75 Fifth Street NW, Suite 312
Atlanta, GA 30308, USA
voice: +1-404-592-6897
web: www.Intercax.com
email: info@Intercax.com

Dirk Zwemer, Intercax LLC

Technote: Application of MBE to the Internet of Things

Abstract 1

Introduction 1

Model-Based Systems Engineering for the IoT 3

Modeling Product vs. Project 3

Modeling the Product 4

Modeling the Project 6

Model-Based Engineering for the IoT 8

Syndeia 8

Building the TSM 9

Viewing the TSM 10

Querying the TSM 11

The Future of MBE for the Internet of Things 12

About the Author 13

Abstract

Model-based engineering (MBE), the creation and use of a single system model distributed over multiple tools and repositories, is a critical enabler for the development of Internet of Things (IoT) products. In this paper, we present an example using a development approach starting from a SysML architecture model and using it as the core of a Total System Model (TSM) involving PLM, ALM, CAD, simulation, requirements and project management models in a diverse set of engineering software tools. Graph database technology is used to capture and explore the TSM using a range of potentially useful queries. We use IBM Rational Rhapsody as the SysML modeling tool and InterCAX's MBE solution, Syndeia, to demonstrate these concepts.

The SysML models may be downloaded from the Intercax website for Rhapsody and MagicDraw.

Introduction

IoT product development represents a rigorous series of modeling challenges.

- Cyberphysical – As a combination of software, electronic and mechanical components (and sometimes more), IoT products require multidisciplinary approaches, where no single design or analysis tool is sufficient.
- Agile – IoT products are designed to change rapidly, so the development process must be closely coupled to configuration and project management.
- Secure – As network elements, IoT products are vulnerable to outside actors. Building in security, safety and reliability requires recognition of non-obvious extended connections between features and functions.
- System-of-Systems (SoS) – As components of larger networks, IoT models must be easily federated into larger models to evaluate emergent behaviors.

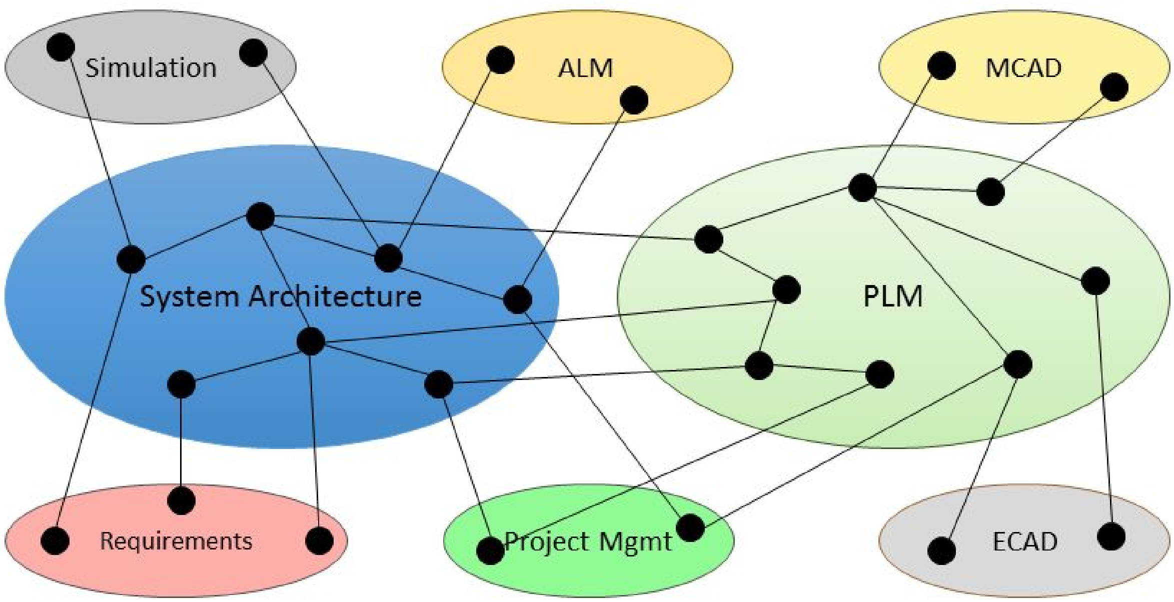


Figure 1 Total System Model as a Network of Connections inside and between Engineering Software Tools and Repositories

Model-Based Engineering (MBE) depends on a single, self-consistent digital model of the system, spread across multiple engineering tools and repositories, as illustrated in Figure 1. It extends the concept of Model-Based Systems Engineering (MBSE) of capturing a system's specification as a model, rather than a series of static disconnected documents. As the MBSE idea developed, it became clear that a single model in a single tool, for example, a SysML system architecture model, was insufficient for the purpose. In Model-based Engineering, all the disciplines and tools in the engineering process are engaged in an ongoing network.

In the first part of this discussion, we are going to use a SysML model as the core of a network of models used in the MBSE development of an IoT product, a Locator-Pager that registers nearby RFID nametags and directs audible pages to the appropriate location as part of an office-based network.

- SysML models are inclusive of requirements, behaviors, structure and analysis, making them an appropriate clearinghouse for all aspects of system data,
- They are able to incorporate both the product specification and the system development process, and
- SysML is an object-oriented modeling language, facilitating model inclusion in larger models.

In the second part of the discussion, we will consider the other models used in our system development, for example,

- Hardware designers use PLM and CAD
- Software engineers use ALM and simulation
- Systems engineers use requirements management tools
- Project managers use schedule, issue tracking and ERP systems.

An MBE approach addresses the disadvantages of this diverse toolset, such as the potential for inconsistencies between models, the difficulty in finding needed data, and the problem of identifying extended relationships between requirements, behavior and structure that give rise to emergent system effects. We will show how our MBE platform, Syndeia, creates and maintains a graph of the connections inside and between models, preserving model integrity and facilitating queries and visualization of the connections.

Model-Based Systems Engineering for the IoT

In this section, we will consider Model-Based Systems Engineering, MBSE, as representing a SysML-centric approach, although other modeling languages could also be used. This SysML model is available for download (link) for closer inspection by the reader. In the second section of this presentation, we will extend our approach to Model-Based Engineering (MBE), where multiple tools and models interact as peers within the TSM.

