

Semantic Integration of Heterogeneous NASA Mission Data Sources

Richard M. Keller¹, Daniel C. Berrios², Shawn R. Wolfe¹, David R. Hall³, Ian B. Sturken³

¹National Aeronautics and Space Administration

²University of California, Santa Cruz

³QSS Group, Inc.

Intelligent Systems Division, NASA Ames Research Center

Mail Stop 269-2, Moffett Field, CA 94035-1000

{keller, berrios, shawn, dhall, sturken}@email.arc.nasa.gov

One of the most important and challenging knowledge management problems faced by NASA is the integration of heterogeneous information sources. NASA mission and project personnel access information stored in many different formats from a wide variety of diverse information sources, including databases, web servers, document repositories, ftp servers, special-purpose application servers, and Web service applications. For example, diagnosing a problem with the International Space Station (ISS) communications systems might require a flight controller to access multiple pieces of information, including the repair history of specific system components (e.g., from a relational database), a historical listing of system anomalies (from a text file accessed through a web server), and crew communication procedures (stored as Microsoft Word documents on a document server). Even more challenging is the ability to discern how the information stored on these different data sources in different formats is *semantically* related, and therefore how it can be coherently integrated. The presence of similar field names (e.g., database column names, HTML labels, Web form descriptors, etc.) is no guarantee of conceptual similarity. For example, a field called “temp” that is identically named across two different sources may reflect a *temperature* in one source and a *temporary* quantity in the other. Even if we are certain they are both temperatures, do they both represent degrees Celsius? Are they both measuring the same physical aspect? Of the same physical system?

This heterogeneous and semantically ambiguous environment makes it difficult to get a truly comprehensive, integrated view of data and information resources required for problem solving. Accessing and searching for information across sources is tedious and error-prone for users because one must separately probe each source, verify semantic congruence, and then manually integrate the results. In NASA’s high-risk operational environment, even small errors in performing this integration can impact human safety and vehicle health.

Although the potential benefits of access to integrated information are widely acknowledged, this process is considered to be very expensive. The standard integration

solution is to build custom software that integrates a fixed set of data sources. Generally, the software must be reworked whenever an existing data source is modified or a new information source needs to be added. The brittleness of this approach is what makes information integration systems costly to build and maintain.

To address these problems, we have designed and implemented a generalized data mediation architecture called *SemanticIntegrator*. In contrast with specialized integration solutions, this architecture is more easily reused for different domains and information sources, resulting in reduced software engineering costs compared to conventional hard-coded solutions. Our approach uses semantic integration techniques (Noy, Doan, and Halevy 2005) to combine information sources based on semantic models of the stored data and explicit integration rules. For each source, a *data source ontology* is developed that captures the semantics of the underlying data (Crubézy, Pincus, and Musen 2003). In addition, a software wrapper is written that exposes the underlying data source as a “virtual” semantic resource. Turning the data sources into semantic resources enables them to be queried using a common, semantic query language. The wrapper takes semantic queries as input and dispatches native data source queries (e.g., in SQL) to the actual data sources. In addition to the data source ontologies, an *integrating ontology* is developed to capture the customer’s view of relevant data and relationships across the various sources. To access integrated data, a client application queries the integrated ontology. Using a set of ontology translation rules, this query is mapped into a set of separate queries against the data source ontologies. The results are then translated back into the integrating ontology language and presented to the client.

In this architecture, the data source ontologies are designed in a comprehensive and task-neutral fashion, without regard to the customer and application; the integrating ontology, on the other hand, is custom crafted to support a specific customer and a specific set of tasks. As a result, the data source ontologies can be reused for multiple applications, assuming a suitable integrating ontology and a corresponding set of translation rules is developed. Some of the translation rules will be

sufficiently general that they can be reused across applications, as well.

