Real-Life Data Mart Design Challenges

Leslie Tierstein
Data mart design challenges addressed in this project:

- Dynamically defined dimensions
- Potentially 100’s of dimensions
- Loading XML data into the data mart
- An incremental refresh that includes updates
- Supporting multiple databases with a standard toolset and code base
- Designing the physical database to map to the BI tools’ business view
Data mart design challenges not addressed in this project:

- Performance tuning for very large databases (VLDB’s)
- Data cleansing and data integrity
- Integrating data from heterogeneous databases or multiple applications
The OLTP System

- **Service Catalog Management**

- An IT department defines a catalog of all the services it provides to users
  - End User Computing: Add Visio
  - Advanced Computing: Configure a server

- Users request a service
  - System tracks the service request, all tasks required to complete it
Requestor View – Upgrade Memory

- User fills out and submits an order (requisition)
IT View – Upgrade Memory

- User request has criteria for completion
- Request is routed to appropriate people for authorization and task delivery

<table>
<thead>
<tr>
<th>Services</th>
<th>Service Level Description</th>
<th>Standard Duration</th>
<th>Quantity</th>
<th>Unit Cost</th>
<th>Subtotal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Memory - Upgrade</td>
<td>3 business days</td>
<td>1</td>
<td>300.00</td>
<td>300.00</td>
<td></td>
</tr>
</tbody>
</table>

*Total Cost: 300.00*

*Standard Duration applies to delivery after any required authorizations have been completed.*

### Delivery Process

<table>
<thead>
<tr>
<th>Process Milestone</th>
<th>Due Date</th>
<th>Completed On</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Group Authorization</td>
<td>12/28/2006 4:00</td>
<td></td>
<td>In Progress</td>
</tr>
<tr>
<td>Delivery project for Computer Memory - Upgrade</td>
<td>01/02/2007 10:00</td>
<td></td>
<td>Pending</td>
</tr>
</tbody>
</table>
Service Designers’ View

- Standard information (duration, cost, delivery plan, authorizations) defined via UI
- Service-specific data entry requirements defined via “dictionaries”
The OLTP System

- Vastly simplified ERD of transactional data
- Metadata on service definitions is also maintained
- Where is form (dictionary-based) data?
Data Mart Requirements

- Data mart must include form data
- Refresh needs to run in nightly maintenance window
  - Minimize impact on global accounts
- Data mart must include data on both open (in progress) and completed requisitions and tasks
  - Refresh must perform “updates”
Extract XML data from the requisition BLOB

- Cognos8 includes Composite, which can extract data from XML
  - Performance is problematic
- Adding a custom plug-in in the ETL tool provided insufficient power/flexibility
- Custom Java program was needed

Would the custom program perform acceptably?
ETL Performance Studies

- 100,000 requisitions per month (an order of magnitude higher than current usage)

<table>
<thead>
<tr>
<th>Data Points</th>
<th>#</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average # of requisitions started per day</td>
<td>5000</td>
<td>Requisitions</td>
</tr>
<tr>
<td>Average # of entries per requisition</td>
<td>1.5</td>
<td>Entries per Requisition</td>
</tr>
<tr>
<td>Days from requisition start until it's closed</td>
<td>10</td>
<td>Days</td>
</tr>
<tr>
<td>Average # of tasks in a service</td>
<td>6</td>
<td>Tasks per Service</td>
</tr>
<tr>
<td>Average # of actions taken per task</td>
<td>1.6</td>
<td>Actions per Task</td>
</tr>
<tr>
<td>Average # of dictionary data elements</td>
<td>10</td>
<td>Data Elements per Dictionary</td>
</tr>
<tr>
<td>Average # of dictionaries per service</td>
<td>4</td>
<td>Dictionaries per Service</td>
</tr>
<tr>
<td>% requisition entries with reportable dictionaries</td>
<td>70%</td>
<td>Requisition Entries</td>
</tr>
<tr>
<td>Time to extract one data element to data mart</td>
<td>8</td>
<td>Milliseconds</td>
</tr>
</tbody>
</table>
## ETL Performance Studies

