Introduction to MDM
Part 3 - Case study - From modeling to UI

Master Data Management

Education
Pierre Bonnet, IS Consultant, January 2012
Based on the work achieved with Ms Tran Thi Linh of IFI solution Hanoi
Last update: March 15, 2012

pierre.bonnet@orchestranetworks.com
skype: pierre.orchestra

www.orchestranetworks.com
Objective

Based on a case study about user acceptance test management

- Understand the core semantic data modeling
- Understand how the data model is used to build a concrete MDM system
- Understand what a logical data model is
Case study functional description

We want to build a MDM collecting all information related to user test acceptance of a system

- Test campaign definition
- Test campaign execution
- Test scenario
- Test step
- Test report
- Detailed test result
Business dictionary

(Test) Campaign definition
- A campaign is defined through an ordered set of test scenarios
- This set describes how to test a system

(Test) Campaign execution
- A campaign execution collects all outcomes related to a test of a system with referenced data
  - start date of the campaign execution
  - end date of the campaign execution
  - version of the system tested
  - configuration used to test the system
  - all status and bugs
Business dictionary

(Test) Scenario

- A scenario describes an ordered set of steps to test
- A scenario is defined through some reference data such as
  - functional objective
  - type: manual or automatic execution
  - category: test data, test data model, test permission, etc.
  - initial database: which database is used to achieve the test

Example of a scenario

- Name = "Create Permission for user, then check each user whether permissions are implemented correctly or not"
- Type = manual
- Category = data set
- Initial database = empty
Business dictionary

(Test) Step
- A step describes an atomic action that the user or the system must achieve to enforce a test
- A step is defined through some reference data such as
  - functional objective
  - all input and expected output test data required to achieve the test and to check whether outcomes are conformed

Example of a step
- Name = "Create Permission for user on dataset RealEstate"
- Category = create
- Data input = PermissionDataset-user1
- Action code = Submit&Closed
- Expected data output = PermissionDataset-user1
- Return = Create successful
Business dictionary

Test report

● A test report collects all information related to test of a scenario within a campaign execution
  ○ test date
  ○ actor achieving the test
  ○ execution time of the test
  ○ test status (ok, ko, n/a)

● The user can re-execute a same scenario several times within a campaign execution
  ○ each test scenario is then identified with help of a sequence number from 1 to unbounded
Business dictionary

Detailed test result
- A detailed test result collects detailed information about outcomes of a test step within a test scenario, namely for a test report as described previously
  - result
  - status (ok, ko, n/a)
  - bugs description
Business rules

Scenario
● A scenario can be reused by many campaign definitions

Step
● A step is defined for a scenario only

Campaign execution
● A campaign execution is based on the same scenarios and steps than its related campaign definition

This is the first set of business rules
Others will be defined later during the modeling process
Semantic data modeling

Representing in a formal way business concepts and business rules described above
To fully understand this UML semantic data model some key concepts must be known

- See next slides

Semantic Data Model
UML key concepts - Qualifier attribute

- Order is a qualifier attribute
- A Scenario uses zero to unbounded Steps in a well-identified ordering
- A Step is used by one Scenario only
UML key concepts - Ternary relationship

- Campaign execution, Test report and Scenario are linked with each other.

- It means that a Test report is related to a Scenario and a Campaign execution.
UML key concepts - Constraint on relationship

- (Same) = when creating a Campaign execution from a Campaign definition all Scenarios and Steps of this Campaign definition are reused to create the new Campaign definition
A Detailed test result exists in the context of one Step and one Test report only.
UML key concepts - Naming convention

- '<- includes' = a Campaign definition includes one to unbounded Scenarios
User Interface

How the semantic data model is used by a MDM system?
The MDM system follows the semantic data model through a mechanism of data hierarchy views.
Scenario without step

APIs
- TestDrivingV3
  - Actions

Scenario-steps
- Delete Real Estate Role for a Party - Manual - Data set - Real Estate - PreviousDB-DBDvhoan
- Party and Real estate modification - Manual - Data set - Real Estate - PreviousDB-DBDvhoan
- Party, Real estate creation - Manual - Data set - Real Estate - PreviousDB-DBDvhoan
- [Scenario unset]
Scenario with steps

Delete Real Estate Role for a Party - Manual - Data set - Real Estate - PreviousDB-DBDvhoan
- Step 0: Read party
- Step 1: Read real estate of the party
- Step 2: Select a real estate
- Step 3: Delete role of the party of the real estate

Party and Real estate modification - Manual - Data set - Real Estate - PreviousDB-DBDvhoan
- Step 0: Read party
- Step 1: Read real estate of the party
- Step 2: Select a real estate of the party
- Step 3: Modify role of the party for the real estate

Party, Real estate creation - Manual - Data set - Real Estate - PreviousDB-DBDvhoan
- Step 0: Create permission for user
- Step 1: Login by user
- Step 2: Create a party
- Step 3: Create a real estate
- Step 4: Attach a party has a owner to real estate
- [ Scenario unset ]
From semantic data model to logical data model