Modeling Product vs. Project

Every system development project has both product-specific and project-specific considerations:

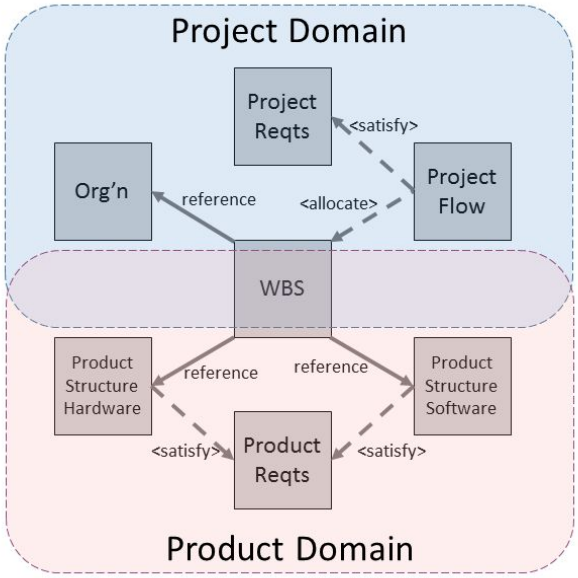


Figure 2 Intersection of Project and Product Domains

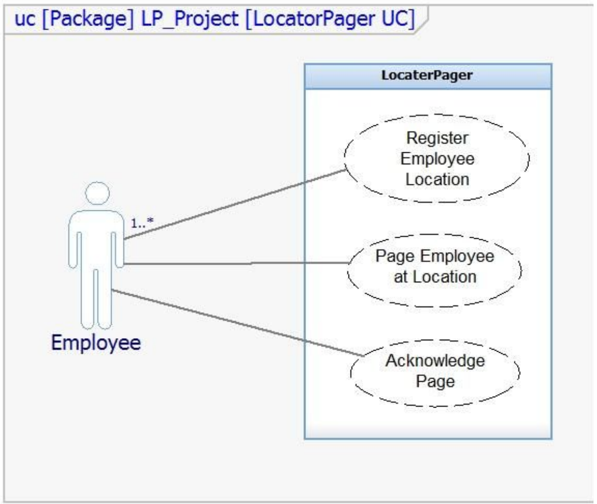
- Product-specific includes product requirements (market, technical, regulatory), product function and hardware and software design
- Project-specific includes organization structure, project requirements, and product development methodology

The intersection of these two domains is often the Work Breakdown Structure (WBS) which captures the product-specific tasks in the context of the project organization, as illustrated in Figure 2.

Ordinarily, systems engineering is concerned with the first set of considerations, project management with the second, each area with its own set of tools and models. However, the need to coordinate the two is widely recognized. A common model in a common modeling language facilitates this.

Modeling the Product

The product SysML model follows a normal pattern, starting with identification of use cases with the context of the product domain. The SysML use case diagram in Figure 3 captures the product purposes. Figure 4 and Figure 5 show the composition and interconnections of the domain, the LocaterPager (LP) interacting with RFID tags, a remote employee Location Register, and a workplace internal communications network.



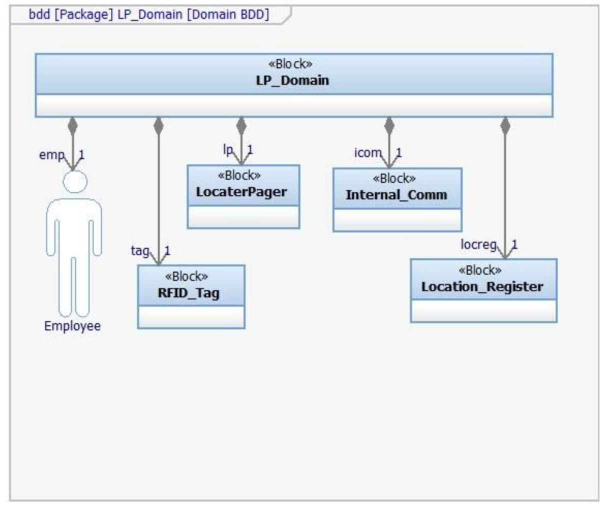


Figure 3 Product Use Case Diagram

Figure 4 Product Domain Decomposition

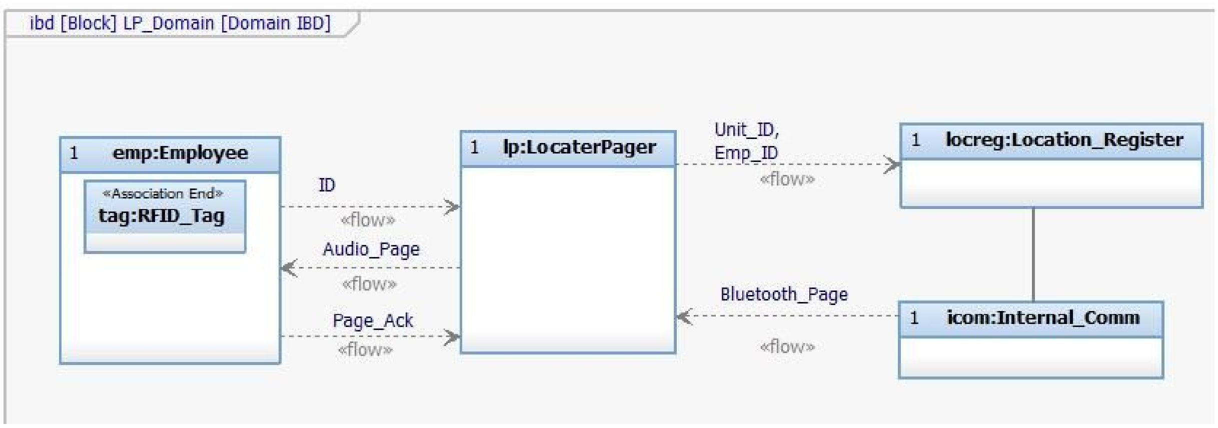


Figure 5 Product Domain Connectivity

It is outside the scope of this technote to describe a specific MBSE methodology for IoT product development, or a complete detailed model for a real product. In our example, we simply present provide a product decomposition for the LP (Figure 6) and an internal block diagram (Figure 7) for the software components. These provide the basis for demonstrating the ability to connect these elements with PLM, ALM, MCAD and simulation models in the next section.