As a first test of the SemanticIntegrator architecture, we built the SIMA system (SemanticIntegrator for Mobile Agents) to demonstrate information integration in the context of planetary exploration operations. The Mobile Agents project (Clancey et al. 2004) involves simulating planetary surface exploration with collaborating teams of humans and robots deployed in Martian and lunar analog environments on Earth, such as the Utah desert and the Haughton impact crater in the Canadian Arctic. During these simulations, scientific data are collected, including geological samples, photos, and voice recordings, along with GPS-ascertained coordinates of the collection sites. Using SIMA, we integrated this source of field-collected scientific data with two additional information sources: 1) satellite imagery and GIS information from Microsoft's TerraServer¹, and 2) physical and optical properties of minerals from a web-based mineralogy database². Using the SemanticIntegrator architecture, the data source ontologies, and the applicable rules, the SIMA sources can be reused for a different integration task with much less effort than required to integrate from scratch.

An interesting aspect of this work relates to our experiences in developing cross-ontology mapping rules for a real-world application. We used the rule language from Jess³, the Java expert system shell, to specify ontology mappings. Though this language was sufficiently flexible for our application and permitted sophisticated mappings, it was difficult to specify, understand, and maintain the mappings. A review of different approaches to cross-ontology mapping confirms that these problems broadly apply to both commercially and academically developed mapping approaches.

Commercial semantic integration platforms (e.g., Software AG's EntireX XML Mediator⁴) provide simple, understandable term-to-term ontology mapping capabilities, but the mapping constructs are not sufficiently powerful for more complex real-world applications. In these cases, commercial tools often support an external language call feature to use C or Java code to perform mappings. However, these types of 'procedural' mappings are even less easy to reuse and maintain than mappings formulated within the native mapping language.

More sophisticated representations, such as those developed within the academic ontology mapping community (e.g., Beneventano et al. 2002; Franconi and Tessaris 2004) are more capable of specifying complex real-world mappings. However, they can be even more difficult to use than our Jess-based language, and more unwieldy for users and maintainers of semantic integration systems. Other approaches simplify maintenance problems while sacrificing accuracy using automatic methods (Bouquet, Serafini, and Zanobini 2003).

What is needed is a language that strikes the right balance – one that is relatively simple to use, yet allows

users to specify declaratively the more sophisticated mappings sometimes required for, real-world applications. Developing such a language will be a focus of our future research.

Semantic integration technologies show great promise for addressing NASA's complex information management needs. However, more work is necessary to design and implement mapping approaches that address the large-scale application needs that face NASA and other major data providers and consumers. Aside from the need for sophisticated mappings, usability and maintainability are primary concerns in real-world settings.

Notes

1. See terraserver.microsoft.com
2. See webmineral.com
3. See www.jessrules.com
4. See www.softwareag.com

References

- Beneventano, D.; Bergamaschi S.; Castano, S.; De Antonellis, V.; Ferrara, A.; Guerra, F.; Ornetti, G.; and Vicini, M. 2002. Semantic Integration and Query Optimization of Heterogeneous Data Sources (Invited Paper). In *1st Int. Workshop on Efficient Web-based Information Systems (EWIS 2002)*, Montpellier, France.
- Bouquet, P.; Serafini, L.; and Zanobini, S. Semantic coordination: a new approach and an application. In Proc. of the 2nd International Semantic Web Conference (ISWO'03). Sanibel Islands, Florida, USA, October 2003.
- Clancey, W.J.; Sierhuis, M.; Alena, R.; Crawford, S.; Dowding, J.; Graham, J.; Kaskiris, C.; Tyree, K. S.; and vanHoof, R. 2004. Mobile Agents: A distributed voice-commanded sensory and robotic system for surface EVA assistance, In *Engineering, Construction, and Operations in Challenging Environments: Earth and Space 2004*, ed. R. B. Malla and A. Maji, 85-92. Houston, TX: ASCE.
- Crubézy, M.; Pincus, Z.; and Musen, M.A. 2003. Mediating Knowledge between Application Components. In *Proceedings of the Semantic Integration Workshop of the Second International Semantic Web Conference (ISWC-03)*, Sanibel Island, Florida.
- Franconi, E; and Tessaris, S. 2004. Rules and queries with ontologies: a unified logical framework. In *Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR-04)*.
- Noy, N.F; Doan, A.; Halevy, A. Y.; ed. 2005. Special Issue on Semantic Integration, *AI Magazine* 26(1).