PolarLake Background

Founded in 2003

Offices in Dublin, London, New York

XML integration heritage – patents in XML processing

Warren Buckley – CTO and Founder
PolarLake Clients

6 of the top 10 Investment Banks

2 of the top 5 Prime Brokers

2 of the top 10 Asset managers

Control Pricing and Reference Data
Distribution and Integration with PolarLake
The need for PolarLake?

• Frustration with complexity, cost and time to market of ETL/EAI/DB replication solutions to distribute Pricing & Reference Data Distribution and Trade Integration
• Inflexibility of fixed data models
• Experienced Trade and Reference Data integration specialists

The PolarLake Benefits

• Control & Confidence in delivery – end to end transparency of data and trade supply chain
• Time to Market – up to 95% rule generation, 75% less time, 80% less resources than traditional approaches
• Manage ongoing complexity – purpose built for change and unstructured data. Practical approach to Data Management
Diverse Connectivity Patterns

Reference Metadata Aware

Execution of Data Supply Chain Policies

Confidence through Transparent Operations

Supporting constant change new data, formats and requirements

Diverse Connectivity Patterns

Data Consumers

Front Office
Trading

Middle Office
Risk Management

Back Office
Clearing and Settlement

Suppliers

Internal Applications

“Golden Copy”

Internal and Commercial

Vendor Feeds

Consumers

Internal and Commercial

Data Suppliers
Pricing, Reference Data & Trade Integration
PolarLake and Oracle

Why Oracle XMLDB
EDM – The default approach

1. Design Canonical Model
2. Map Sources to Model
3. Distributed from Model

Vendor A

EDM Vendor X

Vendor B
EDM – The default approach has issues

1. Design Canonical Model
2. Map Sources to Model
3. Distributed from Model

EDM Vendor X

1. Rigid Canonical Models
2. Never complete – ‘semantic’ mismatch
3. Source models change and new ones arrive
4. Poor at conflicting data and classifications
5. Cause of Data Loss
6. Many failed projects
PolarLake Data Management

- PolarLake Policy Engine
  - XQuery
  - Linkages
  - Metadata imports
  - Classification imports

Data Loading
- Generic XML Storage
- Change Tolerant

Vendor A Data
Vendor B Data

Oracle 11g – XML DB
Oracle 11g – Semantics

Customer Models
Vendor A Metadata
Vendor B Metadata
PolarLake and Oracle 11g

Oracle 11g

PolarLake Policy Engine

RDBMS
- PolarLake Solution
- Definitions

XMLDB
- Reference Data

Semantics
- Metadata
Scope

• Technology Platform
  – J2SE and J2EE
  – PolarLake Messaging Integrator (XML OS)
  – Primary port Linux (Solaris, Windows, IBM mainframes, etc.)

• Data volume
  – ~ 8 million documents a day
  – 4 types (pricing, ratings, reference, etc.)
  – Documents vary in size from 2K to 1M
    • trade management has less volume but more complex and larger documents)
Oracle XML DB

- Binary XML Storage
  - Unstructured XML Index
- XMLIndex
  - Performance improvements
- Java API – oracle.xdb.XMLType

- Combining XMLDB, Semantics and Oracle Text is extremely powerful.
SELECT rddsem.file, rddxml.doc, rddxml.src  FROM 

TABLE(SEM_MATCH(  
    ('?filetype :contains :Issue)  
    (?filetype :providedBy :Reuters)  
    (?file rdf:type ?filetype)',  
    SEM_Models('rdd'),  
    SDO_RDF_Rulebases('RDFS'),  
    SEM_ALIASES(SEM_ALIAS('','http://www.polarlake.com/rdd#'),  
    null)) rddsem,  

XMLTable('  
    for $i in ora:view("borec") return $i  
' COLUMNS 
    doc XMLTYPE PATH '.',  
    src varchar2(100) PATH '/xml/@src') rddxml

where rddxml.src=rddsem.file  
and contains(rddxml.doc,'$(Oracle) INPATH (xml/LONG_COMP_NAME) ') >0;
Conclusions

- XQuery and SPARQL joins are powerful.
- Performance improvements by pushing logic to the database tier.
- Mid Tier APIs look promising.
- Need to separate XML Query and XML Construction.
- Need to separate Procedural and Declarative approaches.
- Careful crafting of XQuery for performance is essential.
- Careful design of paths in XML Indices required.
- Oracle 11g Release 2 - XML in the database has arrived for real.
Questions

warren.buckley@polarlake.com