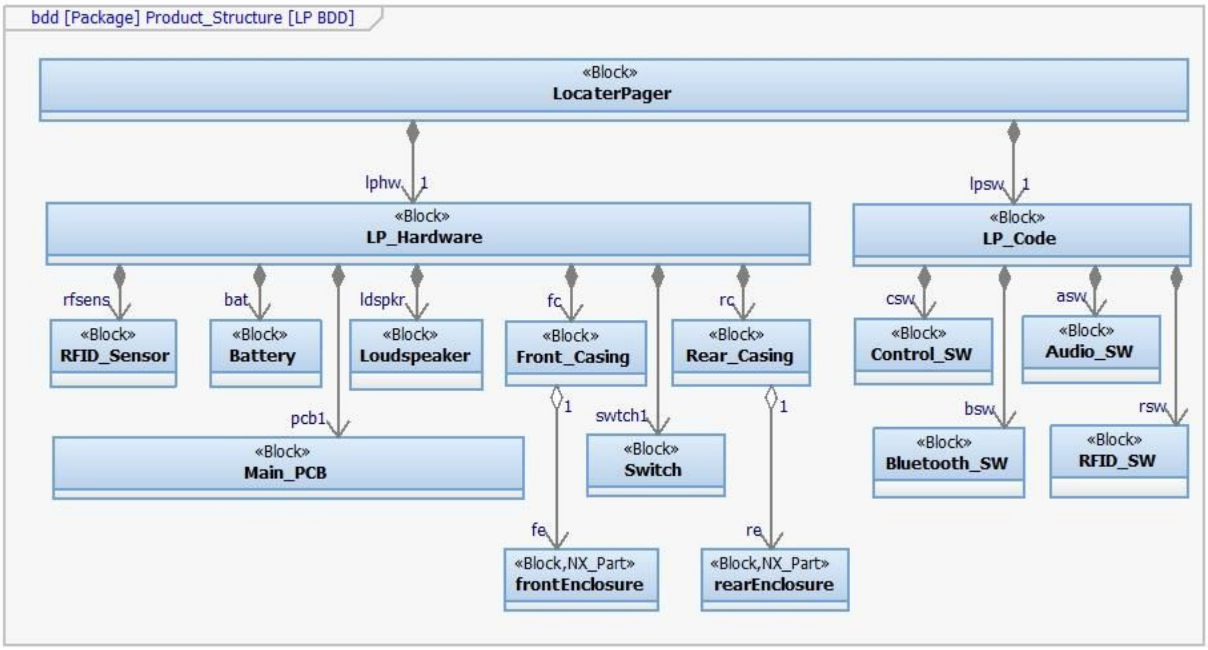


Figure 6 LocaterPager Product Decomposition

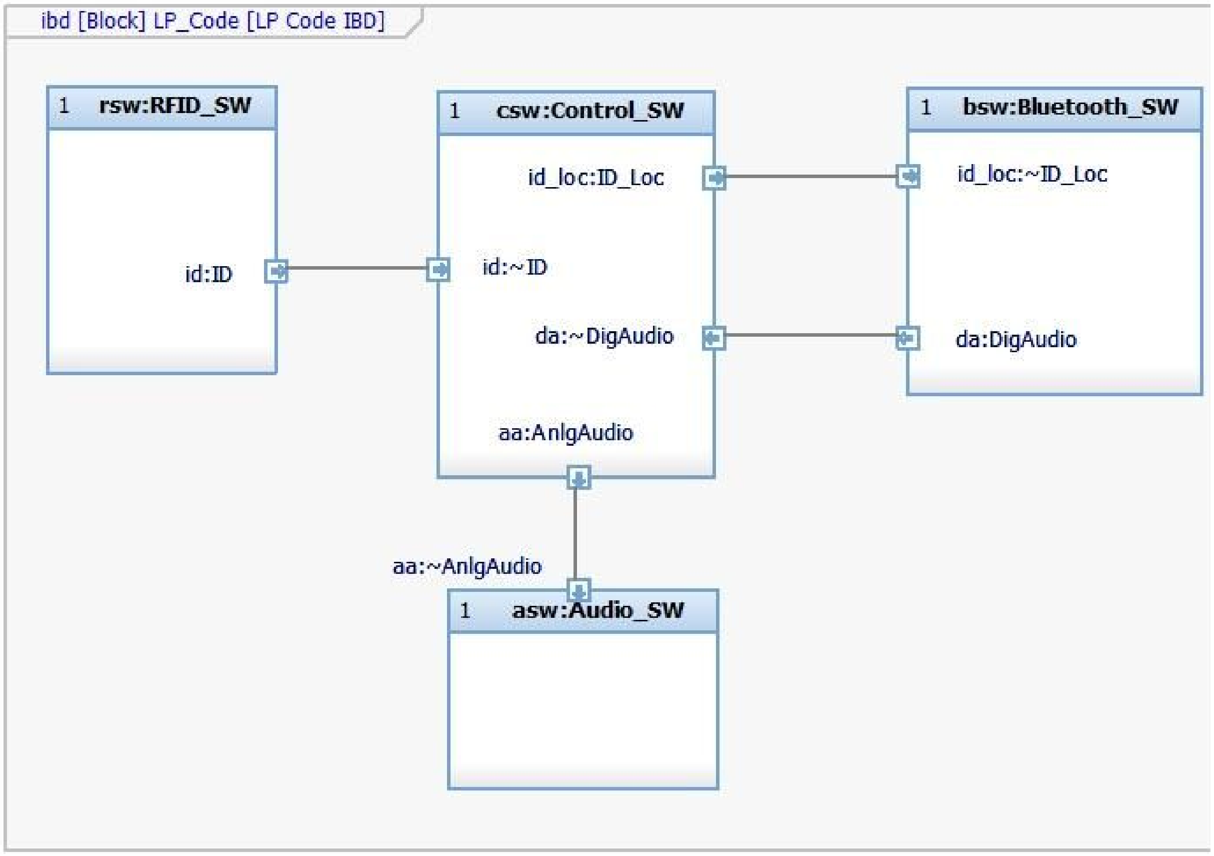


Figure 7 LocaterPager Internal Software Connections

Modeling the Project

Capturing the project organization in SysML is relatively straightforward, as illustrated in Figure 8. This differs from a conventional organization chart in being model-based rather than a simple image. The top three levels represent organizational groups rather than individuals. Individual group members who might be assigned to projects are represented generically at the bottom level with 1..* multiplicity; specific employees will be represented as specializations or instances of their roles. As we will see in the next section, employee data can be stored in an external database rather than the SysML model itself.

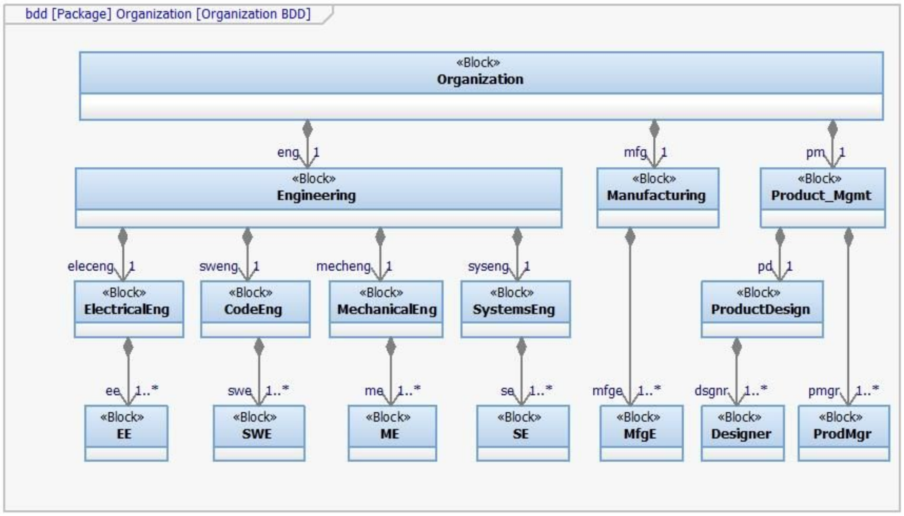


Figure 8 Project Organization Decomposition

In our example, projects are governed by a standard set of project requirements, which are realized in a standard project flow. Figure 9 and Figure 10 show representative requirements and activities from the first phase of the project, the activity Phase_1_Create_Project_Plan. Similar activities are created for Product Plan, Product Design and Product Review phases, although in our model, the Product Design phase is driven by product requirements rather than project requirements.

