Behaviour Driven TP Design

ebrifol

Addressing the flaws in the current design process, this proposal allows a single point of design for TPs and enables a single flow. Once the TP model is defined, the automated build and test process should run with no manual intervention.

# Current Scenario

The current TP design process is initiated by feeding two versions of a node model through the MOM Tool. The MOM Tool will produce a delta document that is then taken by a designer and applied to the existing TP model. Manual modifications are made to some TPs currently delivered in ENIQ-S but the way these modifications are designed requires the TP to be created first. This creates a potentially endless loop where designers create and test the TP to discover if modifications are needed and/or working as expected.

# Proposed Approach

The focus of TP design is to define the table schema and then the required loader and aggregations are automatically generated. The subsequent set actions to trigger the aggregations, loader and parser are also automatically generated. Manual modifications to TPs are all performed on the sets to alter the functionality being delivered. Using a behaviour driven design approach to TPs would remove the need for manual modifications as they exist today.

When a node model is received, it defines all the available counters that could be made available to ENIQ-S. The requirement from the designer would then be to define what action we wish to take on those counters i.e. load them. By defining what counters will be loaded, a table schema can be automatically generated. By defining the action and automating the target schema, we can remove the need for manual modifications while still producing the same output TP.

Using the node model in conjunction with a model to describe the actions of the TP, we could identify what actions need to be amended to support the latest node version. It is expected that in most cases, these changes to the actions could be automated also.

# Advantages

1. A single design point for TPs as opposed to multiple tools for difference stages
2. Production of a single model to describe a TP
3. Fully automatable design and test flow with no manual intervention required.
4. Addresses the manual modifications issue
5. Enables introduction of model driven TP design

# Disadvantages

1. Significant redesign of sets generation code
2. Fundamental change to the design approach and thinking around TPs today