OpenII: An Open Source Information Integration Toolkit

Len Seligman[1], Peter Mork[1], Alon Halevy[2], Ken Smith[1], Michael J. Carey[3],
Kuang Chen[4], Chris Wolf[1], Jayant Madhavan[2], Akshay Kannan[4], Doug Burdick[1]

[1]The MITRE Corporation, [2]Google,
[3]University of California at Irvine, [4]University of California at Berkeley

{seligman, pmork, kps, cwolf, dburdick}@mitre.org, {halevy, jayant}@google.com,
mjcarey@ics.uci.edu, {kuangc, ak}@cs.berkeley.edu

ABSTRACT

OpenII (openintegration.org) is a collaborative effort to create a suite of open-source tools for information integration (II). The project is leveraging the latest developments in II research to create a platform on which integration tools can be built and further research conducted. In addition to a scalable, extensible platform, OpenII includes industrial-strength components developed by MITRE, Google, UC-Irvine, and UC-Berkeley that interoperate through a common repository in order to solve II problems. Components of the toolkit have been successfully applied to several large-scale US government II challenges.

Categories and Subject Descriptors

D.2.12 [**Software Engineering**]: Interoperability - *data mapping*

General Terms

Algorithms, Design

Keywords

Information Integration, Data Exchange

1. INTRODUCTION

Despite advances in both commercial and research tools, II solutions are still too costly and labor intensive to build. Among the large enterprises we have supported, it is still common for II efforts to take many staff months and even years. This is simply unacceptable, given the accelerated tempo of modern business and government operations.

There are several reasons that building II solutions is still hard. First, many tools cover only a small part of the problem space (e.g., matching without addressing mapping, code generation for XSLT but not SQL or XQuery). Second, these point solutions are difficult to tie together; ironically, it is difficult to integrate integration tools, each of which has a different representation for schemas and mappings. Third, while some vendors (e.g., IBM and Microsoft) are moving to support integration of their own tools, they have not published their approaches or interfaces. There are obvious advantages to user organizations and small software companies to an *open* framework that allows combining II tools. Fourth, the more full featured tool suites are unaffordable to low resourced organizations, many of which have a great need for II and are willing to share (e.g., among non-governmental organizations that do nature conservation), but cannot pay. Fifth, the needs of II applications vary quite a bit, and as a result, they are hard to satisfy with a single system or product. The goal of OpenII is to supply 90% of the code necessary for an II application, while allowing easy customization for specific needs. Finally, the lack of an open II framework places a burden on researchers. Each researcher wastes resources creating importers, exporters, a user interface, and other components not relevant to the research. In addition, because there is no larger framework to plug their tools into, research successes are too hard to transition into practice.

We developed OpenII to address these challenges. It consists of a scalable, extensible platform as well as a number of industrial strength II tools built on top of the platform. The proposed presentation will describe the platform and component tools, future plans, and our experiences applying the tools to real II problems.

2. THE OPENII PLATFORM

Figure 1 shows the OpenII architecture. The platform includes:

- *Yggdrasil*[1] shared repository, which implements a neutral, extended entity relationship metamodel (called M3) for both schemas and mappings. While there are many such repositories for schemas, our model-independent representation of mappings is novel. Because of this design, much mapping information is reusable across efforts that use different technologies to implement executable mappings (e.g., SQL vs. XQuery). Yggdrasil is implemented on top of a relational database; currently, one can use Postgres for multi-user, enterprise repositories or IBM's lightweight, open source Derby database for a single-user, zero-configuration version. Yggdrasil is a substantial extension of the repository in [5].

- Importers/Exporters: OpenII includes a variety of importers and exporters including XML Schema, SQL DDL, OWL, and Excel spreadsheets. OpenII also includes facilities for quickly developing custom importers.

SIGMOD'10, June 6–10, 2010, Indianapolis, Indiana, USA.
Copyright 2010 ACM 978-1-4503-0032-2/10/06...$10.00.

[1] In Norse mythology, Yggdrasil (pronounced IG-drah-sill) is the Tree of Life at the center of the universe; this seemed an appropriate name for the shared repository through which all OpenII tools share knowledge.

