

System Architecture using SysML - Despite the GAPS!

John Herrold and Brittany
Friedland, The Boeing Co.

GLOBAL PRODUCT DATA INTEROPERABILITY **SUMMIT** 2016



ELYSIUM

Parker Aerospace

NORTHROP GRUMMAN

BOEING



Biography

Global Product Data Interoperability Summit | 2016

- **Brittany Friedland** spent the beginning of her career in the Oil and Gas Industry before making a career change to work in the Aerospace Industry. She currently works for Boeing developing and deploying Model Based Systems Engineering tools and processes with the Boeing Enterprise (Commercial and Defense).
- **John Herrold** is currently the System Architect for the Integrated Product Architecture enterprise systems engineering program that provides a service ready model based systems engineering solution (process, tool and training) for systems and design engineers. John has been a Boeing employee for 35 years and has worked mostly in the engineering analysis domain, supporting many of the Boeing Commercial and Military Airplane products. John is a designated Boeing Technical Lead Engineer and a member of the International Council on Systems Engineering (INCOSE). John has a BSEE from the University of Washington.

Abstract

Global Product Data Interoperability Summit | 2016

System Architecture using SysML - Despite the GAPS!

System Modeling Language (SysML) is the widely adopted standard throughout industry for system modeling. However, based on experience we discovered several gaps in SysML that needed to be address before authoring large scale system architecture models. Some of the most notable gaps was the inability to reuse redundant assemblies (occurrence modeling) and the absence of any change management controls. Customizations can be created to enhance the usability of SysML and automate many of these manual book keeping tasks. Our goal is to enlist industry support for changes to the current SysML specifications standard. This presentation will cover the specific required usability enhancements and a list of other SysML gaps. It will also disclose some of the workarounds we utilized while attempting to use SysML to create large scale system architecture models.

Agenda

Global Product Data Interoperability Summit | 2016

- **Model Based Systems Engineering at Boeing**
- **The Occurrence Model Method - Designing Large Scale Logical System Architectures**
- **Mind the Gap(s)!**
- **Conclusion**

Today's Goal: To inform about Boeing's experience with using SysML for Large Scale Architectures and to begin the conversation with industry peers to identify potential improvements and lobby for changes to the current SysML specifications standard.

Boeing at a Glance

Global Product Data Interoperability Summit | 2016

- **Customers and customer support in 150 countries**
 - Total revenue in 2012: \$81.7 billion
 - 70 percent of commercial airplane revenue from customers outside the United States
- **Manufacturing, service & technology partnerships with companies around the world**
 - Contracts with 22,000 suppliers and partners globally
- **Research, design & technology-development centers & programs in multiple countries**
- **More than 170,000 Boeing employees in 50 states and 70 countries**



(Robert Malone, et al, INCOSE IS 2016, Insights from Large Scale Model Based Systems Engineering at Boeing)

A Sample of Diverse Boeing Products

Global Product Data Interoperability Summit | 2016



6

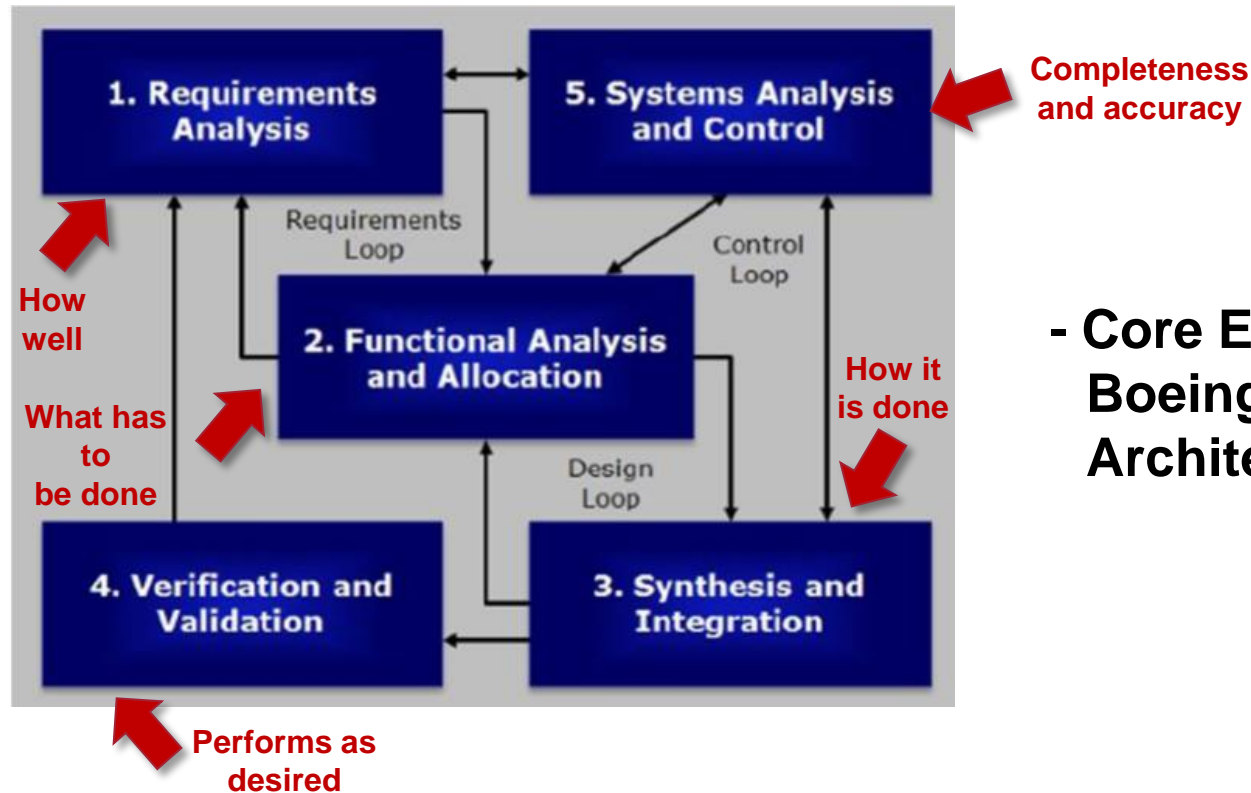
(Robert Malone, et al, INCOSE IS 2016, Insights from Large Scale Model Based Systems Engineering at Boeing)



