A Consumer Viewpoint on “Mediator Languages – a Proposal for a Standard”

Arnon Rosenthal, Eric Hughes, Scott Renner, Len Seligman
The MITRE Corporation, Bedford MA, 01730
{arnie, hughes, sar, seligman} @mitre.org

OQL is proposed as the standard language for requests to mediators. We agree with much of the discussion and will not enumerate points of agreement. However, it is not clear that such a standard would benefit the government agencies whom we advise. Until such a consumer sees their concerns addressed, they will either react with disinterest or suspect flaws.

These consumers are interested in standards that add power to their information systems, not in better mediators. They avoid most query language debates, accepting the current standard. Once persuaded, these agencies may require use of the standard. However, they first must be helped to see how *incremental* changes to their installed base of legacy systems will yield short or medium term benefits.[1] We wrote this response to illustrate the additional arguments needed to motivate such agencies.

CONCERNS AND SUGGESTIONS

The Dog that Didn’t Bark: Sherlock Holmes once observed the curious behavior of the dog who didn’t bark when the murderer entered. Customers may be dismayed to see that Section 2.3 discusses ASN.1 and KIF, but not SQL. This leaves the impression that the workshop focused on new technology, giving little weight to existing clients, servers and skills.

Are mediator researchers a community that needs a standard? Is it more important for mediators to be consistent with each other, or with application development and client-side data access tools?

One argument for using OQL is that mediators need to serve applications that view the world as hypertext or as distributed objects (e.g., managed by CORBA or DCOM). SQL mainly sees the world as distributed tables, while OQL works with all three views.

What does a Mediator-Based System Look Like? To get a warm feeling about large-scale use of mediators, consumers and their technology advisers (e.g., MITRE) need to see how mediators coexist with existing software within a system architecture. An expanded proposal might describe a four-level architecture, with clients, a distributed query processor, source-specific mediators, and varied sources.

Bridges: What sort of bridging is needed to cope with a client who sends SQL to a mediator that requires OQL, or to cope with a mediator who sends OQL to a client that speaks only SQL? How will these bridges affect performance? How will they affect operations other than queries (e.g., updates, trigger definitions, security)? Each language has features that the other lacks — how will the residue be handled? How will clients see mediators that conform to the different proposed levels of the standard?

Can a consumer expect bridges from reputable vendors for their platforms? How will they resolve finger-pointing between bridge and DBMS vendors? Most importantly, how can they gradually move to this new architecture, instead of committing to an all-or-nothing migration plan?

A Negative Scenario: In an organization that does not use OQL, most clients will speak SQL. Can this organization bring mediators into its environment? This appears costly to achieve (tools and skills would need to be upgraded) and to manage (due to heterogeneity). Managers may see better return on investment elsewhere, and forgo mediators. Or they may ignore the standard and allow only SQL, despite its inferiority for certain purposes.

[1] In an ideal world, one could ignore transition issues. But such an ideal world would not need mediators.

ALLAY THE CONCERNS BY LIMITING THE ROLE OF THE PROPOSED STANDARD?

Perhaps the proposal should focus on more limited roles where the benefits are clearer.
OQL as *one* standard interface for mediators: The nicest thing about standards is that there are so many to choose from. Perhaps the proposal should be revised to envision an environment where mediators speak a variety of languages — various levels of both OQL and SQL. One would introduce OQL only where its power is needed, while imposing no burden in areas of the system where SQL-speaking mediators fit well with clients. Parts of Section 3.1 could be recast to identify situations where OQL is a more appropriate interface.

OQL as a standard within the mediator research community: As a fallback, instead of promoting OQL as a standard for deployed mediators, perhaps it should be used among mediator researchers. This would facilitate technology demonstration projects that integrate multiple mediators, each of which may only address a small portion of the gap between user applications and information sources.

An additional argument for OQL is that vendors may be more open to extensions proposed by mediator researchers, compared to SQL vendors with their much larger installed base. Such extensions might help describe desired quality of service, manage annotated data (e.g., with metadata from Section 3.4), or reference the ontologies used in understanding each source's data semantics.

ACCESS TO NON-DATABASE SERVERS

Consumers (especially non-technical management) are frustrated when data in legacy servers is inaccessible; often this is a greater concern than the inability to capture and share newer kinds of information (e.g., rules). The proposal already contains substantial discussion on servers with lesser capability; it should probably be elevated to a major focus.

Mediators should make the presence of such legacy servers transparent to clients. The client-side query language should be driven by the client's needs, not by the server. However, if only for performance reasons, the two are rarely decoupled completely. The proposal's Level 0 of OQL does suit navigational requests, which are a useful bridge to legacy servers.

To implement mediators over such servers, we see several approaches that deserve support. The simplest (but least flexible) approach is to write batch extraction programs and query the data in a warehouse. A second approach, being pioneered by OLE/DB, is to standardize several levels of interfaces, with the intent that any server can be wrapped to support some level; a market can then emerge of mediators that map popular query languages down to the standard levels. The most general approach, e.g., in Tsimmis from Stanford, is to describe the capabilities of each server in a declarative language, and have a mediator that can use that information to split a request into a part that the server can handle, plus a residue.

CONCLUSION

The original proposal left unanswered several questions that are important to user organizations. We raise them as a step toward obtaining a proposal that could be accepted and supported by our sponsors.

***Reply from L. Raschid*:** The response raises some interesting questions. The common standard is proposed mainly for use within the mediator research community. It is not intended to be a public standard, for example, for use by data providers. The core (relational algebra-like) mediator language is similar in both SQL and OQL, and there should be little difficulty in moving from one to another syntax. However, the extensions that are envisioned are in the direction of OQL. There is no advantage to inventing a third syntax. For any proposed extension, we expect that its expressibility in SQL will be one of the factors that will be considered.

This work was sponsored by the US Air Force Electronic Systems Center (ESC) System Integration Office (SIO) under contract AF19628-94-C-0001.