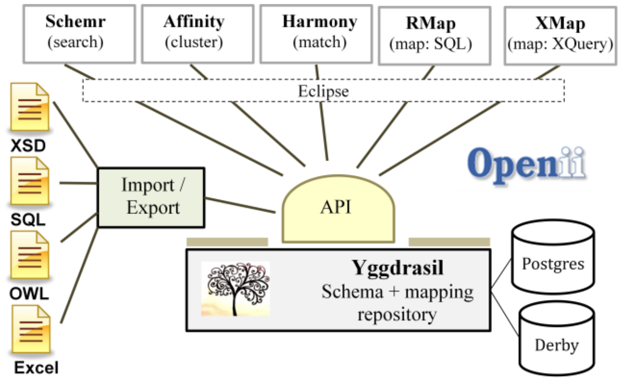

Figure 1. Architecture and Component Tools

- Component tools interoperate by sharing knowledge through the repository via a web service, Java, or other APIs specific to certain tool types (such as schema matchers). These interfaces build upon the Integration Workbench [4], which was first used to integrate a schema mapping tool (BEA's AquaLogic Data Services Platform) with a schema matcher (MITRE's Harmony) [1]. Tools can connect to one or more Yggdrasil repositories.

All component tools are built as plug-ins to IBM's popular open source Eclipse development environment, which provides developers with a familiar interface and which supports easy incorporation of new tools.

3. COMPONENT TOOLS

In addition to the platform, OpenII includes several tools that address particular II challenges.

Schemr schema search [2]. Schemr helps users discover and visualize relevant schemas in Yggdrasil, thereby encouraging sharing and reuse. Users search by keywords or by example --- an existing schema (or fragment thereof) --- or both. Schemr's novel algorithm retrieves candidates via disjunctive text search, and ranks candidates by a structurally-aware tightness-of-fit metric based on schema matching techniques. Schemr presents search results as visualizations that support interactive exploration of schema structure and match quality

Affinity schema clustering and visualization[8, 9]. Affinity provides Chief Information Officers (CIOs) and information architects with a clear overview of enterprise information assets and opportunities for productive data sharing. Affinity applies hierarchical agglomerative clustering to a selected schema set, rendering schemas in correlated panes as: a) a hierarchical dendrogram view, and b) a 2-dimensional point placement view. User interaction permits dynamic exploration of clusters, providing increasingly detailed superimposed clustering annotations. Affinity also provides a seamless transition to a Harmony view of pairs of selected schemas.

Harmony schema matcher[6]. Harmony is a hybrid matcher that combines the scores generated by a collection of element-level linguistic matchers. Harmony is the first matcher of which we are aware to exploit textual definitions of schema elements using sophisticated linguistic processing. Its GUI contributes flexible mechanisms for filtering match results and a variety of features that support iterative development. Harmony can also be easily extended with additional match heuristics. Harmony is now an industrial strength tool, efficiently handling schemas of 10^4 elements each (plus accompanying text documentation) [9].

RMap and *XMap* are data-exchange code generation tools for SQL and XQuery, respectively. These tools interpret a set of correspondences as a collection of tuple-generating dependencies (tgds) [7]. (Such correspondences are generated by Harmony and verified by a human using Harmony's GUI). Both tools include a GUI that allows the user to guide code generation, which is important when multiple tgds can be inferred from a given set of correspondences or when custom transformations need to be introduced (e.g., to convert across units of measure or datatypes). Because these tools share a common representation of mappings, knowledge gathered by RMap can be leveraged by XMap, and vice versa.

More tools are under development, including *Unity*, which will ease creation of mediated schemas.

4. FUTURE WORK

Although the existing OpenII platform and toolkit have many capabilities, there remain gaps and opportunities for further innovation with OpenII.

Schema Evolution. One potential extension of OpenII would be to address the issue of *schema evolution* [3]. Ideally, data exchange artifacts should be updated to reflect changes in the underlying schemas on which they are based. For example, the XQuery output from XMap to perform data exchange between source.xsd and target.xsd should be updated to reflect the evolution of source.xsd. Additionally, the set of correspondences between source.xsd and target.xsd previously identified by Harmony (and used as input to XMap) should be updated as well. Although complete automation of the updating process may be unobtainable, identifying the necessary updates so the integration engineer can make the appropriate modifications should require less time than starting the whole process from scratch, which is the current state of the practice. The repository underlying OpenII facilitates storing previous versions of schemas. Semi-automated support is needed for 1) detecting schema changes, 2) identifying mappings that may no longer be valid [10], and 3) regenerating the data exchange code.

Business Intelligence for II. Most existing II toolkits focus on developing data exchange code as the end product of the integration activity. However, for many II projects, the goal is not data exchange, but instead identifying situations where data sharing makes sense. Instead of asking "***How*** do I exchange data between Schema A to Schema B", these applications are asking the more fundamental question "***Does it makes sense*** to exchange data between schema A and schema B?" In these scenarios, the goal of the integration exercise is to drive investment decisions in enterprise data resources. Examples of such questions are:

- "Can System B replace the functionality of System A, since System B contains enough of A's data?"
- "How much effort would it take to move data from System A to System B?"
- "Which systems are similar enough to System A so that the cost of data exchange is reasonable?"

