

Responses to issues raised in telcon of 24 July 2003

Greg Riccardi riccardi@cs.fsu.edu

31 July 2003

I tried without much success to address two significant issues in the telcon. I consider these more carefully in this document.

How will we assess the completeness, correctness and other semantic characteristics of the operations supported by DAIS services?

What is the relationship between multi-shot and single-shot interaction strategies and the maintenance of state within services?

These two questions are quite interrelated and are both addressed in the emerging conceptual model with datasets. The semantic difficulties arise because services maintain state within and between requests. We would not be concerned with semantics if each service were single shot and returned all of the state in the response.

Suppose a DAIS service supported exactly the JDBC operations and state. The client would assume the burden of semantic correctness. A JDBC connection is state-full with precise operational semantics. If a data activity session had exactly the JDBC methods, it would be semantically complete and consistent. It would not, however, support the DAIS goals.

A service that maintains state for its clients must ensure the correctness of that state. In particular, a DAIS service must provide a naming strategy for retained state and guarantee that named objects retain their semantics. Much of the difficulty exposed by these two questions is in the careful definition of the semantics of state.

DAIS members have been struggling with creating a semantically rich way for a service to create, retain and name objects, and a solution is emerging. One fundamental principle of DAIS is that query results are not always returned to the requesting client, so we are stuck with requiring state creation and maintenance. Fortunately, the notion of dataset provides a simple strategy for being precise about state.

It makes little difference whether the client executes one request at a time (single-shot) or executes multiple requests (multi-shot). In both cases, requests must reference state that is part of the request (e.g., in a bulk load), and state that was created by previous service requests (e.g., by an asynchronous query execution). Requests must also be able to reference state that is maintained by other services.

One additional naming capability required for multi-shot requests is the ability to reference state created in response to the request. E.g., a request to execute a query, transform its result, and deliver the transformed values to a third party.

Treating a request document as a dataset creates a naming scheme that can be used with state created within a single (multi-shot) request, state retained between requests, and state that is part of response documents. I'm working on illustrative requests and responses.

A sample problem of retaining state in data services

As an example, suppose a client of a data activity session opens a transaction and requests asynchronous execution of a query. The query result must be retained for some period. The service responds with an identifier that provides access to the query result.

Does the query result persist beyond the end of the transaction? Can the client commit the transaction before the query execution is completed? What is the response of the service to a request using the identifier?

In JDBC the client who wants the result to persist beyond the end of the transaction would explicitly transform the query result into a disconnected RowSet. Thus no ambiguity arises. ResultSet objects are connected to transactions and disconnected RowSet objects are not.

Specifying semantics in the conceptual model

The emerging DAIS conceptual model specifies that the execution of a query results in the creation of a dataset object. If necessary, some service instance may provide access to the dataset. The model specification must declare whether the dataset object is bound to the transaction context that created it or not.

My opinion is that the semantics must provide disconnected datasets as the default. The values contained in a dataset are a snapshot of the data resource. Thus datasets can be moved and duplicated with no semantic difficulty.

The DAIS specification must encourage implementers to create scalable services. As such, long transactions must be discouraged and state must be maintained independently of service instances as much as possible. The Web services community argues strongly and convincingly that maintaining state in service instances destroys scalability. The database community argues strongly and convincingly that maintaining state outside of databases destroys consistency and durability.

The Web services community advocates maintaining state in the messages (service requests and response documents) and the database community advocates maintaining state in databases. Both communities argue vehemently against maintaining state in service instances.

A data activity session may provide session-based connected datasets through extensions to the document schema. A request for query execution may include a tag or attribute that declares the dataset to be dependent on the transaction context.