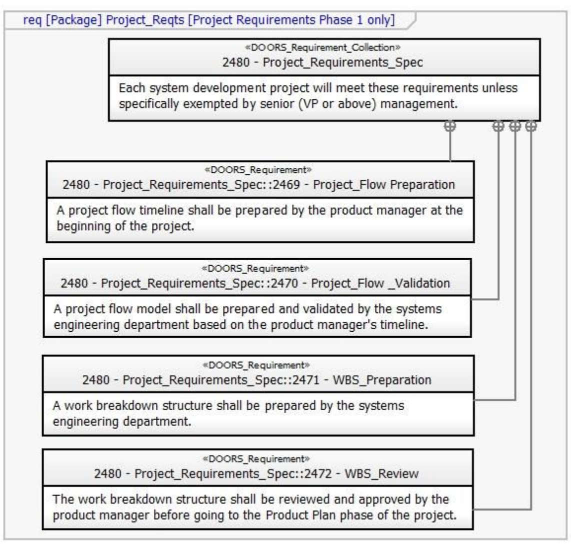


Figure 9 Project Requirements (Phase 1)

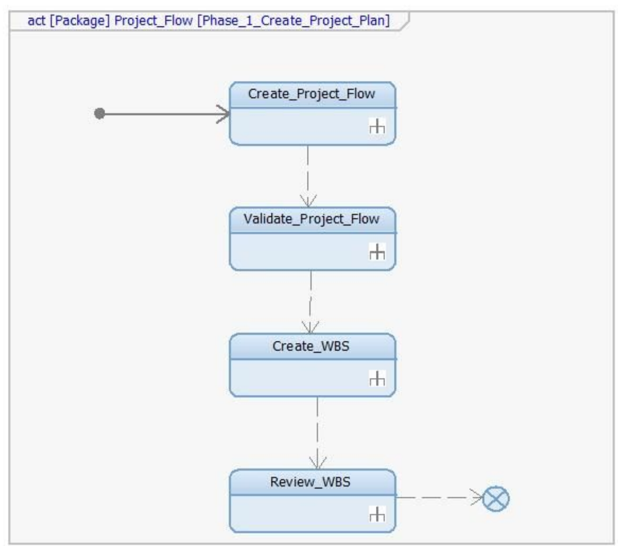


Figure 10 Project Flowchart (Phase 1)

The Work Breakdown Structure (WBS) specific to this development project includes both standard project tasks such as Product_Reqts_Creation and product-specific tasks such as Product_Externals_Design (the design of the outer casing). As with the earlier requirements and activities, each WBS block in Figure 11 can be decomposed into finer levels of detail as needed.

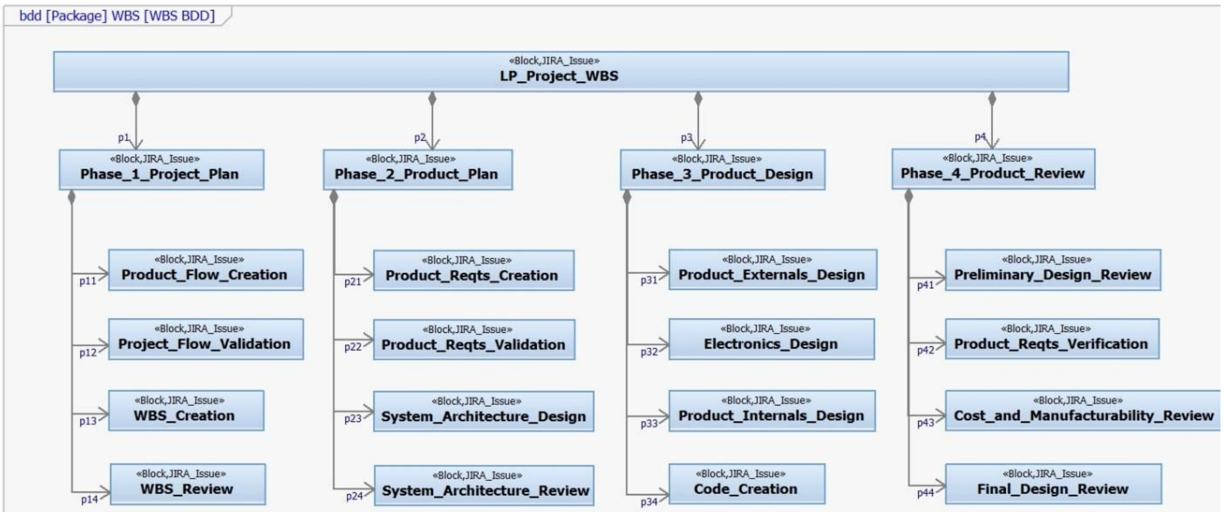


Figure 11 WBS Decomposition

There are potentially many ways to connect the four models in the project domain. In Figure 12, we present one solution, shown for the first phase project elements, using standard SysML relationships

- Project activities representing process flow “satisfy” project requirements
- Project activities are also “allocated” to WBS blocks
- WBS blocks “reference” departments within the organization responsible for the task

When the project structure is instantiated, specific employees within each of the departments can be assigned to the WBS tasks.