Model Based Systems Engineering at Boeing

Model Based Systems Engineering at Boeing

Global Product Data Interoperability Summit | 2016



- Core Elements of Boeing Systems Architecture Models

(Robert Malone, et al, INCOSE IS 2016, Insights from Large Scale Model Based Systems Engineering at Boeing)

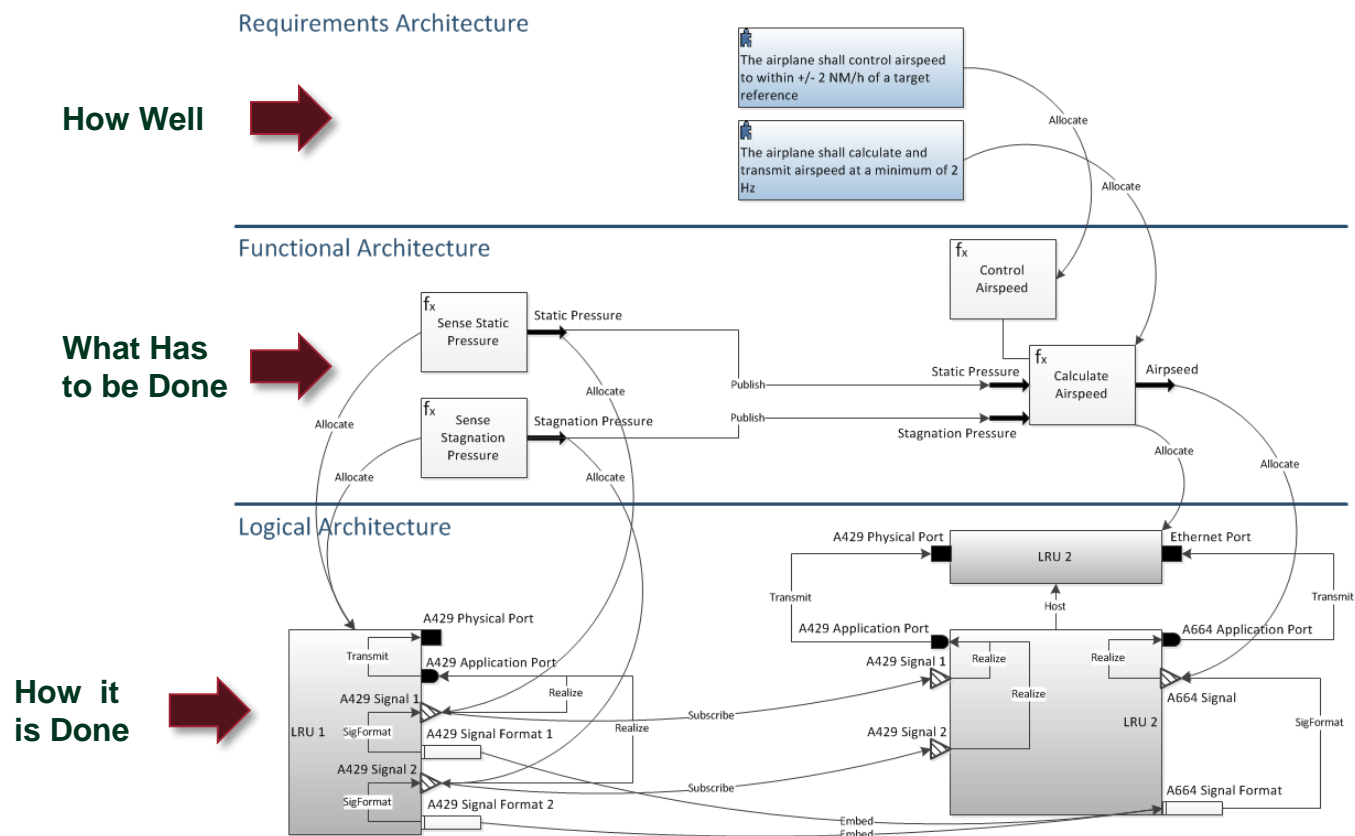
Boeing Successfully Uses Two Primary MBSE Methods

Global Product Data Interoperability Summit | 2016

- **Functions Based Systems Engineering Method (R-F-L-P)**
 - Various in-house developed technologies have introduced object-oriented concepts to this approach, including classes, objects and inheritance
 - Used for modeling and analyzing large intricate designs with multi-layered assemblies and associated detailed digital networks (black box/white box applications)
- **Object Oriented Systems Engineering Method**
 - Standard SysML application using domain specific developed custom profiles
 - Proven valuable for tying operational needs to system behavior
- **Investigating a blended approach for future applications**
 - Tool suite solution must be extendable to support both methods and integration between the methods
 - Understanding how much of SysML can be applied to large scale intricate designs – identifying the gaps

RFLP Example: A Simple Integrated System Architecture Model

Global Product Data Interoperability Summit | 2016



(Robert Malone, et al, INCOSE IS 2016, Insights from Large Scale Model Based Systems Engineering at Boeing)

Scaling it Up - Intricate Large Scale Integrated System Architecture within Boeing

Global Product Data Interoperability Summit | 2016

- **Features include:**
 - **Multi-leveled system assemblies (ensures adherence to integration and connectivity requirements)**
 - **Redundant system modeling (ensures adherence to failover safety requirements)**
 - **Detailed digital network interfaces (ensure adherence to networking functional, performance and design requirements)**
- **Boeing employs the Occurrence Modeling method to apply these features to Logical System Architecture Models**
 - **Investigation - Can SysML also support these features?**