- Identifier
- Foreign key
- Labeling
- Formal naming
Logical data model

- Data Models
  - Data Model Configuration
  - Data Types
  - Data Structure

- APIs
  - Actions
  - Publish

- Services
  - Business Objects and Rules
  - AJAX Components
  - Java Bindings

Data structure

- root
  - TestDriving
    - TestConfiguration
      - ReferenceData
      - CampaignDefinition
      - CampaignDefinitionIncludesOrderScenario
    - Scenario
      - oid (Integer)
      - objective (Text)
      - FKTTestFixture (Test type)
      - FKTTestCategory (Test category)
      - FKInitialDatabase (Initial Database)
      - FKDaptpattern (Data pattern)
    - Step
      - oid (Integer)
      - objective (Text)
    - ScenarioUsesOrderStep
      - FKSnano (Scenario)
      - FKStep (Step)
    - FKStep (Step)
  - CampaignExecution

Orchestra Networks

YOUR DATA IN PERFECT HARMONY
Data hierarchy configuration (1/2)

● The MDM system follows the semantic data model to display all possible data hierarchies from a leaf table (aka node)
  ○ here the Step table plays the leaf role

● Then the user selects the group Scenario (aka dimension) to obtain a hierarchy view from Steps by Scenario
The MDM follows the semantic data model by suggesting the use of the order qualifier attribute to enforce an ordering of the Steps within a Scenario.

Moreover, the labeling of each data hierarchy level is customized.
When authoring a campaign definition the MDM displays its scenarios automatically.
Campaign definition and its Scenarios (2/3)

Join table to manage the many-to-many relationship between Campaign definition and Scenario

User navigation link computed automatically by the MDM system
Attachment of a scenario to a campaign definition (1/2)

- Data hierarchy with
  - leaf (aka node) = Scenario
  - group (aka dimension) = Campaign definition

- Three scenarios are not yet attached to a campaign definition
  - The popup menu allows the user to manage nodes and then attach them to the right group, namely a Campaign definition
The user has attached two scenarios to the Campaign definition 'Regular campaign'.

Since the data hierarchy is ordering each Scenario can be moved to a right position in the Campaign definition (0, 1, 2, etc.)
A campaign definition and its scenarios

Now the consultation of the Campaign definition 'Regular campaign' shows the two scenarios that have been attached previously.

- Now the consultation of the Campaign definition 'Regular campaign' shows the two scenarios that have been attached previously.
Starting context: no campaign execution

- This screen shows the initial situation without any Campaign execution defined in the MDM system
Creation of a campaign execution from a campaign definition
Campaign execution with two validation rules in failure

The user enforces the correction
Campaign execution with its scenarios

- Data hierarchy with
  - leaf (aka node) = Scenario
  - group (aka dimension) = Campaign execution

- This screen shows the two scenarios that have been attached to the campaign execution automatically
Test report and Detailed test result have been populated automatically when creating the new campaign execution.
Automatic feeding of Test report and data authoring

- Data hierarchy with
  - leaf (aka node) = Test report
  - group (aka dimension) = Campaign execution
Automatic feeding of Detailed test report and data authoring (1/2)

- Data hierarchy with
  - **leaf (aka node)** = Detailed test result
  - **group (aka dimension)** = Campaign execution
Automatic feeding of Detailed test report and data authoring (2/2)

- Detailed test results values after data authoring
New execution of an existing test report - 'runId' management (1/2)

Create new runId

After modification

TestReport
-runId
-testRunDate
-testedBy
-executionTime
-status
The service 'Create new runid' allows the user to create a new scenario execution within a same campaign execution
  ○ (Scenario 0, runId=0) and (Scenario 0, runId=1) for the same Campaign execution named 'Regular campaign'
Campaign execution with test reports and detailed test results

Data hierarchy with
- leaf (aka node) = Detailed test result
- sub group 1 (aka dimension 1) = Campaign execution
- sub group 2 (aka dimension 2) = Test report
Creation of new campaign execution from the same definition (1/2)
Creation of new campaign execution from the same definition (2/2)
New business rules

Enrichment of the initial data modeling
When a campaign definition is updated

We want no impact on existing campaign execution

- **Issue #1: what happens if a scenario is deleted?**
  - current situation: all records in Test report with a FK to this scenario will be in failure
  - target=> two ways for deleting a scenario are offered to the user
  - deleting a scenario must be encapsulated by a java service checking that there is no campaign execution for this scenario.
  - Reading Test report table. If at least one campaign execution exist for this scenario then the deletion is not permitted
  - a new service ‘Inactivate scenario’ allows the user to remove logically a scenario so that when a new campaign execution is created this scenario is no longer used. A service ‘Activate scenario’ allows for rendering normal the scenario. A new boolean field ‘active’ in the Scenario table is added to tackle this need
When a campaign definition is updated