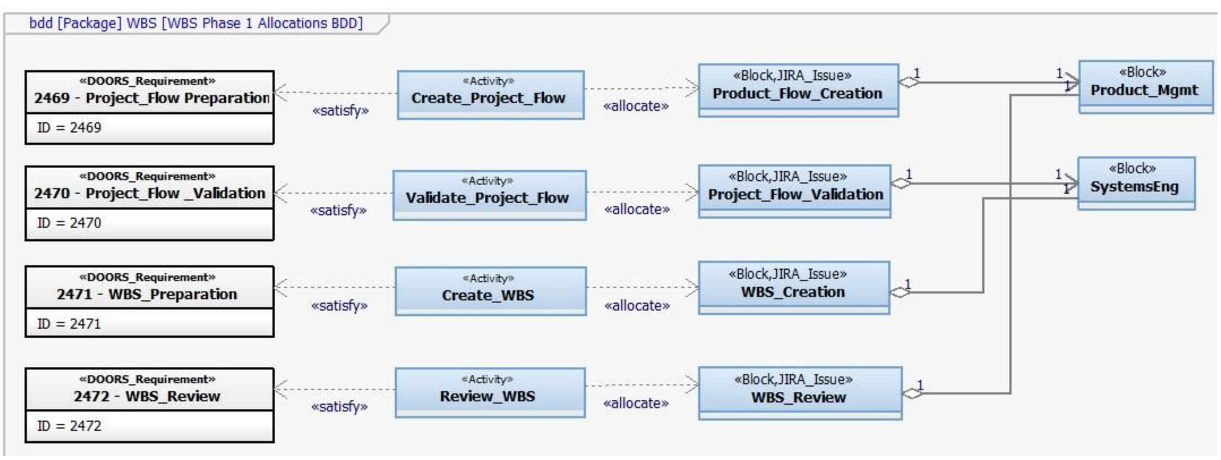


Figure 12 SysML relationships between WBS, organizations, process and requirements

Model-Based Engineering for the IoT

Syndeia

Syndeia™, from Intercax, is a platform for the practice of MBE. It uses a SysML system architecture model (Figure 13) as central hub and connects SysML elements to elements in PLM, CAD, ALM (application lifecycle management), project management, requirements, simulation and other engineering tools. We call the resulting network the Total System Model (TSM).

SysML is an effective medium for building a high-level roadmap of the system because it provides a rich language for connecting structure, behavior, requirements and analysis concepts, concepts that can map to corresponding concepts in more specialized tools. Currently, Syndeia supports two SysML modeling tools, MagicDraw (No Magic Inc.) and IBM Rational Rhapsody, and a variety of other tools, including Teamcenter and NX (Siemens), Windchill and Creo (PTC), MySQL (Oracle), DOORS NG (IBM Rational), Simulink (The Mathworks), JIRA (Atlassian), GitHub (GitHub, Inc.) and Excel (Microsoft).

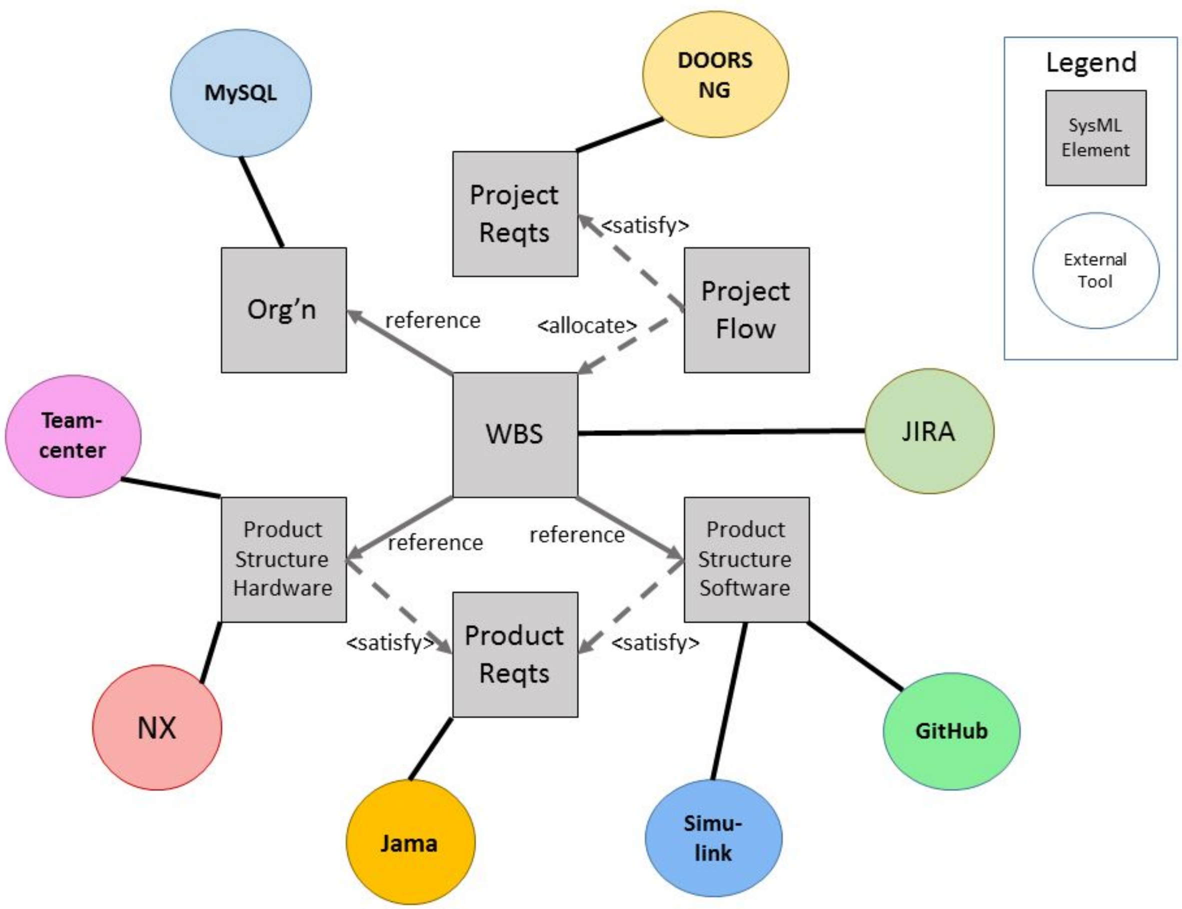


Figure 13 Total System Model concept showing multiple tools and repositories connected through SysML model

Building the TSM

To build this network of connections, we used Syndeia, the MBE platform from Intercax. Using a simple drag-and-drop interface (Figure 14), Syndeia can create a variety of inter-model connection types, ranging from simple reference links to full model transforms which allow comparison and synchronization of data between tools (Figure 15). Using Syndeia, we can populate the SysML model from the external tool, populate the external tool from the SysML model, or create a reference connection between pre-existing model elements, depending on the tools and use cases. In each case, a persistent network of connections is created and maintained by Syndeia.

