IBM Software

Business Analytics Forum 2010

See The Future Of Decision Making

November 3-5, 2010
Gold Coast – QLD – Australia

IBM Cognos 8 BI Modeling

Travis Murphy

Agenda

- Welcome

- Process for Model Development
  - Planning, Process
  - Modeling Practices and Priorities

- Q&A

IBM Cognos 8 BI Application Modeling Review Workshop Overview

- This workshop will provide guidelines & suggested areas to review when conducting a Metadata Modeling project
- This session will not address specific performance tuning processes, nor it will go into detailed OLAP modeling review

Project Start

- So, you’re about to start developing the Framework Manager metadata model.
- And you know ‘it’s easy’, as seen in your training and available presentations.

BUT – do not “import” anything just yet!

Project Start - cont

In fact - **CLOSE** your Framework Manager!

And instead…

- Open your Calendar and schedule following meetings:
  - Solution Architect
    - Objective: to understand the overall solution, all app components & Data sources, and overall priorities & expectations, collaboration and segmentation needs
  - Data Architect
    - Objective: to understand the data structures, data quality, aggregation capacity, logical and physical views, etc.
  - Business user community representatives and Business Analysts
    - Objective: to understand the needs and priorities for ad-hoc reporting and analysis, preferences re studios, business layer content and structure, row level security, bursting approach
    - Draft “test cases”, “test queries” to use during the development to validate concept
    - Define tooltips, descriptions, etc
- Make sure to set the expectations about the iterative dev. process

First Step

- Start from the “top”
  - Draft the Presentation layer structure, folder structure, packages structure, dmr structure
    - Still no hands-on FM modeling
  - Drill down to Business Layer Structure
    - Define the business rules, calculations, filtering and prompting, user-friendly object naming standards
  - Drill down to Import Layer Structure
    - Define the organization of multiple import sources (separate folders, separate namespaces, etc)
    - Understand the back-end potential (available objects) and (if) need for additional back-end work (e.g. views, aggregations, SP-s…)
    - Define what exactly, from where and why you need to use/import.
  - Define the rules for Governors, With clause usage, Data Source processing,
  - Document and draft everything – by the end of this, the picture of the final product should be almost clear

Development

- Now – Start from the “bottom”
  - Create Import layer import folder structure.
  - Keep the import layer CLEAN
    - Import ONLY what is absolutely needed
    - Set the correct JOINS
      - Remove ambiguity, create role playing dimensions
      - Resolve circular joins
    - Set the correct properties for data items
      - Fact, Attribute, Identifier, Aggregation, Default prompting type
    - Create appropriate parameter maps
  - Create TEST folder and develop set of test queries to check:
    - SQL quality, Group by behavior, min-sql generation, and general performance (explain plans) for representative queries, multifact scenarios (stitch quality with shared dim identification, and proper/efficient “where” placement, check the “query test response” to identify metadata callbacks)
  - Make sure that shared/common dimensionality is properly defined
    - Set the correct determinants on dimensions
    - Collapse, merge, simplify the dimensionality
  - Set the Object Security rules
  - Embed the data level security
  - Create the “Export/Externalization” layer with Queries needed (ref) for data externalization (transformer) – this is a separate topic but the dev process is similar.

Development continued

- Create Business Layer
  - Naming, user-friendly naming
  - Business rules definition (prompting, filtering, calculations)
    - Make sure to make the right stand-alone vs embedded object decision
  - Right decision re ‘pass-through’ or manual vs model query subject generation (there are many legitimate reasons for this)
  - Test the Schema (same as in Import Layer)
    - Create demonstration package based on Business Layer

Development continued

- Perform a review session with the Solution Architect, BA and Business users.
  - Demonstrate, solicit feedback, refine the approach
    - Ad-hoc, reporting, naming, Folder Structure, performance expectations, security, etc.
    - At this stage – the model will produce representative code, and Interaction with your DBA-s and Data Architects is very important.
      - As many as possible scenarios should be tested, code generated, prompting and filtering included – so that execution plans can be reviewed and opportunities for improving the back-end identified:
        - Statistics, Indexes, Aggregations, Views/SP-s to simplify the model & ensure predictable queries.
        - At this stage one should be able to identify (if any) expectations gap – for purely relational queries, and address the issues timely

Development continued

- Create Presentation Layer
  - Dimensions and Hierarchies
  - Measure dimensions
  - Scope Relationships review
    - Test aggregation results
  - Folder Structure for ad-hoc package
  - Create Final Presentation package (include Presentation Layer and Business Layer)
- Publish the package
  - Secure the access
  - Test Security
  - Create set of ad-hoc queries and managed reports to test:
    - Functionality, interface
    - Navigation and ease of use
    - Model/query/report performance
  - Refine the model as needed

Development continued

- Final presentation for
  - Solution Architect & BA and Business User reps
  - Final Changes
- Document the final product
- Deploy to Dev, open for User Access, Developer access