We want no impact on existing campaign execution

- Issue #2: what happens if a step is deleted?
  - current situation: all records in Detailed test result with a FK to this step will be in failure
  - target=> two ways for deleting a step are offered to the user
    - a step can be deleted within a scenario only if this scenario is not used in a campaign execution. This checking is the same as one described for the scenario deletion. A java service must be developed to meet this need
    - a new service ‘Inactivate step’ allows a same user procedure than for the scenario but applied to a step. A new boolean field ‘active’ in the Step table is added
When a campaign definition is updated

We want no impact on existing campaign execution

- Issue #3: what happens if a scenario or a step is added?
  - Test report and Detailed test result are not impacted => no impact on existing campaign execution

- Issue #4: what happens if a step or a scenario is updated
  - Test report and Detailed test result are not impacted but existing campaign execution will automatically inherit from these modification
Creating a new Campaign definition based on an existing one

- Issue #5: what happens when creating a new campaign B from an existing one A
  - Creation of a new record in Campaign definition for B
  - CampaignDefinitionIncludesOrderScenario of A are used to create a new set of records in campaignDefinitionIncludesOrderScenario for B - And all scenarios and their steps attached to A are recreated as new ones and attached to the campaign B
What can a user do on a Campaign Execution

- Issue #6: adding or removing a record in Test report table
  - no because it will be considered as a modification of the campaign definition

- Issue #7: adding or removing a record in Detailed test result table
  - adding => yes the user add a step or reuse an existing one but he must manually create the record in Detailed test result
  - removing => no. the user can already set as n/a the status
<table>
<thead>
<tr>
<th>Service name</th>
<th>Functional description</th>
</tr>
</thead>
</table>
| **Create new campaign execution from the underlying campaign definition**   | o/ Campaign Definition table: -> on single record  
    o/ TODO:  
    ✓ create a new Campaign Execution over defined Campaign definition  
    1. Populate Scenarios to Test report table  
    2. Populate Steps to Detailed test result table |
| **Create new runId**                                                         | o/ Test report table: -> on single record  
    o/ TODO:  
    ✓ Create new Test report record with new RunId but remains information:  
    - Campaign execution  
    - Scenario  
    - Order  
    - Server configuration (if defined)  
    - Client configuration (if defined)  
    ✓ Create corresponding Detailed test results |
| **All steps OK**                                                             | o/ Detailed test result table:  
    - on view [Test report with detailed test result by campaign execution] (i.e. Detailed test result by II Test report by I Campaign execution)  
    - at Test report level  
    o/ TODO:  
    ✓ Set all steps status within a test case (Scenario) = OK  
    ✓ Set Scenario status = OK |
| **Scenario status calculation**                                             | o/ Detailed test result table:  
    - on view [Test report with detailed test result by campaign execution] (i.e. Detailed test result by II Test report by I Campaign execution)  
    - at Test report level  
    o/ TODO:  
    ✓ Calculate Scenario status based on containing steps status :  
    All steps = OK -> Scenario = OK  
    At least one step = KO -> Scenario = KO  
    One step = N/A and all other step = OK -> Scenario = N/A |
| **Duplicate Campaign definition**                                           | o/ Campaign definition table  
    o/ TODO:  
    ✓ Duplicate Campaign definition record within table  
    ✓ Populate including Scenarios to join table CampaignDefinition-scenarios. |

Catalog of services (business rules) the MDM system must enforce to ensure the integrity and quality of data.
Services are located within the semantic data model as operations

- The table owner of an operation is not decided from a pure object-oriented approach. It depends on from where the operation will be launched at the UI level
  - **Use of a dedicated stereotype if you want to avoid any confusion with usual object modeling**

- Eg. `createNewCampaignExecution()` is located in the Campaign definition class
Lesson learnt

Findings
Lesson learnt

- Semantic data modeling is a formal way of representing data business concepts and a first set of business rules.

- A second level of business rules requires a formal specification of services that are located as operations within the semantic data modeling.

- A derivation of the semantic data modeling into the logical data modeling allows to render executable the model by a MDM system and then prevent for useless hard-coding software development.

- The MDM system must follow the data model to ensure the computation of all possible data hierarchy views and the modeler must configure the suitable ones for users depending on their needs and profiles.
In this presentation we didn't tackle these topics

● Some data business concepts must be modeled through a business state life-cycle. Therefore the semantic data modeling is not only a static point of view on data but also a dynamic representation.

● Eg. A Campaign execution might have such state values over time
   ○ Not started, started, suspended, cancelled, archived
   ○ And depending on these states some operations on the Campaign execution are permitted or not

● Organizational issues
   ○ Data approval workflow (activity diagram)
   ○ Permission management (master data)
To get further information

Please attend the fourth part of this MDM introduction training course

This fourth part is dedicated to the enforcement of an Enterprise Data Architecture
Stay tuned

pierre.bonnet@orchestranetworks.com
skype: pierre.orchestra

www.orchestranetworks.com