

Information Integration Isn't Simple

Laura Haas, laura@almaden.ibm.com

IBM Almaden Research Center

Introduction

There are many information integration problems. Information integration is more than “data integration” [1], more than warehousing [3], more than “data exchange” [2]. These well-studied problems all address aspects of the overall information integration problem, as does research on discovery, ontologies, data cleansing, entity resolution and even, to some extent, search. In both the research and the commercial worlds, solutions to each of these problems come in many flavors. Thus, a key challenge to anyone attempting to integrate information is, first, to identify the technologies and tools needed, and then, to compose the integration engines and tooling into a usable system that accomplishes their goals. In this paper, we examine briefly the multiplicity of solutions that might be brought to bear to address a typical integration task. We then suggest specific areas of research to address the “meta-integration” challenge.

The “Meta-Integration” Challenge

There is a new form of heterogeneity plaguing would-be consumers of data today: the many technologies and tools available for information integration. There are two common styles of integration engine: federation [4] and consolidation. This latter category consists of the tools to build and maintain warehouses, including *ETL* (extract, transform, and load) engines [5, 6] and replication engines [7]. While many integration tasks can be handled by any of these major engine types, they have quite different characteristics in terms of performance, data currency, security and compliance, application development time, maintenance, and so on. Other means of integrating information are also available, such as search, which easily brings together diverse information, but offers less precision in terms of query specification and less power and flexibility for result-structuring. Given all these choices, the consumer can often spend months just investigating alternatives and planning an integration strategy.

Of course, before an integration engine can be brought to bear, someone must figure out what information is needed, and where to obtain it. This is rarely as simple as it sounds. Often there are multiple sources for the same information, some more current, more complete or more accurate than others, some with overlapping content, perhaps modeled differently. Descriptions of the information and its properties are often missing or purely operational. Even when metadata does exist, it too comes in many different forms, and is rarely complete. Many tools have been developed to help with finding information, cleansing [10], and identifying data that refer to the same objects [9]. Yet these tools often rely on metadata that may not exist, or may be in the wrong format, requiring information-gathering and transformations before the user can even begin designing the integration. Still other tools help the user design the mappings [8] between source and target – but different tools may be needed, depending on the type of

integration engine. For example, one tool may be needed to generate SQL for a federation engine, while a different one supports the design of ETL jobs.

Integrating the Integration Tools

This proliferation of integration engines and tools poses major challenges for consumers. For example, while there are rules of thumb for when to use federation versus ETL, it is easy to make the wrong choice. In truth, most integration scenarios require the use of multiple integration methods. So the question becomes not which engine to use, but which engine to use where – and how to combine them.

We need a way to choose between the techniques automatically, based on the desired service levels across a set of dimensions such as performance, latency, precision. In essence, we need an *integration optimizer* that can take a nonprocedural statement of the desired integration, and produce the configured integration system. Many research questions are implicit in this statement. How can we specify the integration desired? What dimensions and service levels should be considered? How do we create a suitable objective function for optimization? How do we model the various “operators” (in this case, integration technologies), so that we can optimize the objective function? Are the integration techniques we have today the right primitive operators for such a system?

While such an optimizer would simplify the integration design task, there is still the need to discover sources and mappings, resolve object identities, discover and resolve conflicts – and again, a plethora of independent tools and overlapping technologies. Can these “stages” of an integration task be abstracted and made part of the specification hypothesized above? Can metadata help in unifying these technologies? What types of metadata are needed, and how would they be used?

Unifying and simplifying our integration technologies so that they work for real integration scenarios should be a key focus of the next generation of research.

Bibliography

- [1] Maurizio Lenzerini: Data Integration: A Theoretical Perspective. PODS 2002: 233-246
- [2] Phokion G. Kolaitis: Schema mappings, data exchange, and metadata management. PODS 2005: 61-75
- [3] Surajit Chaudhuri, Umeshwar Dayal: An Overview of Data Warehousing and OLAP Technology. SIGMOD Record 26(1): 65-74 (1997)
- [4] Laura M. Haas, Eileen Tien Lin, Mary Tork Roth: Data integration through database federation. IBM Systems Journal 41(4): 578-596 (2002)
- [5] <http://ibm.ascential.com/products/datastage.html>
- [6] <http://www.informatica.com/products/powercenter/default.htm>
- [7] Marie Buretta: Data Replication: Tools and Techniques for Managing Distributed Information. John Wiley and sons (1997)
- [8] Renée J. Miller, Laura M. Haas, Mauricio A. Hernández: Schema Mapping as Query Discovery. VLDB 2000: 77-88
- [9] Sheila Tejada, Craig A. Knoblock, Steven Minton: Learning object identification rules for information integration. Inf. Syst. 26(8): 607-633 (2001)
- [10] Mauricio A. Hernández, Salvatore J. Stolfo: The Merge/Purge Problem for Large Databases. SIGMOD Conference 1995: 127-138