Methods for Reliable Weight Tracking and Reporting with Configuration Changes and Multiple Similar Vehicles
Introduction

• Doing weight tracking for series production of vehicles with configuration changes raises some challenges
  – How can we efficiently store the weight items
  – How can we efficiently deploy a weight item to a group of vehicles
Storing of weight items

• Less desirable solutions:
  – Each vehicle has its own project DB and copy of all items
    • Can create huge amounts of data
    • Update of items is impractical
  – Each vehicle has its own project DB, but references weight items from a shared item library
    • Can still create huge amounts of data
    • Difficult to know which quantities and properties to share in the library
    • Difficult to deploy efficiently

• Suggested solution:
  – Reuse weight items by including all items in the same DB and tag/deploy them to the vehicles
Tagging and Deploying

• Less desirable solutions:
  – Tagging of items not belonging to the baseline
    • Will work well if you have individual changes only
  – Parent/Child hierarchy
    • Referencing the baseline and other parent vehicles in a hierarchy
    • Reuse of items limited to the hierarchy
  – Individual Flags
    • Works OK if number of vehicles are limited (one column needed for every flag/vehicle
    • Deployment not efficient

• Suggested solution:
  – Phase coding
Phase Coding

• Phase coding means looking at the series production as a timeline, where:
  – The timeline is starting with vehicle 1 and ending at vehicle n.
  – You define the necessary phases, ranging from any start and endpoint on the timeline
  – Weight items are tagged to a phase, meaning that they are to be included for vehicles in this phase
  – The total weight of an individual vehicle is the sum of all items tagged to the phases that the ”time” of the vehicle corresponds to

General Example

- Allowing to set "Time In" and "Time Out"
  - Defining the "time" an item is to be included
- To be used when "Flags" are not sufficient
- To be combined with Global Filter

<table>
<thead>
<tr>
<th>Code</th>
<th>Time In</th>
<th>Time Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase AC</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>Phase AZ</td>
<td>20</td>
<td>80</td>
</tr>
<tr>
<td>Phase CX</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Phase CG</td>
<td>30</td>
<td>100</td>
</tr>
<tr>
<td>Phase AZ</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Phase SZ</td>
<td>70</td>
<td>100</td>
</tr>
</tbody>
</table>
Example: Vessels

• Series of 10 hulls with modifications
Example: Aircrafts

- Aircraft series production
  - Various configurations during production

Filter: Aircraft no 73
Baseline + Mod. 2, 3 & 4
Example: Offshore Construction

• Lifecycle phases of an offshore construction
Example: Satelite

- Satelite "phases"
Advantages of the Phase Code Approach

• Items are only needed once (one row) in the database regardless of how many vehicles are in the production line

• It is easy to deploy weight items to a range of vehicles, and to change the extent of the range

• You can easily find the weight of an individual vehicle by slicing through the phases at the time of the vehicle

• Same approach can be used for multiple phases of a single vehicle throughout its life-cycle
Demonstration of Phase Codes in ShipWeight
Questions
Thank you