The Occurrence Model Method - Designing Large Scale Logical System Architectures

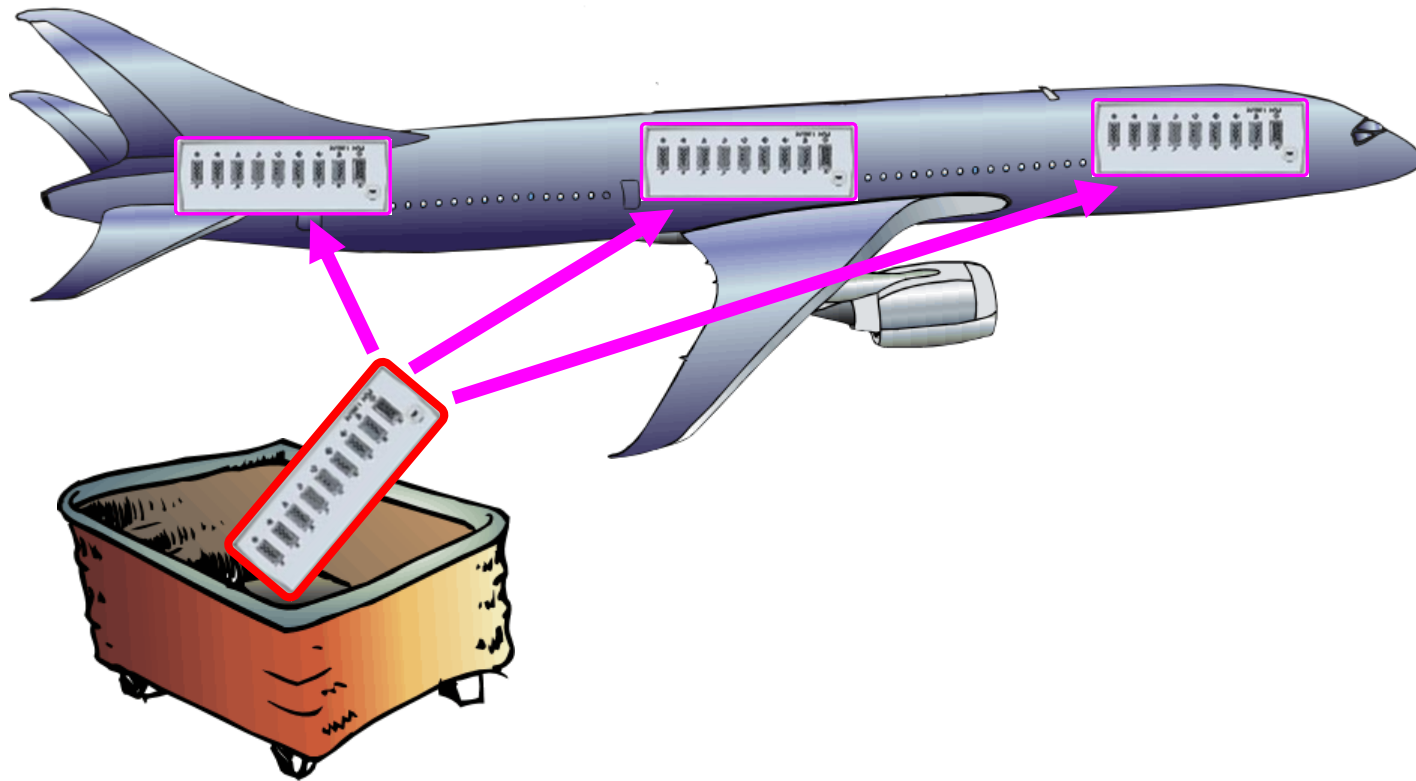
The Occurrence Model – An Overview

Global Product Data Interoperability Summit | 2016

- **Important Features**
 - Integrates a hierarchical product structure modeling perspective with a high fidelity interface control model (energy, data and physical medium transfers)
 - Provides scalability by enabling reuse of system components throughout the system design
 - However, within the context of where the system component is installed, allows for unique connections and specification of unique properties on both the system occurrence and child port occurrences
 - Enables automatic synchronization in the product structure of multiple installations of a single part definition each time the definition changes
 - Supports the design, configuration and analysis of IMA architectures
 - Supports the design of H/W with embedded S/W
 - Supports interface analysis activities that have the potential to identify multitudes of discrepancies, which were resolved prior to lab and flight testing
 - Provides downstream reuse of the verified product structure and detailed interface data by Safety, MMEL, Simulation, Labs, and Flight Test
 - Design supports product lines (variants) and large data sets

