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

Dr. Dirk Zwemer, InterCAX LLC

Tech Note: Smart Home – Modeling the Internet-of-Things with SysML

Part 3 Procurement to Evaluation

Contents

Abstract

This is the third in a series of technical notes describing the application of Model-Based Systems Engineering (MBSE) to the specification, design, procurement and evaluation of an Internet-of-Things (IoT) system. This section uses the SysML Reference Architecture in combination with specific product data added in the previous section to build a configured system model. The parametric constraints embedded in the model are used to evaluate cost and power consumption of specific model instantiations. Finally, the SysML model is used to seed a Bill of Materials in a commercial PLM repository, laying the groundwork for the next stages in the engineering process.

Introduction

Model-Based Systems Engineering (MBSE) has wide applications, from aerospace to consumer products, because we live in an increasingly complex and interconnected world. We also live in an increasingly customized world, where digitally sophisticated "Makers" use CAD and 3D printing to create new products of their own unique design.

One exciting prospect is that SysML will become more widely known outside the systems engineering community. Individuals motivated to design their own IoTs could adopt it as their means of specifying, procuring and evaluating possible networks. SysML has the potential (already emergent in some areas) to grow in some key ways critical to wider adoption

- It can be customized and simplified for IoT tasks, even to the extent of creating an IoT domain-specific language (DSL)
- It can become Software as a Service, available over web browsers on an as needed basis
- Libraries, profiles and reference models can be distributed to support the community's needs, as 3-D product CAD files for 3-D printing are being distributed today

In the first note in this series, we described an MBSE process going from requirements specification to a detailed multi-level functional design. In the second of the series, we created a structural reference architecture, allocated functions to structural elements, added parametric modeling for cost and power consumption, and imported IoT component product data from external databases. In the third note, we will

- create a specific configuration, a four-room home,
- check functionality against the reference model,
- build a complete four-room IBD connection model,
- evaluate cost and power consumption using the embedded parametric models and
- use the final design to create a Bill of Materials in a commercial PLM system.

The next notes in the series describe how this model can link to models of different type and scale, including CAD models of the home, time-varying simulation models, and larger SysML models, e.g. the Smart Grid for electrical power.

Creating a 4 Room Smart Home Configuration

At the end of Part 2, we had added a number of components to the SysML model from external product data repositories (in this example, a MySQL database). These represent real, commercially-available products and are treated as specializations of the generic components in the Reference Architecture, with the addition of ports, operations and default value for the value properties. It is now necessary to create a structure framework in which to use these in a specific design configuration.

There are multiple ways to create variant configurations in SysML. We will first create an architectural variant using specialization, redefinition and subsetting, in order to describe the connectivity of the system. Later, we will create instances of that architectural variant to evaluate cost and power consumption in a range of scenarios, e.g. external temperature.

Figure 1 defines a 4Rm Home as a specialization of Home, with three subsystems that represent redefinitions of the generic subsystems (Audio, Security, and HVAC) of the Reference Architecture.

