodt.jpl.nasa.gov/doc/papers/italy_codata/italy_paper.pdf

[2] ISO/IEC, "Framework for the Specification and Standardization of Data Elements 11179-1," *Specification and Standardization of Data Elements 11179*, International Organization for Standardization, Geneva, 1999.

[3] D.J. Crichton et al., 2000, Science Search and Retrieval using XML, *The Second National Conference on Scientific and Technical Data, U.S. National Committee for CODATA, National Research Council*, http://oodt.jpl.nasa.gov/doc/papers/codata/paper.pdf.

[4] W3C. Extensible Markup Language (XML), Version 1.0, http://www.w3.org/TR/REC-xml.

[5] Simple Object Access Protocol (SOAP) 1.1, http://www.w3.org/TR/SOAP/.

[6] Xerox, Inc. DocuShare: Knowledge Sharing Software, Rochester: Xerox, Inc., http://docushare.xerox.com/.

[7] Douglas C. Schmidt and Frank Buschmann., 2003, Patterns, Frameworks, and Middleware: Their Synergistic Relationships, *Proceedings of the IEEE/ACM International Conference on Software Engineering*, Portland, Oregon.

[8] Uniform Resource Identifiers (URI): Generic Syntax. http://www.ietf.org/rfc/rfc2396.txt.

[9] Dublin Core Metadata Element Set, Version 1.1: Reference Description, http://dublincore.org/documents/dces/