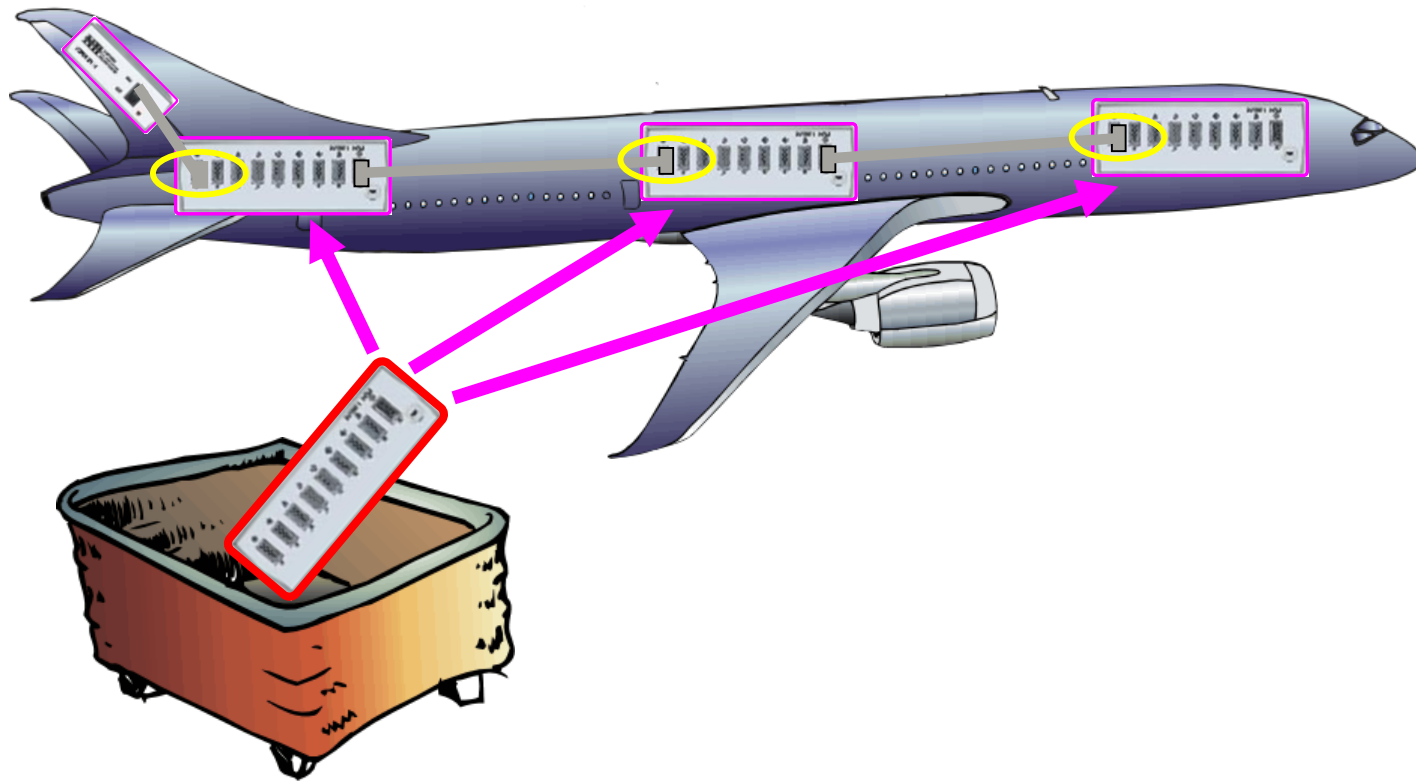
The Occurrence Model – An Overview

Global Product Data Interoperability Summit | 2016



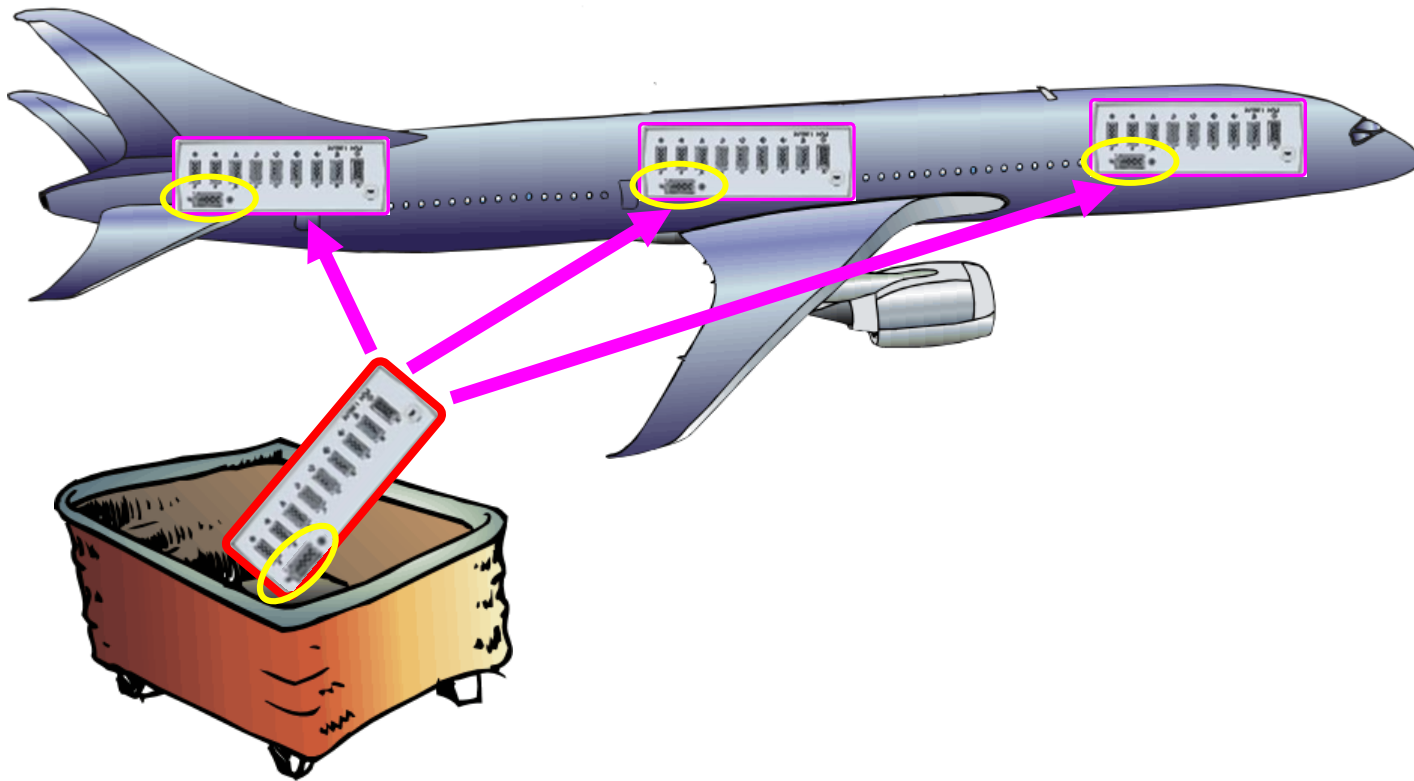
The Occurrence Model – An Overview

Global Product Data Interoperability Summit | 2016



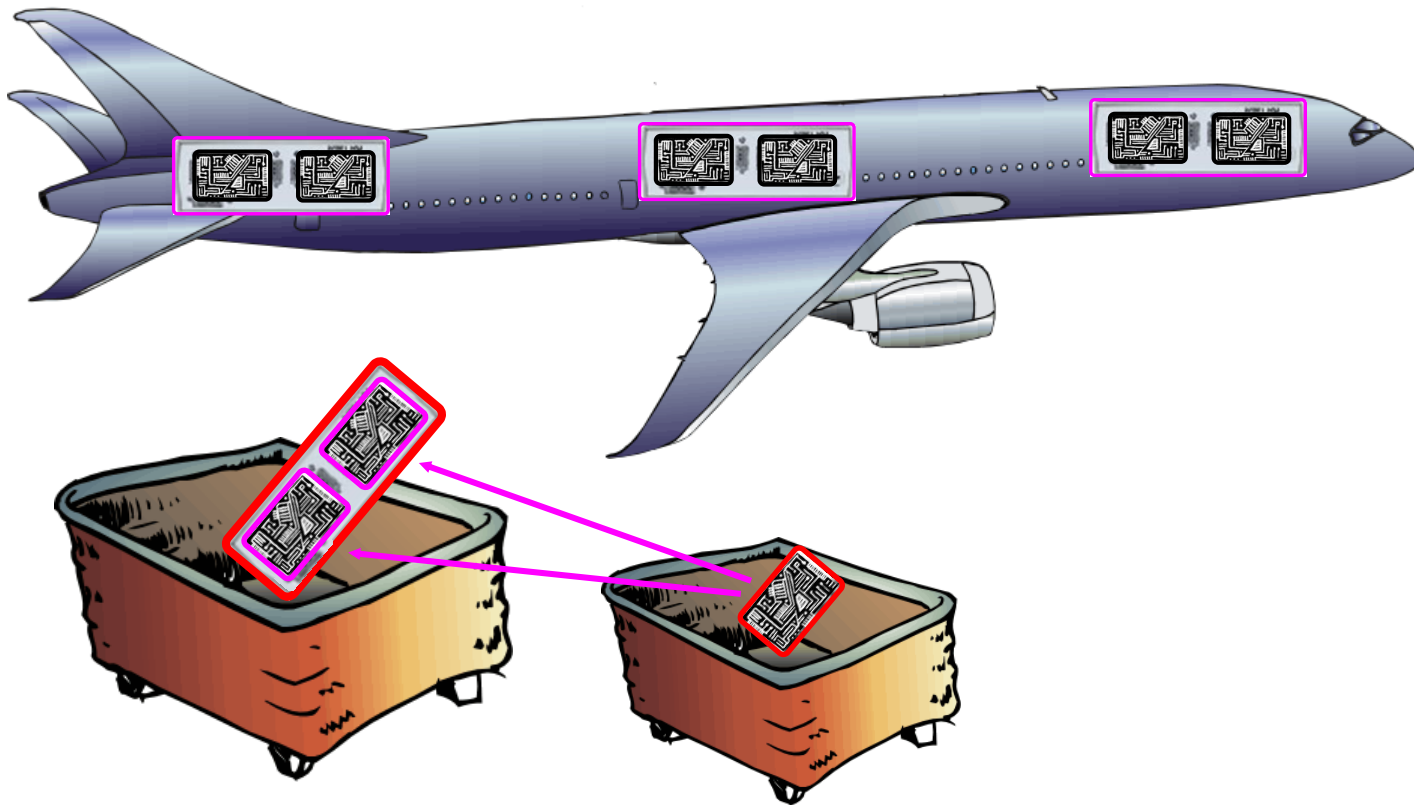
The Occurrence Model – An Overview

Global Product Data Interoperability Summit | 2016



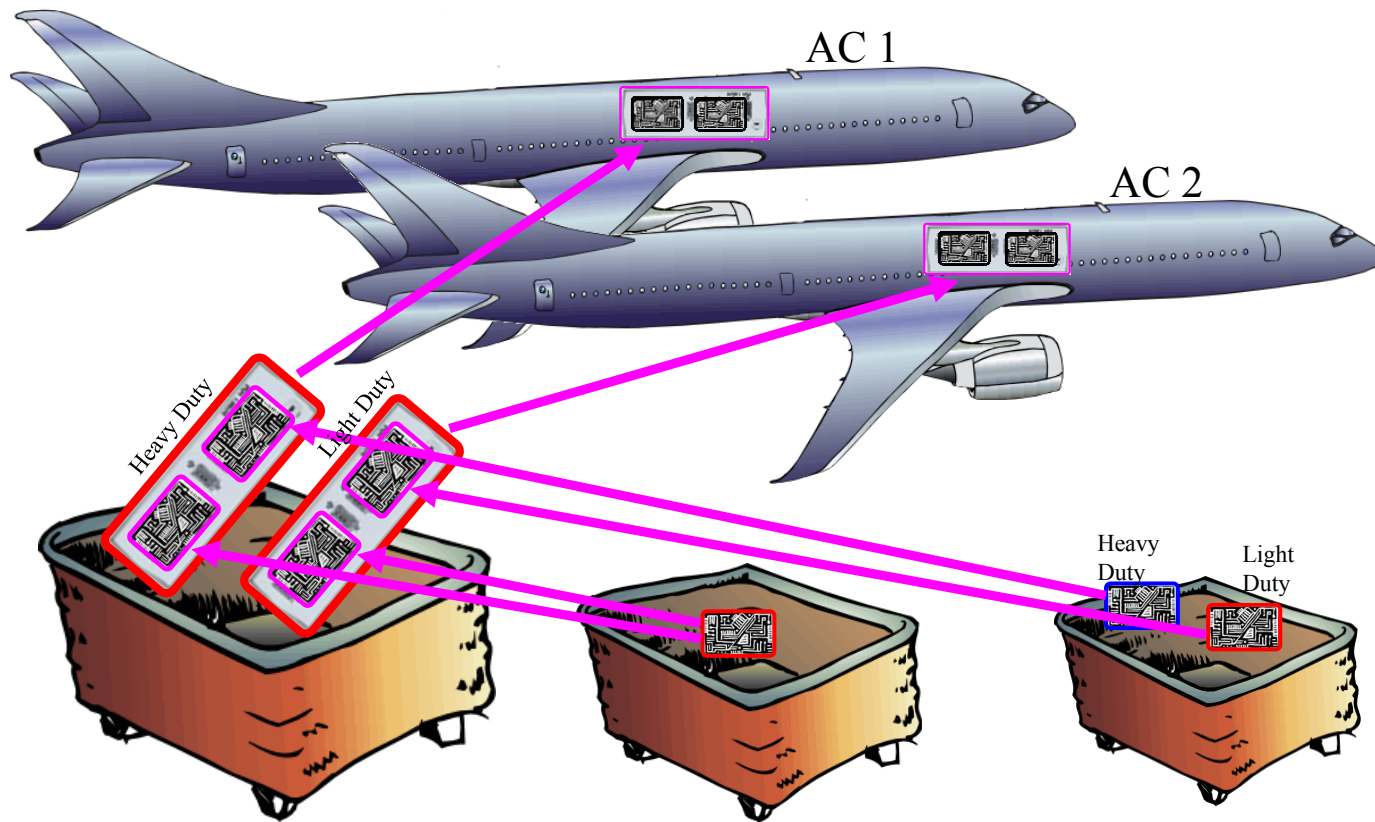
The Occurrence Model – An Overview

Global Product Data Interoperability Summit | 2016



The Occurrence Model – An Overview

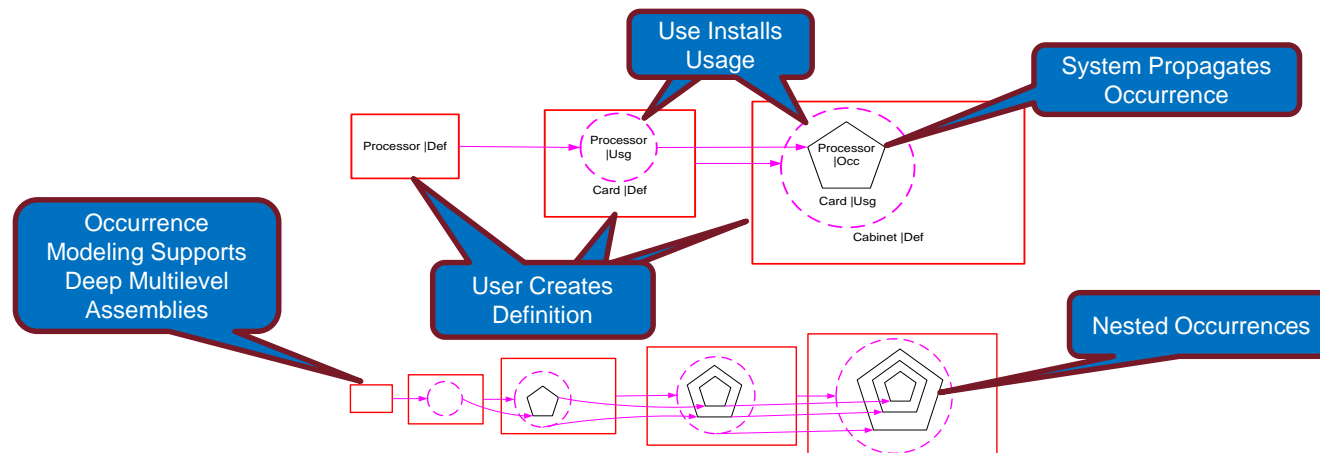