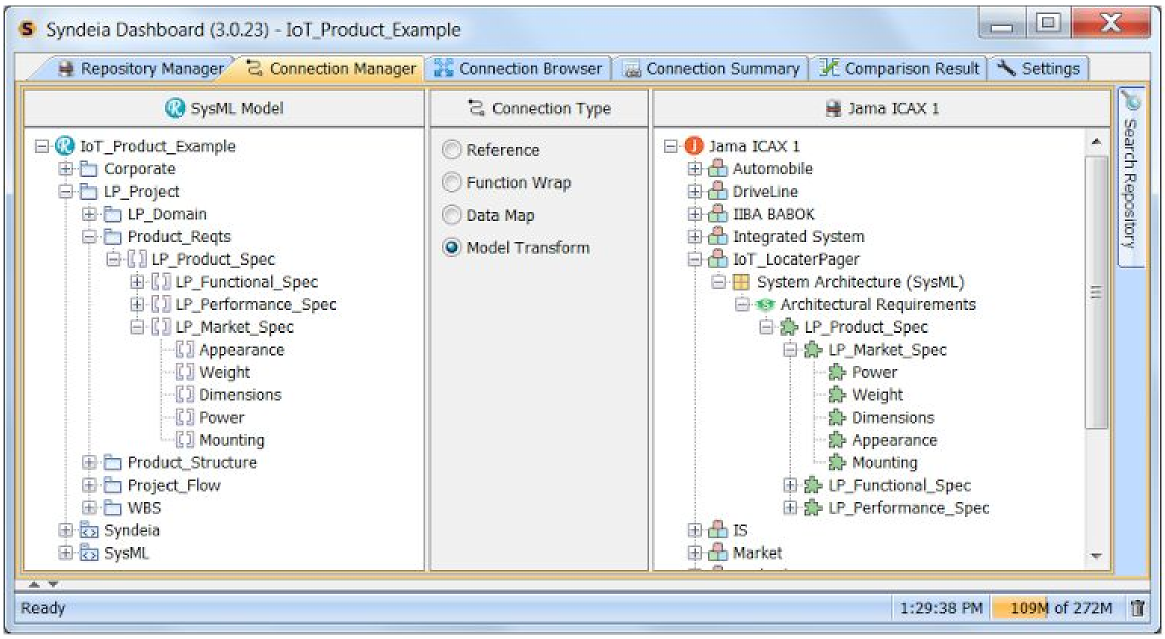


Figure 14 Syndeia dashboard showing linked product requirements in SysML (left side) and Jama (right side)

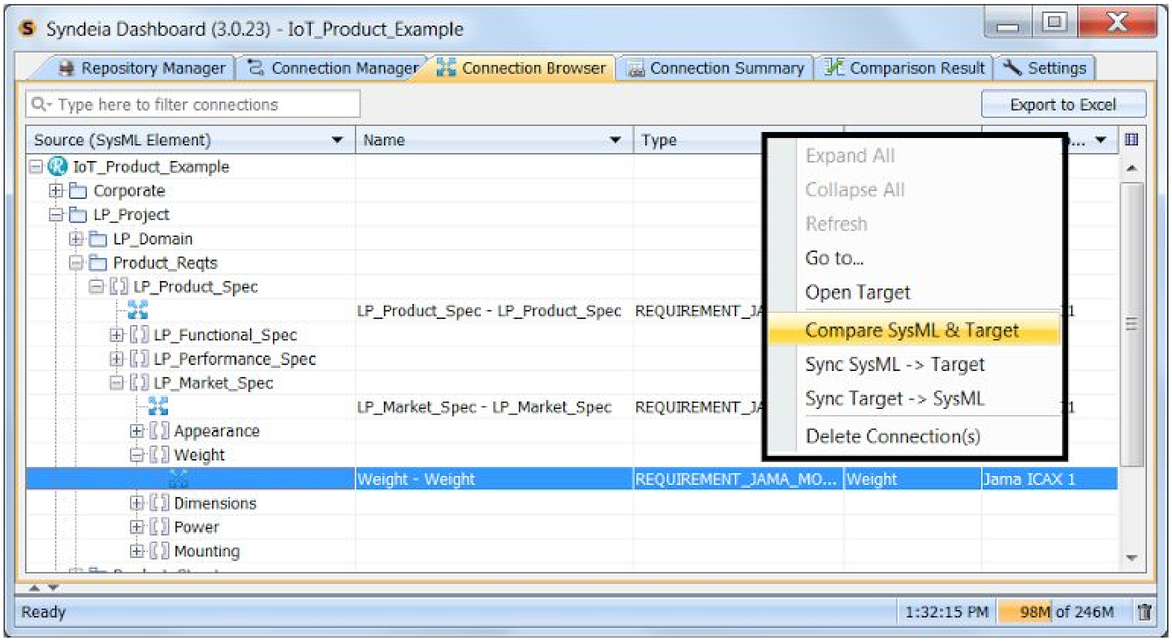


Figure 15 Syndeia dashboard showing compare and synch options across model transform connection between requirements

Viewing the TSM

A major benefit of realizing the TSM is being able to trace connections across the graph, both inter-model connections between tools and intra-model connections inside a single tool. With Syndeia, we can view all the inter-model connections in a chord plot, as in Figure 16, where the peripheral nodes in different colors represent Rhapsody, Teamcenter, DOORS NG, Simulink, NX, GitHub, JIRA, Jama and MySQL. A search box lets us look for specific node and connections.

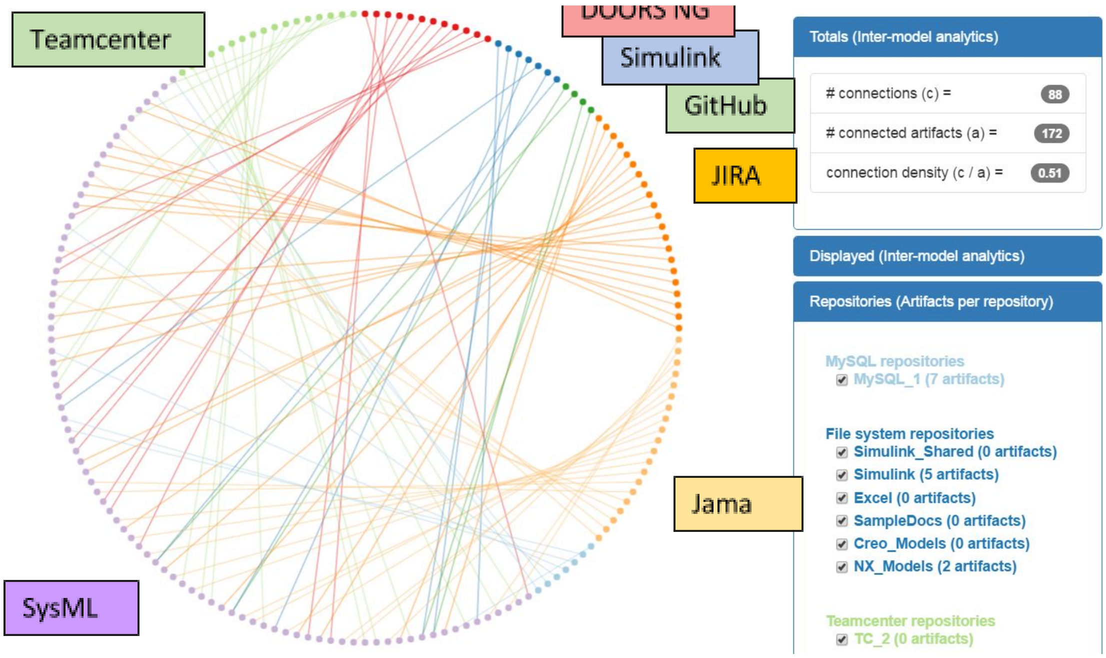


Figure 16 Chord plot of inter-model connections

While plots such as Figure 16 are useful in seeing at a glance what models are connected and how densely. We may also want to launch our visualization from a single element and expose its nearest neighbors, next nearest, and so forth, interactively. Figure 17 shows such a plot originating at the WBS Review SysML block (with the thick red outline) with its extended network of connections.

