Enabling Design Reuse for a Complete Digital Thread

Sean Callahan, PhD, TF
Kevin Puterbaugh

RROI #17-00630
Biographies

Global Product Data Interoperability Summit | 2017

• **Sean Callahan** is a Technical Fellow working for The Boeing Company. He has received a PhD in physics from the California Institute of Technology and an MBA from Seattle University. He has been at Boeing since 1988 and his research focuses on developing product information models to support design automation, knowledge-based engineering and computer aided engineering. He developed and patented Extended Generic Product Structure (EGPS) technology which solves the problem of sharing detailed design content between configurations (versions, variants and branches) of hierarchical, generalized product structures. EGPS has been implemented in a Boeing internal application that compares all design content differences between configurations of CATIA models of parts, assemblies and installations. His current focus is on applying this technology across abstractions levels of product design from requirements to functions to logical system designs to various physical levels of design representation. He has previously focused on high performance visualization, solid modeling, and rule-based, semi-automated design. Sean was the design data architect for a Boeing-developed system design tool being used on the 787 program, based on Siemens PLM’s product SLATE, to capture a computer sensible representation of the system design. Sean has served as an associate editor for the Journal of Computer Science in Engineering (JCISE) in informatics and has published the results of his research in several publications.

• **Kevin Puterbaugh** is a research scientist working for The Boeing Company. He received a BS in Computer Science from Michigan State University. He has been at Boeing since 1986. His work focuses on large-scale visualization, product representation and design automation tools. He is the co-architect and lead developer for an advanced system that analyses fine-grain changes between general, hierarchical structures. He has implemented specific solutions of this tool for Autodesk Inventor, CATIA V5 and various EBOM structures. His work has resulted in several patents and journal publications.
Main Point

Global Product Data Interoperability Summit | 2017

- Major gaps in PLM data-management:
  - Fine grain design-reuse / change-analysis
  - Design change propagation
  - Sharing and remembering are mutually exclusive

- These gaps disables supporting digital twin and digital thread visions across lifecycle.

- This talk will present, Extended Generic Product Structure (EGPS) a patented, validated solution to these gaps.
• Part 1 – Business Case and High-level Technical Descriptions
Boeing’s Operational Foundation

Boeing depends on change analysis across business operations:
- Certification is net-change based
- Design review
- Manufacturing planning updates
- Part and material ordering
- Subassembly supplier cost adjustments (72 step process to determine change!)
- ...

**NOTE:** At companies that design and manufacture complex product families, workers can spend between 20-80% of their time understanding design change, before they can do their real job.

Key associated pain point and PLM gap:
- COTS PLM systems’ design reuse capabilities are insufficient to automate change understanding, **leading to manual labor, extra flow time and rework.**
The digital landscape

NOTE: There is no time in this picture

Requirements

Functional Design
Logical System Design
Physical EBOM/CAD
MBOM
Mnfturing Assembly Sequence
Physical Kits/Parts

Design Domains
Accountability Mappings
Every level of design has a, potentially, full definition of this structure.
Abstracting the product enables design content sharing.

All instructions (text and geometric) are auto-updated with explicit call-out of change content.

Minimal in-context instructions not requiring extensive research – “full kit”!!!
Abstracting the product enables design content sharing.

Variation of a product structure implies a logical structure

Information Architecture Concepts:
1. Family
2. Stable Usage
The logical structure can be constructed from configs

Information Architecture Concepts:
1. Family
2. Stable Usage
# Twelve-brick Variant Comparison

## Global Product Data Interoperability Summit | 2017

### EBOM-CATIA: Comparing 2 versions of Module "multibrick"

<table>
<thead>
<tr>
<th>Status</th>
<th>Logical Structure</th>
<th>Reference Definition</th>
<th>Variant: 12_1</th>
<th>Variant: 12_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mod</td>
<td>multibrick</td>
<td></td>
<td>12_1</td>
<td>12_2</td>
</tr>
<tr>
<td>Eqv</td>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mod</td>
<td>Connections</td>
<td></td>
<td>12_1</td>
<td>12_2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ignore</td>
<td>Document</td>
<td></td>
<td>12_1</td>
<td>12_2</td>
</tr>
<tr>
<td></td>
<td>Right</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Diagram

