

Design Tip #106 Can the Data Warehouse Benefit from SOA?

By Ralph Kimball

The Service Oriented Architecture (SOA) movement has captured the imagination, if not the budgets, of many IT departments. In a nutshell, organizing your environment around SOA means identifying reusable services, and implementing these services as centralized resources typically accessed over the web. The appeal comes from the promised cost savings of implementing a service only once in a large organization, and making the service independent of specific hardware and operating system platforms because all communications take place via a neutral communications protocol, most often WSDL-SOAP-XML.

Well, pretty much all of these advantages of SOA can be realized, but early SOA pioneers have learned some valuable lessons that give one pause for thought. The names of these lessons are data quality, data integration, and governance. To make long story short, SOA initiatives fail 1) when they sit on a platform of poor quality data, 2) attempt to share data that is not integrated across the enterprise, and 3) are implemented with insufficient thought given to security, compliance, and change management. SOA architects have also learned to “avoid the overly detailed use case.” Services meet the goals of SOA architecture when they are simple, conservative in their scope, and not dependent on complex business rules of an underlying application.

So, does SOA have anything to offer data warehousing? Can we identify “abstract services” in the data warehouse world, commonly recognized, simply stated, and independent of specific data sources, business processes, and BI deployments? I think we can.

Consider the relationship between the dimension manager and the fact provider. Remember that a dimension manager is the centralized resource for defining and publishing a conformed dimension to the rest of the enterprise. A master data management (MDM) resource is an ideal dimension manager but few of us are lucky enough to have a functioning MDM resource. More likely, the data warehouse team is a kind of “downstream MDM” function that gathers incompatible descriptions of an entity such as Customer and publishes the cleaned, conformed, and deduplicated dimension to the rest of the data warehouse community. The subscribers to this dimension are almost always owners of fact tables who want to attach this high quality conformed dimension to their fact tables so that BI tools around the enterprise can perform drill across reports on the conformed contents of the dimension. If the vocabulary of this paragraph is unfamiliar, then you have some reading to do! See the [Data Warehouse Lifecycle Toolkit, 2nd Edition](#) book, or the recent series of introductory articles I have written in the last year for *DMReview*.

Every dimension manager publisher needs to provide the following services to their fact table subscribers. *Fetch* means that fact table provider pulls information from the dimension manager, and *Alert* means that the dimension manager pushes information to the fact providers.

- Fetch specific dimension member (we assume in this and the following steps that the dimension record has a surrogate primary key, a dimension version number, and that the information transmitted is consistent with the security and privacy privileges of the requester).

- Fetch all dimension members.
- Fetch dimension members changed since specific date-time with designated SCD Types 1, 2, and 3.
- Fetch natural key to surrogate key correspondence table unique to fact table provider.
- Alert providers of new dimension release. (A major dimension release requires providers to update their dimensions because Type 1 or Type 3 changes have been made to selected attributes.)
- Alert providers to late arriving dimension members. (Requires fact table provider to overwrite selected foreign keys in fact tables).

These services are generic to all integrated data warehouses. In a dimensionally modeled data warehouse, we can describe the administrative processing steps with great specificity, without regard to the underlying subject matter of the fact or dimension tables. That is why, in SOA parlance, dimensional modeling provides a well-defined reference architecture on which to base these services.

These services seem pretty defensible as meeting SOA design requirements. An interesting question is whether a similar set of abstract services could be defined between the fact provider and the BI client. How about “Alert client to KPI change?” Maybe this is possible. I’ll follow this design tip with another if this approach pans out. In the meantime, I’d suggest reading *Applied SOA: Service-Oriented Architecture and Design Strategies* (Rosen, et al, Wiley 2008).