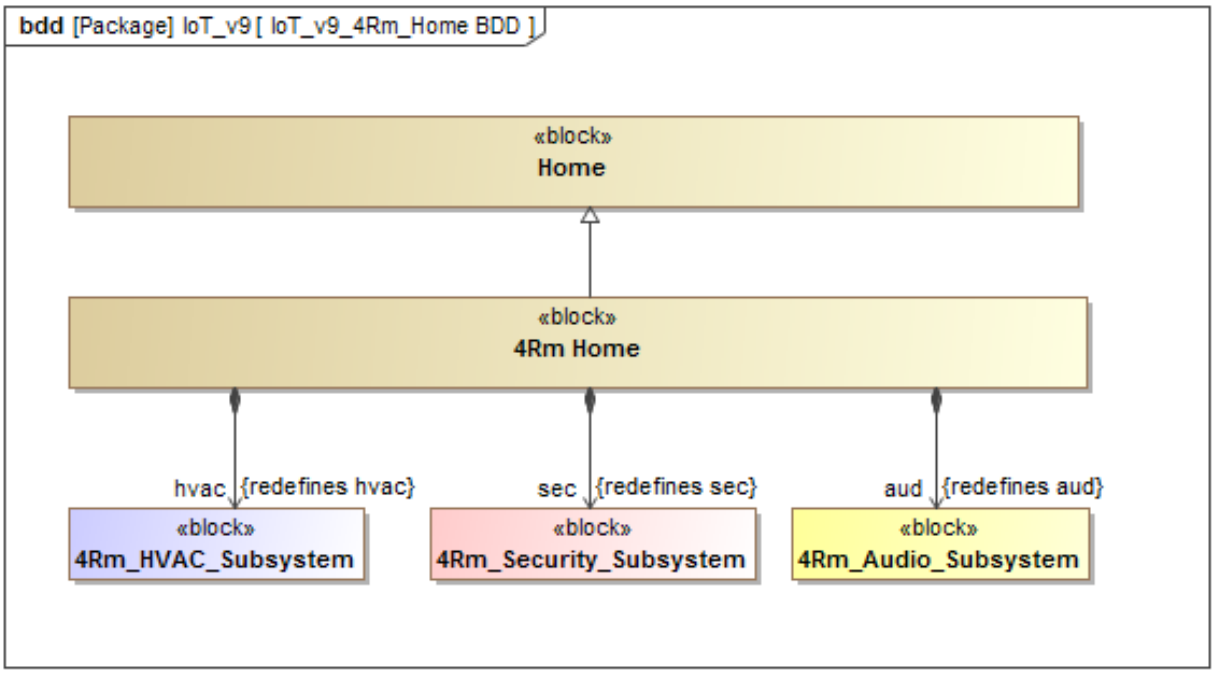

Figure 1 4Rm Home Block Definition Diagram, showing redefinition of part properties

Figure 2 shows the definition of one of those subsystems, the 4Rm_Audio_Subsystem block.

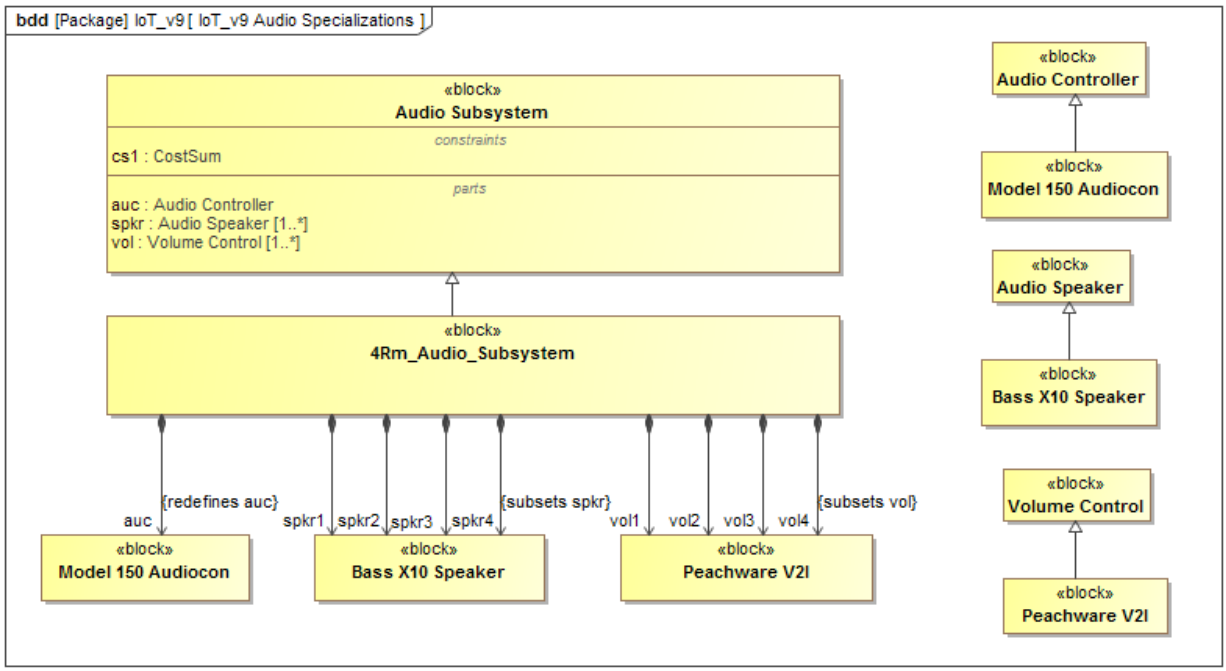

Figure 2 4 Rm Audio Subsystem block definition diagram, showing specialization and subsetting of audio components