The diagram shows a visual comparison of two versions of the "multibrick" module, highlighting differences in properties such as lengths, offsets, and connections. The visual representation complements the table data, providing a clear comparison of the variant differences.
## Twelve-brick Variant Comparison

### Global Product Data Interoperability Summit | 2017

#### EBOM-CATIA: Comparing 2 versions of Module “multibrick”

<table>
<thead>
<tr>
<th>Status</th>
<th>Logical Structure</th>
<th>Reference Definition</th>
<th>Variant: 12_1</th>
<th>Variant: 12_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mod</td>
<td>multibrick</td>
<td>multibrick</td>
<td>12_1</td>
<td>12_2</td>
</tr>
<tr>
<td>Mod</td>
<td>Connections</td>
<td></td>
<td>12_1</td>
<td>12_2</td>
</tr>
<tr>
<td>Mod</td>
<td>Fix.2</td>
<td></td>
<td>12_1</td>
<td>12_2</td>
</tr>
<tr>
<td></td>
<td>Offset.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mod:R</td>
<td>Left</td>
<td>sbbrick</td>
<td>12_1 : 6BNoHoles</td>
<td>12_2 : 6B1H</td>
</tr>
<tr>
<td>Mod:R</td>
<td>Connections</td>
<td></td>
<td>^ : 6BNoHoles</td>
<td>^ : 6B1H</td>
</tr>
<tr>
<td>Mod:R</td>
<td>Middle</td>
<td>twobrick</td>
<td>6BNoHoles : 2BNH</td>
<td>6B1H : 2B</td>
</tr>
<tr>
<td>Mod:R</td>
<td>Connections</td>
<td></td>
<td>^ : 2BNH</td>
<td>^ : 2B</td>
</tr>
<tr>
<td>Mod:R</td>
<td>Left</td>
<td>onebrick</td>
<td>2B1H : NH</td>
<td>2B1H : H</td>
</tr>
<tr>
<td></td>
<td>ExtraBrick1</td>
<td>onebrick</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OneBrick_NH.1</td>
<td>onebrick</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Twelve-brick Variant Comparison

**Global Product Data Interoperability Summit | 2017**

---

#### EBOM-CATIA: Comparing 2 versions of Module "multibrick"

<table>
<thead>
<tr>
<th>Status</th>
<th>Logical Structure</th>
<th>Reference Definition</th>
<th>Variant: 12_1</th>
<th>Variant: 12_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mod</td>
<td>multibrick</td>
<td></td>
<td>12_1</td>
<td>12_2</td>
</tr>
<tr>
<td>Mod</td>
<td>Connections</td>
<td></td>
<td>12_1</td>
<td>12_2</td>
</tr>
<tr>
<td>Mod</td>
<td>Fix.2</td>
<td></td>
<td>12_2</td>
<td>12_2</td>
</tr>
<tr>
<td>Mod</td>
<td>Offset.6</td>
<td></td>
<td>12_1</td>
<td>12_2</td>
</tr>
<tr>
<td>Mod:R</td>
<td>Left</td>
<td>skbrick</td>
<td>12_1 : 6BNoHoles</td>
<td>12_2 : 6B1H</td>
</tr>
<tr>
<td>Mod:R</td>
<td>Connections</td>
<td></td>
<td>6BNoHoles</td>
<td>6B1H</td>
</tr>
<tr>
<td>Mod:R</td>
<td>Middle</td>
<td>twobrick</td>
<td>6BNoHoles : 2B1H</td>
<td>6B1H : 2B1H</td>
</tr>
<tr>
<td>Mod:R</td>
<td>Center</td>
<td></td>
<td>6BNoHoles : 2B1H</td>
<td>6B1H : 2B1H</td>
</tr>
<tr>
<td>Mod:R</td>
<td>Left</td>
<td>Reports/Exports</td>
<td>Delta Report</td>
<td>Output Instance Map Report</td>
</tr>
<tr>
<td>Mod:R</td>
<td>ExtraBrick</td>
<td></td>
<td>Info</td>
<td>Reframe On</td>
</tr>
<tr>
<td>Mod:R</td>
<td>OneBrick</td>
<td></td>
<td>Expand all children</td>
<td>Highlight</td>
</tr>
</tbody>
</table>