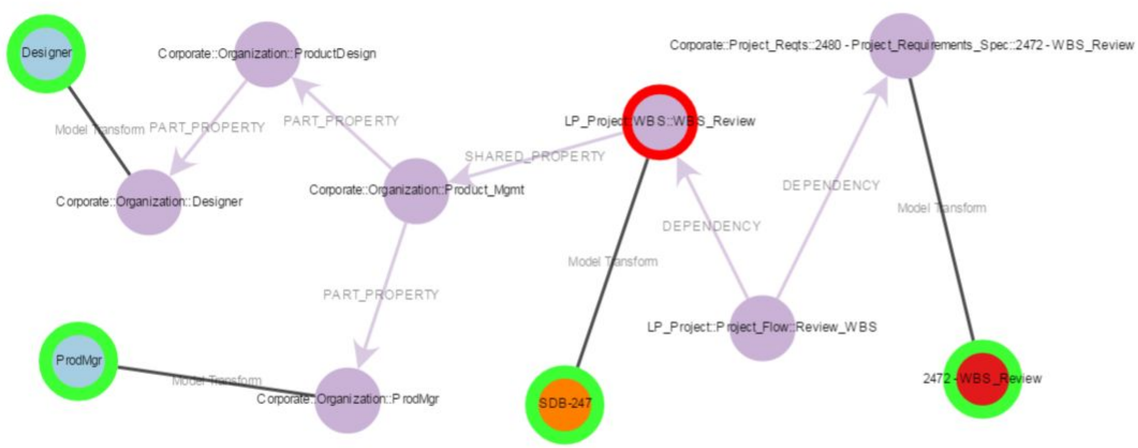


Figure 17 Auto-layout plot of inter-model and intra-model connections starting from WBS Review block

Querying the TSM

However, as the total system model becomes large, new methods to identify critical connections efficiently are required. In particular, we want to be able to explore extended chains of connections, where multiple direct links combine to connect system elements in non-obvious ways. It is such extended connections that can give rise to critical emergent behaviors and vulnerabilities that systems engineers are expected to identify.

The power of modern graph database technologies makes this possible. Using a prototype of Syndeia 3.1, to be released early in 2017, we export the graph to a Neo4j graph database with powerful query tools. Using its SQL-like query language Cypher, we can ask the graph database a variety of key questions and see the results as a diagram or a text list.

1. Show all SysML blocks connected to Jama requirements - Figure 9 shows the graphical results to this query. Jama requirements are in gray and SysML requirements are in yellow. Inter-model (Model Transform) connections links Jama and SysML elements and intra-model (SysML Containment) relationships connect the SysML requirements. Note that the graphical output does not display the full element name, but such information is easily read using the text-based format or selecting individual graphical elements.

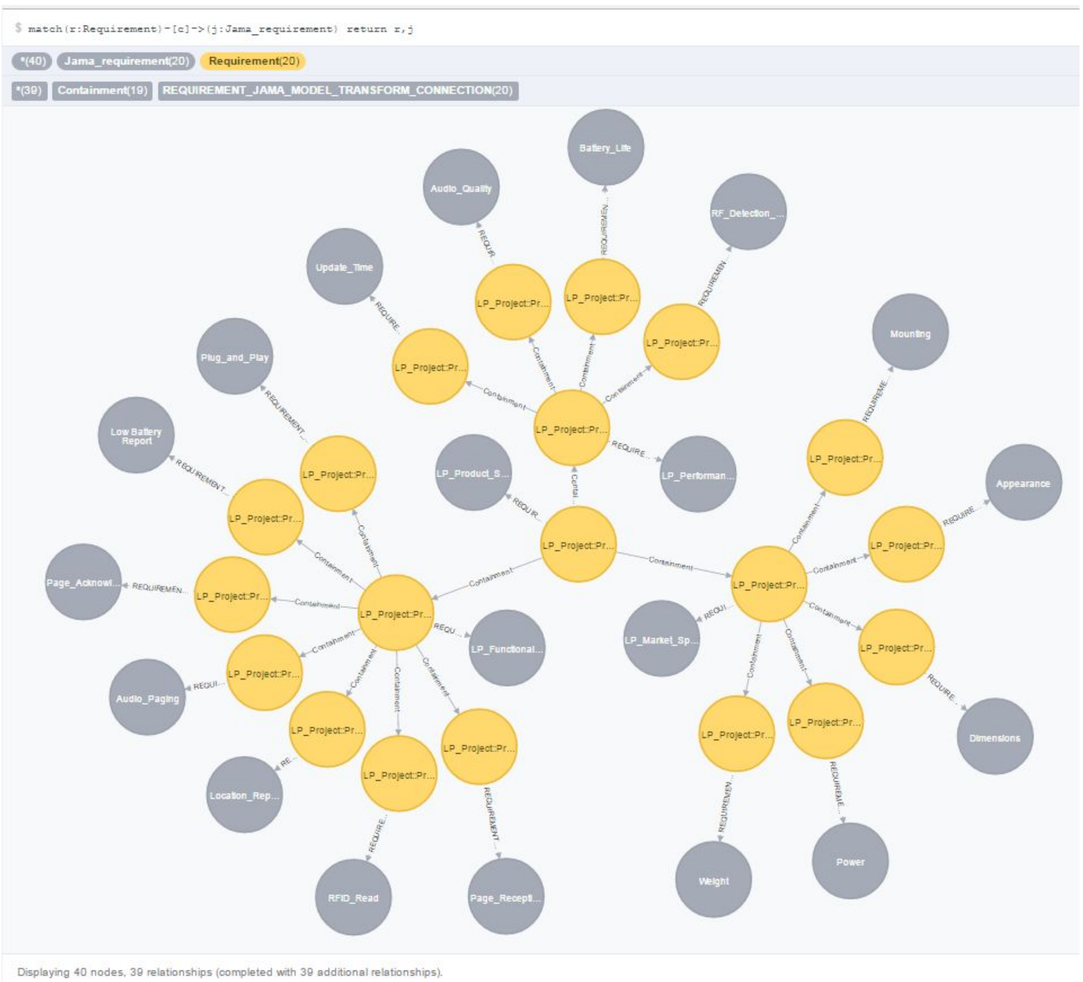


Figure 18 Query 1 - Show all SysML blocks connected to Jama requirements

2. Show all requirements that impact the front casing part, directly or indirectly – The results in

Figure 11 include Dimensions, Mounting and Weight, which are linked indirectly to the front casing as part of the overall product, as well as Appearance, which is linked directly.