4Rm_Audio_Subsystem inherits the parametric cost model from the Audio Subsystem block. It redefines the Audio Subsystem block's part properties, typing them by the real audio component blocks brought in from the MySQL database. Where Audio Subsystem had the potential for one-to-many parts of a particular type, e.g. spkr, 4Rm_Audio_Subsystem creates four individual properties that subset the original. As individual part properties, they can be shown individually on an IBD. As subsets of the original 1..* part property, they participate in the parametric relationships defined for the Reference Architecture. Block definition diagrams very similar to Figure 2 have been created for the other two subsystems, 4Rm_HVAC_Subsystem and 4RM_Security_Subsystem.

Allocating Reference Activities to Operations

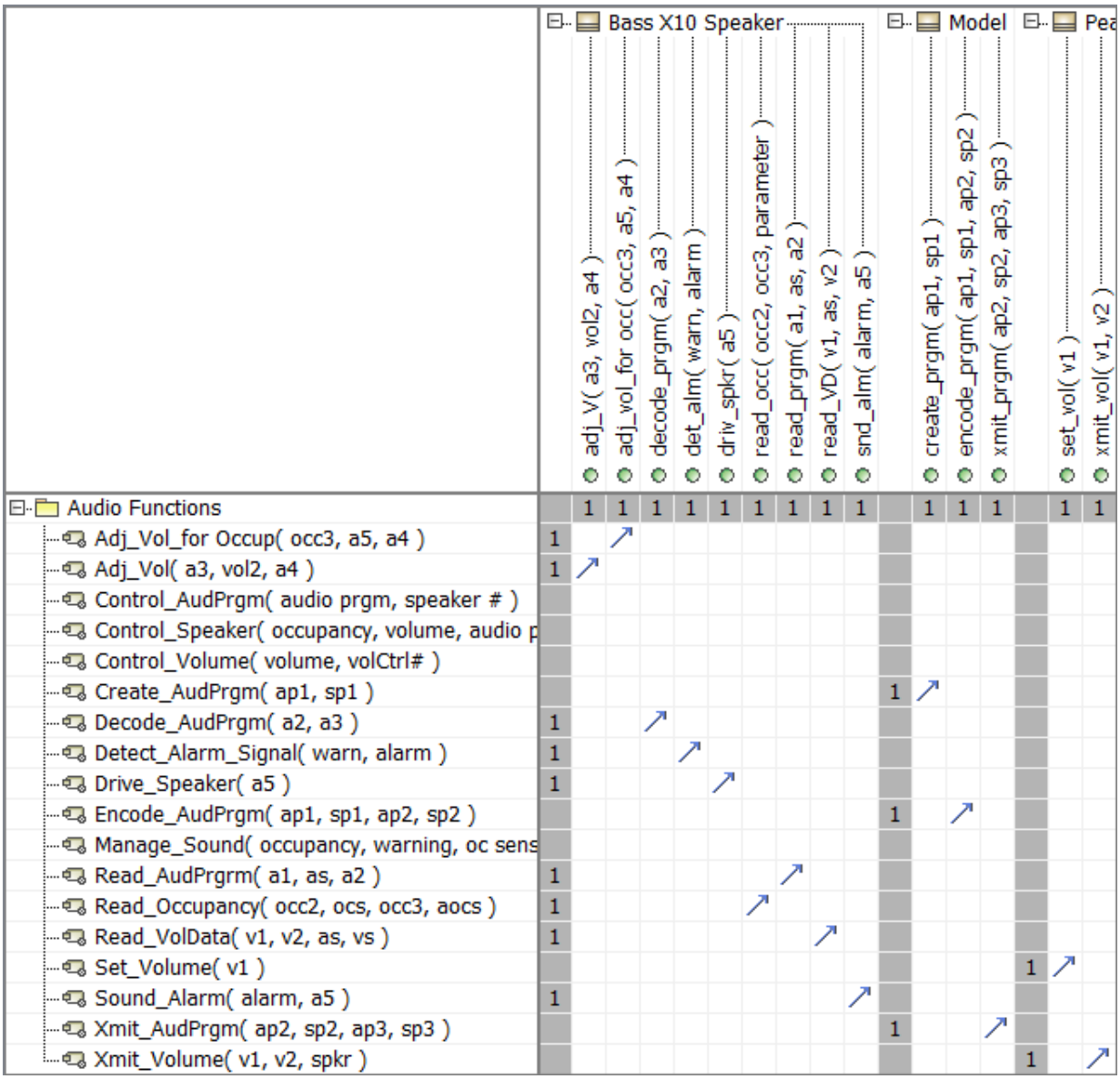

Table 1 Allocation matrix showing Reference Model Activities allocated to Audio Component Operations

Before proceeding further with the 4Rm configuration, it is important to check that the functionality of the real components, e.g. Bass X10 Speaker, supports the required functionality of the Reference Architecture. One way to approach this is shown in Table 1, where allocation relations have been created from the activities in the Reference Architecture to the operations of the components assigned to the 4Rm Home. In this table, only audio behaviors are shown, but the modeling is also done for the other subsystems. Note that all the leaf-level activities are covered by operations, but higher-level activities are not.