Global Product Data Interoperability Summit | 2016



The Occurrence Model – An Overview

Global Product Data Interoperability Summit | 2016

- In Summary:
 - A Definition represents a part in a library.
 - It contains information that will always be true for the part, regardless of installation.
 - A user creates a Usage by installing the part on another part Definition (such as the System of Interest).
 - The definition information propagates to the Usage, but the user may further define installation specific information on the Usage. (Ex: All Definitions of a car have 4 wheels, but one Usage might be painted red and the other blue.)
 - When a user installs a part on another part Definition, the system propagates its child Usages as Occurrences. (Ex: 4 wheels are Used on a Car, and 5 Cars are Used in a Fleet.)
 - This nesting may occur many levels deep, however, Occurrences always propagate to Occurrences. (Definition -> Usage -> Occurrence -> Occurrence -> ... -> Occurrence.)
 - The quantity of systems installed is traced by counting the leaves of the occurrence tree.





Applying SysML to Occurrence Modeling

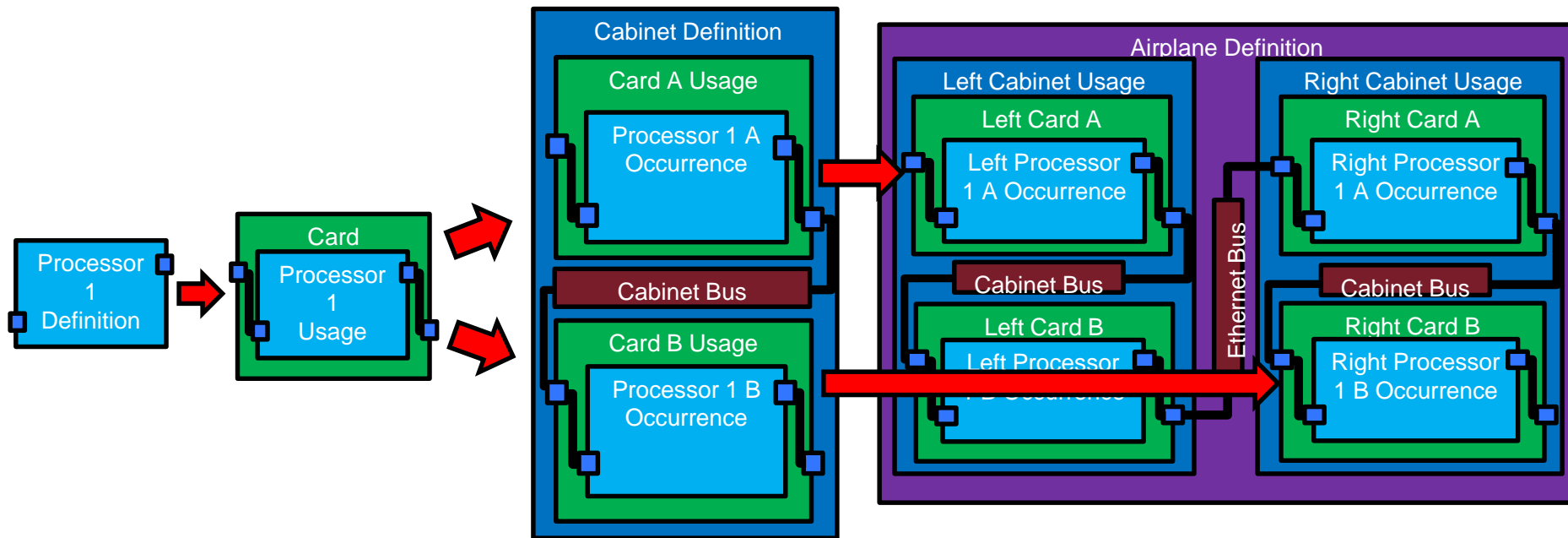
Motivation for Using SysML For Occurrence Modeling

