BEA AquaLogic
Data Services Platform

Jose Luis Ambite

[Based on slides by Mike Carey]

Declarative Integration via XML & XQuery

Requirements
- A standard for data format and data interchange
- A standard for describing and modeling data
- A standard for interfacing into applications
- A standard for querying both relational and non-relational data
- A standard Java programming model (read + write)
- A standard for publishing consumable services

Standards
- XML
- XML Schema
- Web Services
- XQuery
- SDO (Service Data Objects)
- Web Services

Data Services: BEA AquaLogic DSP

- Data services group operations related to (coarse-grained) business entities
- Logical models capture data access and integration complexity one time
- Uniform data model, programming model, and API for all enterprise data
- Service clients may be portals, business processes, composite applications...

Example: Customer Profile Data Service

Service Modeling (Logical Services)

Service Modeling (Physical Services)
Data Service – "Get All" Read Method

```xml
<![CDATA[
for $cust in cus:CUSTOMER() return <tns:CustomerProfile>
  for $ord in cus:getORDER($cust) return <ORDER>
  ...
</tns:CustomerProfile>]]>
```

ALDSP Query Processing Overview

- Inline function composition
- Split sorting and grouping
- SQL sub-query pushdown to relational sources
- Distributed join optimization
- Slow data source handling
- Support for timeout and failover
- Optional function result caching
Query Processing, Example (getProfile)

Key Point: It’s left up to ALDSP to decide how to optimize and execute the getProfile data service call. (Just like with SQL: just say what, don’t say how.)

Query Processing, Example 2 (query getProfile)

Key Point: Declarative data services, such as those built with ALDSP, support the performant reuse of data services.

Updates: Service Data Objects (SDO)

ALDSP inspired by MultiBase (Early 1980’s)

- One of the first DDBMS projects to relax the homogeneity assumption (vs. Ingres*, R*, ...)  
- Interesting foundation and technical contributions:
  - Functional data model (i.e., “everything is a function”)  
  - Prehistoric objects with identity  
  - Functions model attribute access, relationship navigation  
  - Model realized via the DAPLEX query language  
  - Important technical achievements  
  - Function-based model to normalize relational, network, and other models  
  - Early results on federated query processing, sets/multisets, and more  
  - Note: This is why computer science students should study “history”...  
  - Functions everywhere –> Web services

Summary of Information Integration

- Components of Information Integration  
  - Understand sources, terminologies  
  - Cleanse, Record Linkage  
  - Define target-mediated Schema  
- Approaches to Information Integration  
  - ETL workflows  
  - Virtual Integration  
- Representation/Query language  
  - Relational  
  - XML/XQuery/Web Services