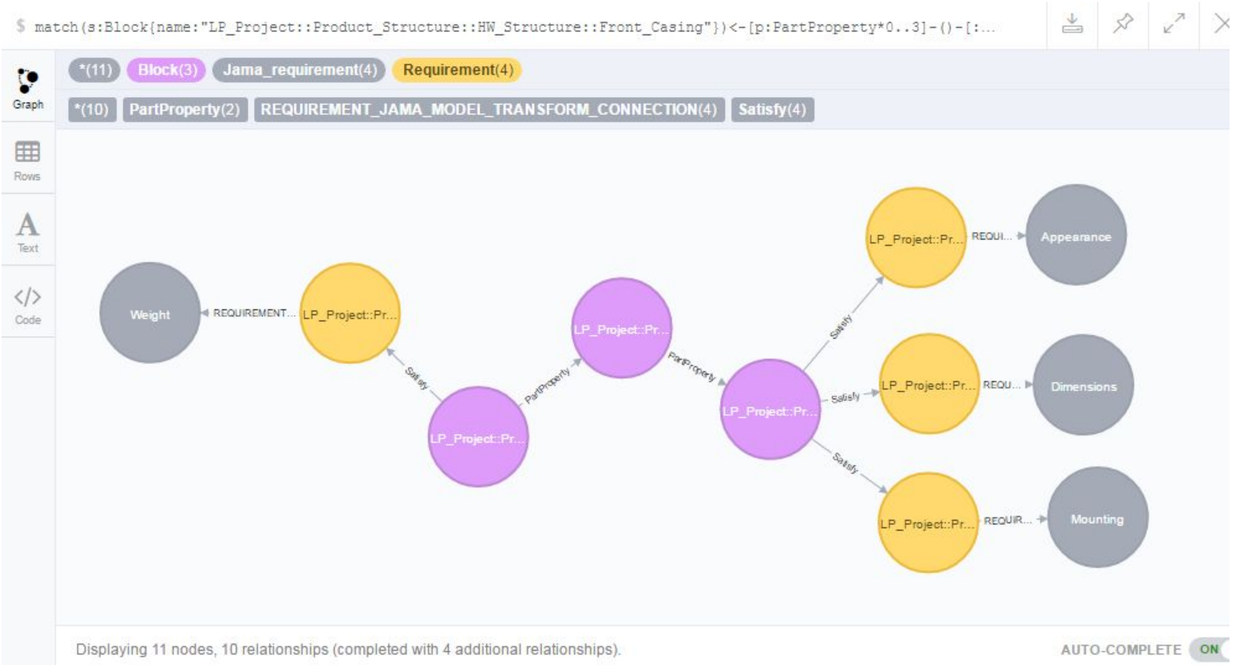


Figure 19 Query 2 – show all requirements linked to front casing part

3. Show any connection between the GitHub file “BluetoothSW” and the Jama requirement “Location_Report” – Results in Figure 12 are shown both as a text list and a diagram. There is one linkage, via a SysML requirement satisfied by a SysML block.

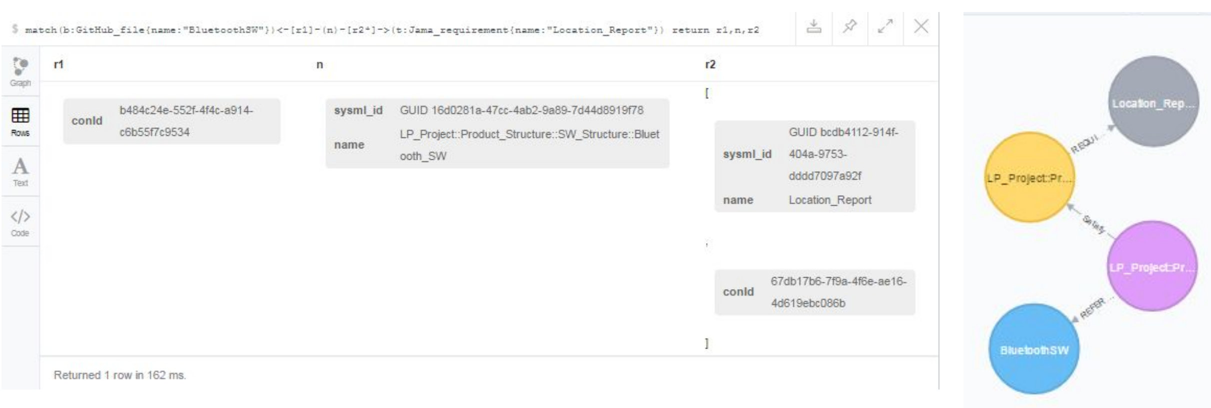


Figure 20 Query 3 – show linkage between GitHub file “BluetoothSW” and the Jama requirement “Location_Report”

The Future of MBE for the Internet of Things

The field of Model-Based Engineering is evolving rapidly. Driven by new technologies, including the Internet-of-Things, engineering software tools are being pushed to the limits. To meet our needs, MBE tools will need to evolve as well.

Two areas where further development is needed are

1. MBE will become less SysML-centric.
    a. MBE tools will become stand-alone enterprise applications rather than SysML plug-ins.
    b. Intra-model and inter-model connections <u>not</u> involving SysML elements will be stored.
    c. The TSM will be accessible to all team members, not just systems engineers.
    d. Team members will access the TSM through other channels than SysML diagrams.
2. MBE tools will focus more on user management and access. The value of collaboration, critical to developing networked IoT products, does not blind us to the need to protect proprietary technology. The ability to link models must be accompanied by the ability to restrict what parts of the models are accessible to what users.

Work in these areas is in progress at several organizations, including both users and software vendors. Standards development for both IoT and MBE is facilitating this, although standards should not outpace industry experience in developing real systems. As more and more products and services become IoT-based, we will see organizations adopt scaleable, robust MBE solutions to help develop them.

About the Author

Dr. Dirk Zwemer (dirk.zwemer@intercax.com) is President of Intercax LLC (Atlanta, GA), a supplier of MBE engineering software platforms like Syndeia and ParaMagic. He is an active teacher and consultant in the field and holds Level 4 Model Builder-Advanced certification as an OMG System Modeling Professional.
For further information, visit us at www.intercax.com or contact us at info@intercax.com.