---

---

---

---
Twelve-brick Variant Comparison
Twelve-brick Variant Comparison
Change Analysis Assembly Difference View

Actual change
ChangeAnalysis Assembly Difference View

Turn off Same and Equivalent in browser
Zooming and Highlighting in Geometric Comparison

Highlight two common components in Rev-
Summary of Drill Down Process
Conclusion

COTS tools do not support the kind of precise-configuration design-sharing presented here. They can either have “max-config” sharing or duplicated precise configurations. Extended Generic Product Structure offers a proven, implementable solution that enables design content sharing across the digital thread. Two implementation paths:

- Full object-oriented or graph database that literally shares common content or
- Using duplicated, precise configurations, but capturing “bread crumbs” enabling shared representation reconstruction.
It is possible to derive complete shared representation is sufficient information is captured and managed at authoring time, in an duplicated representation.
• Part 2 – Technical Product Structure and Data Model Level Description.
2005 Extended Generic Product Structure: SV

Single Variant Product Structure Pattern

C1, 2, 3 - Composition

Part is a type (subclass) of PSE

2006 Publication: Extended Generic Product Structure: An Information Model for Representing Product Families
EGPS Patents and Applications Relevant to this Presentation


Publication covering 3-5, above:
2005 Extended Generic Product Structure: MV

Traditional Master Version Model

Single Variant Product Structure Pattern

Original EGPS Multivariant Product Structure Pattern
2005 Extended Generic Product Structure: MV

Legend:
- Reusable PSE
- In-place PSE
- Reusable PSE Version
- In-place PSE Version

Occurrence:
- Usage
- Usage Version
- Occurrence Version

Component Link
Instance Link
Component Configuration Link

Usage:
- Reusable PSE Version
- In-place PSE Version
- Usage Version
- Occurrence Version

Reusable PSE
In-place PSE
In-place PSE Version
Reusable PSE Version

Component Configuration Link

Product Structure Element:
- 6B-A
- 2B-A
- 1B-NH
- 1B-H
- 1B-NH
- 1B-H
- 2B-A
- 2B-B
- Top Brick
- Middle: TwoBrick
- Lower: TwoBrick
- SixBrick
- TwoBrick
- Brick

Occurrence:
- Usage
- Usage Version
- Occurrence Version

Component Link
Instance Link
Component Configuration Link

Usage:
- Reusable PSE Version
- In-place PSE Version
- Usage Version
- Occurrence Version

Reusable PSE
In-place PSE
In-place PSE Version
Reusable PSE Version

Component Configuration Link

Usage:
- Reusable PSE Version
- In-place PSE Version
- Usage Version
- Occurrence Version

Reusable PSE
In-place PSE
In-place PSE Version
Reusable PSE Version

Component Configuration Link

Usage:
- Reusable PSE Version
- In-place PSE Version
- Usage Version
- Occurrence Version

Reusable PSE
In-place PSE
In-place PSE Version
Reusable PSE Version

Component Configuration Link

Usage:
- Reusable PSE Version
- In-place PSE Version
- Usage Version
- Occurrence Version

Reusable PSE
In-place PSE
In-place PSE Version
Reusable PSE Version

Component Configuration Link
2006 Extended Generic Product Structure: MV

Global Product Data Interoperability Summit | 2017
2015 EGPS: Generalize Family Pattern

From

PSE (Master) 1..n Version

Traditional Master Version Model

To

Element 1..n Variant 1..n BranchInstance 1..n Version

Single Element Variant-Branch-Version Pattern

+arch

+inst 0..n

+ref

+context

+config

+prev

+next

0..1

0-1

0..1

+from

+to

Immutable Timestamp

Prev

Merge

Prev
2015 EGPS: Variant, Branch, Version

Extended Generic Product Structure (EGPS) Pattern
• COTS tools do not support the kind of precise-configuration design-sharing presented here.
• They can either have “max-config” sharing or duplicated precise configurations.
• Extended Generic Product Structure offers a proven, implementable solution that enables design content sharing across the digital thread.
• Two implementation paths:
  • Full object-oriented or graph database that literally shares common content or
  • Using duplicated, precise configurations, but capturing “bread crumbs” enabling shared representation reconstruction.
It is possible to derive complete shared representation if sufficient information is captured and managed at authoring time, in an duplicated representation.