We have encountered several examples of applications which ask such *analytical* questions in practice, with some mentioned in Section 5. However, existing II toolkits provide little support for answering such questions. Some of these questions, like the third question above, involve conceptually evaluating all candidate schema pairs for sharing, which is prohibitive in practice. For N schemas, we must run the Harmony schema matcher for each of the $O(N^2)$ schema pairs to estimate the cost of integration for each pair. For real-world enterprises with N numbering into the hundreds or thousands, this is infeasible. And worse, the approaches to reliably translate a set of correspondences output by Harmony to a cost estimate are lacking in practice.

The core issue is that existing information integration tools are not designed to answer exploratory *business intelligence (BI)* queries, and instead focus on optimizing the creation of mappings and data exchange code among previously identified source and target schemata. The distinction between data exchange code generation and the questions above are similar to the one between transactional workloads and analytical workloads found in business intelligence applications. The latter do not require exact answers (i.e., generated exchange code), but instead require efficient computation of an artifact easily digestible by executive decision makers.

The Affinity tool can be thought of as a first step toward bringing the BI paradigm to data integration. The clusters of similar schemas Affinity identifies provide an easily interpreted overview of an enterprise's data assets. Developing information integration analogues for other business intelligence queries is an area for future work.

5. APPLYING THE TOOLKIT

OpenII tools have been successfully applied to several US government II tasks. [9] describes one effort, whose goal was to help enterprise planners determine the overlap between two systems to decide if one could be retired. Another effort created mappings among a few organizations' sizable data models about critical infrastructure, such as airports and power plants. The US Air Force is using Harmony to match not only schemas but also large collections of code lists and their definitions. A couple of organizations are experimenting with Affinity to help them visualize the relationships among hundreds of schemas in their existing repositories. Other organizations are evaluating how OpenII might speed adoption of data exchange standards, by making it easier for existing systems to create messages conforming to the standards.

We encourage both developers and researchers to download OpenII (from http://code.google.com/p/openii/). We use the forgiving Apache license, which allows unlimited modification and even incorporation into vendor products. OpenII can also serve as a research platform, providing many useful components, so researchers can focus on the truly novel aspects of their work. Finally, we welcome additional collaborators. Please contact the authors if you are interested in contributing components.

6. ACKNOWLEDGMENTS

Thanks to David Allen, Craig Bonaceto, Raman Grover, Jeff Hoyt, Joel Korb, Maya Li, Catherine Macheret, Michael Morse, Arnie Rosenthal, Ken Samuel, and Beth Yost for their many contributions to OpenII.

7. REFERENCES

[1] Carey, M.J., S. Ghandeharizadeh, K. Mehta, P. Mork, L.J. Seligman, and S. Thatte, "AL$MONY: Exploring Semantically-Assisted Matching in an XQuery-Based Data Mapping Tool," *Proc. International Workshop on Semantic Data and Service Integration*, 2007.

[2] Chen, K., J. Madhavan, and A. Halevy, "Exploring Schema Repositories with Schemr." *SIGMOD* 2009.

[3] Curino, C.A., H.J. Moon, and C. Zaniolo, "Graceful database schema evolution: the PRISM workbench." *VLDB* 2008.

[4] Mork, P., A. Rosenthal, L. Seligman, J. Korb, and K. Samuel, "Integration Workbench: Integrating Schema Integration Tools." ICDE Workshops 2006: 3.

[5] Mork, P., L. Seligman, A. Rosenthal, M. Morse, C. Wolf, J. Hoyt, and K. Smith, "Galaxy: Encouraging Data Sharing Among Sources with Schema Variants." *ICDE* 2009.

[6] Mork, P., L.J. Seligman, A. Rosenthal, J. Korb, and C. Wolf, "The Harmony Integration Workbench," *Journal on Data Semantics*, vol. 11, 2008.

[7] Popa, L., Y. Velegrakis, R. Miller, M.A. Hernández, and R. Fagin, "Translating Web Data." *VLDB 2002*.

[8] Smith, K., C. Bonaceto, C. Wolf, B. Yost, M. Morse, P. Mork, and D. Burdick, "Exploring Schema Similarity At Multiple Resolutions." *To appear in SIGMOD 2010*.

[9] Smith, K., M. Morse, P. Mork, M. Li, A. Rosenthal, D. Allen, and L.J. Seligman, "The Role of Schema Matching in Large Enterprises." CIDR 2009.

[10] Yu, C. and L. Popa, "Semantic adaptation of schema mappings when schemas evolve." VLDB 2005.