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requisitions open at any one time</td>
<td>50,000 Requisitions</td>
</tr>
<tr>
<td>Requisitions entries open at any one time</td>
<td>75,000 Requisition Entries</td>
</tr>
<tr>
<td>Tasks open at any one time</td>
<td>45,000 Tasks</td>
</tr>
<tr>
<td>Actions recorded per day</td>
<td>72,000 Actions</td>
</tr>
<tr>
<td>Requisitions entries changed in any one day</td>
<td>72,000 Requisition Entries</td>
</tr>
<tr>
<td>Requisition entries with reportable dictionaries changed per day</td>
<td>50,400 Requisition Entries</td>
</tr>
<tr>
<td>Requisition entry dictionaries changed per day</td>
<td>201,600 Requisition Entry Dictionaries</td>
</tr>
<tr>
<td>Requisition entry data elements changed per day</td>
<td>2,016,000 Requisition Entry Data Elements</td>
</tr>
<tr>
<td><strong>Time to extract changed data elements</strong></td>
<td>16,128 Seconds</td>
</tr>
<tr>
<td></td>
<td>269 Minutes</td>
</tr>
<tr>
<td></td>
<td>4.48 Hours</td>
</tr>
</tbody>
</table>
Averages are good, but know your outlyers
- If only to be able to warn users about potential issues
- Hotel/Hospitality Company:
  - 30% of services have 100-160 fields, with a maximum of 250. About 60% have 70-100 fields
  - There are dictionaries with about 40 fields, with a maximum of 43.

Inserts are easy, updates ... NOT
Updating Data Mart Contents

- Data Mart contents need to be updated
  - Tasks and requisitions are added to the data mart as soon as the requisition is submitted
  - Status of requisitions may change, as well as completion date and form data
  - Task status and info is updated as the task is performed (completed)

- Logical “update” <> DML UPDATE
Argh! Updating data mart contents is expensive
- Updates require indexes
- Inserts perform much better without indexes
- Deletes (from large tables) are a disaster

Avoid updates by placing data to be updated in a partition that is truncated
- Refreshed data is written to the same partition or to a permanent partition when complete
Still an issue

Partitioning not implemented because:
  - Small table size and growth rates
  - Lack of support in ETL tool
  - SQLServer (partition views only!)
  - What is partition key?

Using insert/update model

Time will tell if this is robust enough
● Table, View or Metadata?

– Design the dimensional model in conjunction with the business view of the model

• Date dimension(s)
• People dimension(s)
Date Dimensions (1)

- Date required as a dimensional key
  - Due Date – for task and requisition
  - Closed Date – for task and requisition
  - Start Date – for task and requisition

- Complete date and time may also be needed
  - SLAs and OLAs
How many database objects for the date dimensions?

- 3 tables, 3 subject areas
- 1 table, 3 views, 3 subject areas
- 1 table, 3 subject areas

Pros and cons of each approach

- ETL, debugging, user interface in ReportNet
- Where are relationships defined?
  - In the BI tool
  - In the database
Date Dimensions (3)

- Date dimensions are densely populated
  - Likely to have an attribute for every weekday date value

- Date dimensions can be statically populated
  - Rather than via fact/dimension build

- Solution: 1 table, 3 views, 3 subject areas
  - In Oracle, constraints can be defined in the view and created in the BI tool
People Dimensions (1)

- **Required Dimensions**
  - Customer – for requisition
  - Requestor – for requisition
  - Performer – for task
People Dimensions (2)

How many database objects?
- Do people fulfilling different roles intersect? And how much?
  - Customers and Requestors may overlap 100%
  - Performers are a much smaller group
- Implications for prompts and filters
  - Prompted filter should only show people who have filled the corresponding role
Dimensional Modeling

• Dynamic data model to accommodate form data
  – Other companies have done this, but their scenarios are not as complex
    • Oracle Noetix Views and Noetix data warehouse – Flex fields
How to model the dimensions which correspond to dynamically defined dictionaries

- Each attribute in a dictionary is a dimension – rejected
  - W-A-A-Y too many dimensions
- “Junk”, “Degenerate” or “Demographic” dimensions
- “Reference” dimensions
Junk Dimensions

- Populate the dictionary dimension using a unique combination of values for each dictionary across all requisitions.