Global Product Data Interoperability Summit | 2016

- **SysML is an industry standard modeling language that introduces syntax and semantic precision and consistency in a visual based approach**
- **Provides a common language for collaboration**
- **SysML is flexible and extensible to fit program's varying modeling needs**
- **Supported by a wide variety of SE modeling tools**

Using SysML For Occurrence Modeling : The Model Assembly Buildup Example

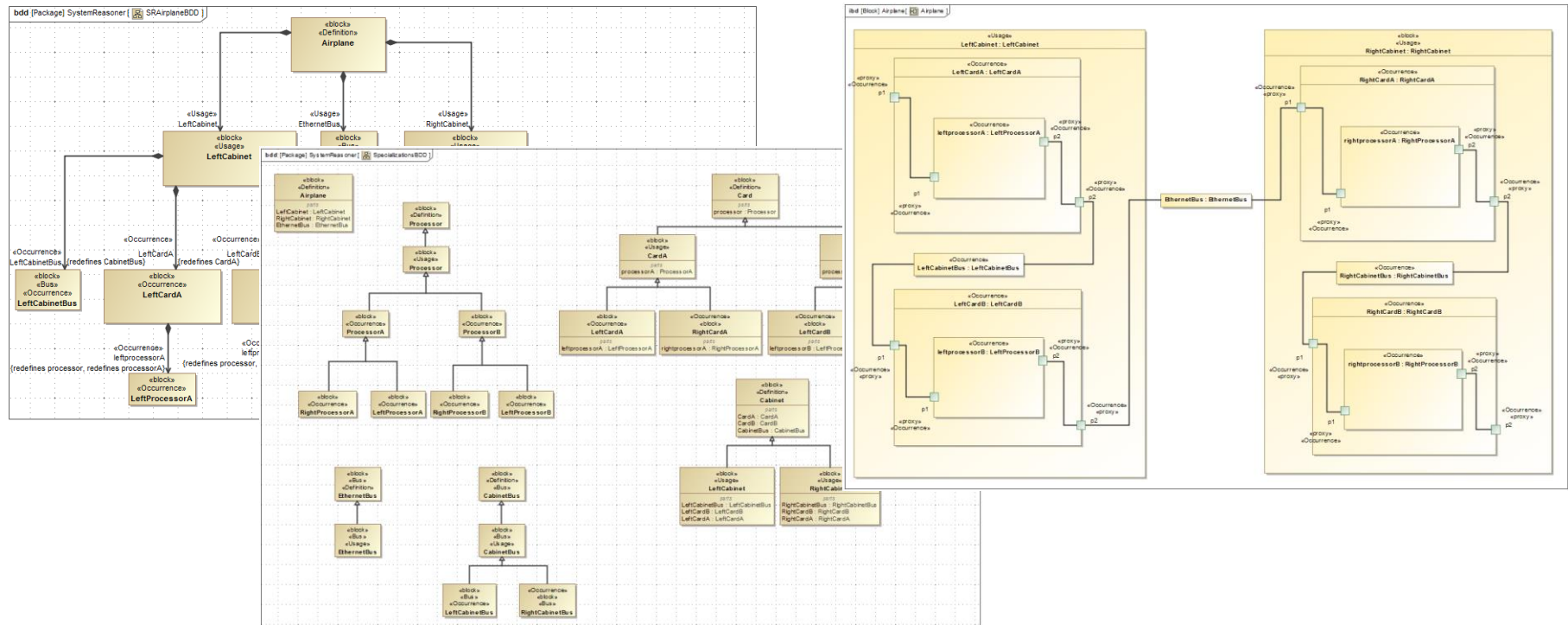
Global Product Data Interoperability Summit | 2016



Using SysML For Occurrence Modeling : The Model Assembly Buildup Example

Global Product Data Interoperability Summit | 2016

- Using SysML for occurrence model is possible, but the modeling is labor intensive and managing synchronization due to change is error prone.



Occurrences and SysML

Global Product Data Interoperability Summit | 2016

- **Summary of Findings**

- **In general, SysML doesn't model multi-level assemblies efficiently**
 - **Blocks (classes) need to be created for each element instance in a nested assembly (definition, usage, occurrence . . .)**
 - **Once these block are defined, their related SysML Parts (objects) can be used in IBDs to create unique port connections.**
 - **Parts in this situation are tightly coupled to their defining classes and represent proxy objects for the classes in the IBDs (1:1).**
 - **They are not reusable instances because they don't own their children ports, they only expose the defining class ports.**
 - **To create a new part instance that can accommodate unique connections, a related class instance with redefined ports must be created.**
 - **Ports need to be redefined at each instance level to allow for unique attributes and connections**
 - **Redefined ports/attributes are “disconnected” from their ancestor ports/attributes. Deletion of an ancestor port doesn't affect the redefined port.**
 - **Building the model and keeping elements (blocks/ports) synchronized is labor intensive without automation**



Mind the Gaps!