- The fact table has a column for a relationship with each dimension – many will be “N/A”.

- A complex fact/dimensional build would be required.

- See Kimball, Design Tips 46, 48.
Reference Dimensions

- Each dimension has a 1:1 relationship with the fact table
- Best to not have basic analytical metrics and frequent query topics in the reference dimension
- Greatly reduces the size of the fact table
- See Kimball, Design Tip 86 (http://www.kimballuniversity.com/html/design tips.html)
Task Star Schema

- **Task**
  - **StartedDate**
  - **DueDate**
  - **ClosedDate**
  - **Queue**
  - **TaskType**
  - **Dictionary1**
    - **Dictionary2**
      - **Dictionary3**
        - **Dictionary..n**
  - **Performer**
  - **Service**
## Tool Suite

<table>
<thead>
<tr>
<th>Tool Type</th>
<th>Cognos Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETL tool (maps OLTP data to data mart)</td>
<td>Data Manager</td>
</tr>
<tr>
<td>BI Framework (maps database to end-user/business view)</td>
<td>Framework Manager</td>
</tr>
<tr>
<td>BI Power User Tool (allows end users to create reports/queries)</td>
<td>ReportStudio</td>
</tr>
<tr>
<td></td>
<td>QueryStudio</td>
</tr>
<tr>
<td>BI End User Tool (provides portal for running reports/queries)</td>
<td>ReportNet</td>
</tr>
</tbody>
</table>
Custom Java program is used in conjunction with Data Manager ETL.
Data Mart Refresh ...

... and metadata creation

- Dictionaries and their attributes must be mapped to tables/columns in the data mart
- This mapping is used in the BI business view
● Specify ETL for facts and traditional dimensions
  – Uses Cognos8 ETL tool, Data Manager
Transform the Data Manager project into a Framework Manager catalog

- (Sigh) two tools, two different sets of metadata
- Data Manager’s ETL destination tables become the basis of the BI tool’s metadata
For dictionaries ("junk" dimensions) use the Framework Manager API to create a business view

- Appropriate names for dictionary dimensions and data elements
Dimensions

- Dynamic dimensions
  - Based on dictionaries designated as reportable
  - Different for each site
- Universal/static dimensions
  - Found at all sites
● Create a report listing tasks performed, the performer, the service in which they were performed, and the current status.
Conclusions (1)

- XML extraction
  - Java programming was much more efficient than ETL tools available to us

- Dimensional modeling
  - Junk dimensions, reference dimensions, degenerate dimensions – who knew?
  - Subscribe to Kimball
  - Hire short-term consultants
Conclusions (2)

- **ETL Performance**
  - Satisfactory
  - Need sophisticated scheduler to track all jobs

- **Ad-hoc Report Performance**
  - Insufficient testing on the largest user data volumes (IMHO)
  - The software went live on January 31, 2007
  - Ask me in a few months
Update – Six Months In

- **Performance issues**
  - Reporting behavior is acceptable except for building filter selection lists on large fact table
  - Rework was required for customer load at client with largest customer base

- **Database issues**
  - Incompatibilities between Oracle SQL, SQLServer SQL, DB2 SQL, and Cognos SQL
  - Implement in database vs implement in tools
Update – Six Months In

- Functional/design issues
  - Should have gone with two people-based tables
    - Requestor/Initiator and Performer
  - Treatment of dimensions with all blank attributes needs more attention
  - Better use of framework to incorporate display definitions – better metadata required
About the Author

- Leslie Tierstein is a Senior Technical Architect at newScale, Inc, in Foster City California.
- She can be reached at ltierstein@earthlink.net or leslie.tierstein@newscale.com
- This paper is available online at: http://home.earthlink.net/~ltierstein