Mind the Gaps! Overview

Global Product Data Interoperability Summit | 2016

- **Through recent SysML modeling experiences, the following gaps have been identified**
 - **Lack of support for intricate system modeling including layered architectures – the Occurrence model**
 - **Usability: Overall complexity and difficulty in collaborating with non-SysML knowledgeable stakeholders**
 - **A need for robust handling of engineering units**
 - **Lack of support for Configuration Management (including Change Management, Product Line Engineering)**
 - **Lack of XML support for handling of SysML tables and diagrams**

Mind the Gaps! Candidate Changes to the SysML Standard

Global Product Data Interoperability Summit | 2016

1. Occurrence Model - Support for nested assemblies

- SysML needs to support deep nested assemblies. SysML only partially supports two levels Definition (SysML Block) and Usage (SysML Part) and there is no SysML equivalence to occurrences. SysML needs to be modified to create one or more part property “occurrences” that has knowledge of its parental context within an assembly.

Example: Define a processor, use that processor on a card then place that card in a cabinet.

Workaround 1: Manually build and manage occurrences, however management of large models is untenable.

Workaround 2: Invest in an automated tool solution (.e.g, System Reasoner in JPL's Open MBEE tool).

Mind the Gaps! Candidate Changes to the SysML Standard

Global Product Data Interoperability Summit | 2016

2. Occurrence Model - Prevent attribute redefinition if desired by user

- SysML Part Properties need the ability to prevent specific attributes inherited from the defining block from being redefined.

Example: Every instance of the hardware or software Definition will have the same parameter with the same size. Data Size of a Parameter is an example of an occurrence unchangeable property.

Workaround 1: Manually check selected attribute values for consistency via report generation.

Workaround 2: Invest in building a model checking capability to report anomalies and allow the users systematically adjust attributes as needed.

Mind the Gaps! Candidate Changes to the SysML Standard

Global Product Data Interoperability Summit | 2016

3. Occurrence Model - Allow attribute redefinition if desired by user, but provide means to reset to definition value

- SysML Part Properties need the ability to redefine attributes, but still have those attributes connected to the original attribute so changes on the original attribute propagate to the “redefined” attributes.

Example: Depending on where and how each usage / occurrence of the Definition is installed, the Latency of the Parameter may differ. Latency is an example of an occurrence changeable property.

Workaround 1: Manually set redefined attributes and generate report showing difference in values and manage accordingly.

Workaround 2: Invest in building a model checking capability to report differences and allow the users systematically adjust attributes as needed.

Mind the Gaps! Candidate Changes to the SysML Standard

Global Product Data Interoperability Summit | 2016

4. Occurrence Model - Ports need to be treated as first order objects

- SysML ports on Part Properties need to be uniquely instanced and not just referenced on the part. This allows proper connectivity (the part property ports are connected, not the block's defining ports) and unique port attribution based on the context of where the part property is being used. Also remove reliance on Interface blocks – attributes should be defined on the ports directly. The port attached the class should define the port type for the port on the part property.

Example: Different software configurations that are hosted on multiple installations of the same processor require unique port connections per installation.

Workaround 1: Manually create block specializations creating unique port instances and assign specialized versions of interface blocks as types

Workaround 2: Invest in an automated tool solution (e.g., Systems Reasoner in JPL's Open MBEE tool)

Mind the Gaps! Candidate Changes to the SysML Standard

Global Product Data Interoperability Summit | 2016

5. Usability – Overall complexity

- SysML intricacies can cause confusion for both modelers and other stakeholders

Example: When collaborating with stakeholders on system design and behavior, there is often confusion as to what the notation means.

Workaround 1: Try to stick to a selected set of notations and use them consistently. Modeling complex systems is hard work - if possible, keep the model notation simple.

Mind the Gaps! Candidate Changes to the SysML Standard

Global Product Data Interoperability Summit | 2016

6. Robust application of engineering units

- SysML implementation of units should be more robust, where the same attribute can support different applied units (for a particular dimension) on instances and support value conversion based on units applied or when changed – the lack of which has attributed to many critical design errors in the past.

Example: A global supplier base where air speed attribute may be applied in different contexts – MPH in U. S. developed components and KPH in European developed components.

Workaround 1: Manually create unique attributes and assign unique value types/units for each different instance and context. Perform manual conversion if reassigning units.

Workaround 2: Invest in an automated tool solution for unit assignment and conversion.

Mind the Gaps! Candidate Changes to the SysML Standard

Global Product Data Interoperability Summit | 2016

7. Support for Configuration Management (including Change Management and Product Line Management)

- SysML should be improved to support the implementation of standard CM principles (rather than relying on vendor tool implementation).

Example: The language could include constructs for uniform model management – for example, support for a git-like (branch, diff, merge) configuration and change management method.

Workaround 1: Invest in 3rd party CM solution outside of primary SysML tool.

Workaround 2: Invest in integrating CM in tool solution (e.g., JPL's Open MBEE EMS application, Cameo's Enterprise Architecture Teamwork Server application.

Mind the Gaps! Candidate Changes to the SysML Standard

Global Product Data Interoperability Summit | 2016

8. A SysML related gap - Lack of XMI support for handling the transfer of SysML tables and diagrams

- XMI standard should support common representation of SysML diagrams and tables. We currently rely on 3rd party proprietary tool approaches and other workarounds.

Workaround: Invest in 3rd party tool solution for transferring SysML diagrams and tables (e.g., Cameo's Datahub)



Conclusion

Conclusion

Global Product Data Interoperability Summit | 2016

- **System Modeling Language (SysML) is the widely adopted standard throughout industry for system modeling.**
- **However, based on experience we discovered several gaps in SysML that needed to be address before authoring large scale system architecture models.**
- **Some of the most notable gaps was the inability to reuse deep redundant assemblies (occurrence modeling) and the absence of any change management controls.**
- **We identified potential customizations and workarounds to enhance the current usability of SysML and automate many of these manual book keeping tasks.**
- **Our end goal is to collaborate with industry to identify potential improvements and lobby for changes to the current SysML specifications standard.**



Questions?