It may be helpful to build those higher-level activities for the 4Rm Home to confirm required functionality. Figure 3 shows the BX10_Control_Speaker activity for the BX10 Speaker. It appears very similar to Figure 8 in Part 1 for the generic Control_Speaker activity, but the actions represent software operations of the BX10 Speaker which were loaded from the MySQL database. With this approach, higher-level functionalities of the specific configuration can be validated with respect to the Reference Architecture.

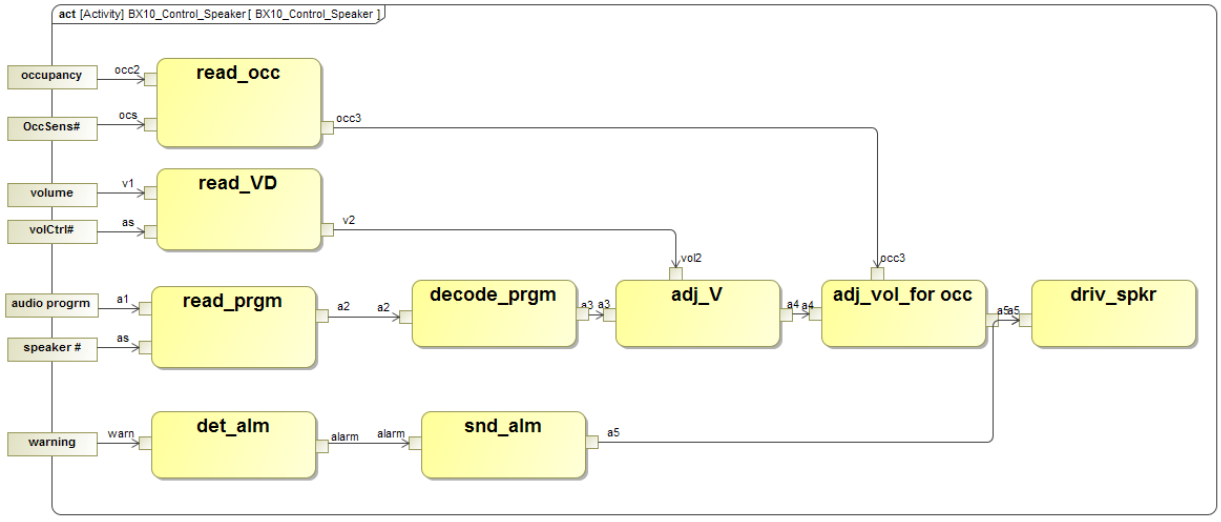

Figure 3 Activity diagram for BX10 Speaker using component operations

Specifying Connections with Internal Block Diagrams

The creation of a "connection" diagram for the 4Rm configuration is simple using the approach described. Figure 4 is an internal block diagram showing connection between all the IoT components (including the kitchen appliances in the upper right corner). The three subsystems (Audio, Security and HVAC) are arranged left to right; the four rooms or zones are stacked vertically. Because each component and each room is shown separately, it is straightforward to include a diversity of components in model and number as well as interconnections, beyond the relative symmetry of this example.

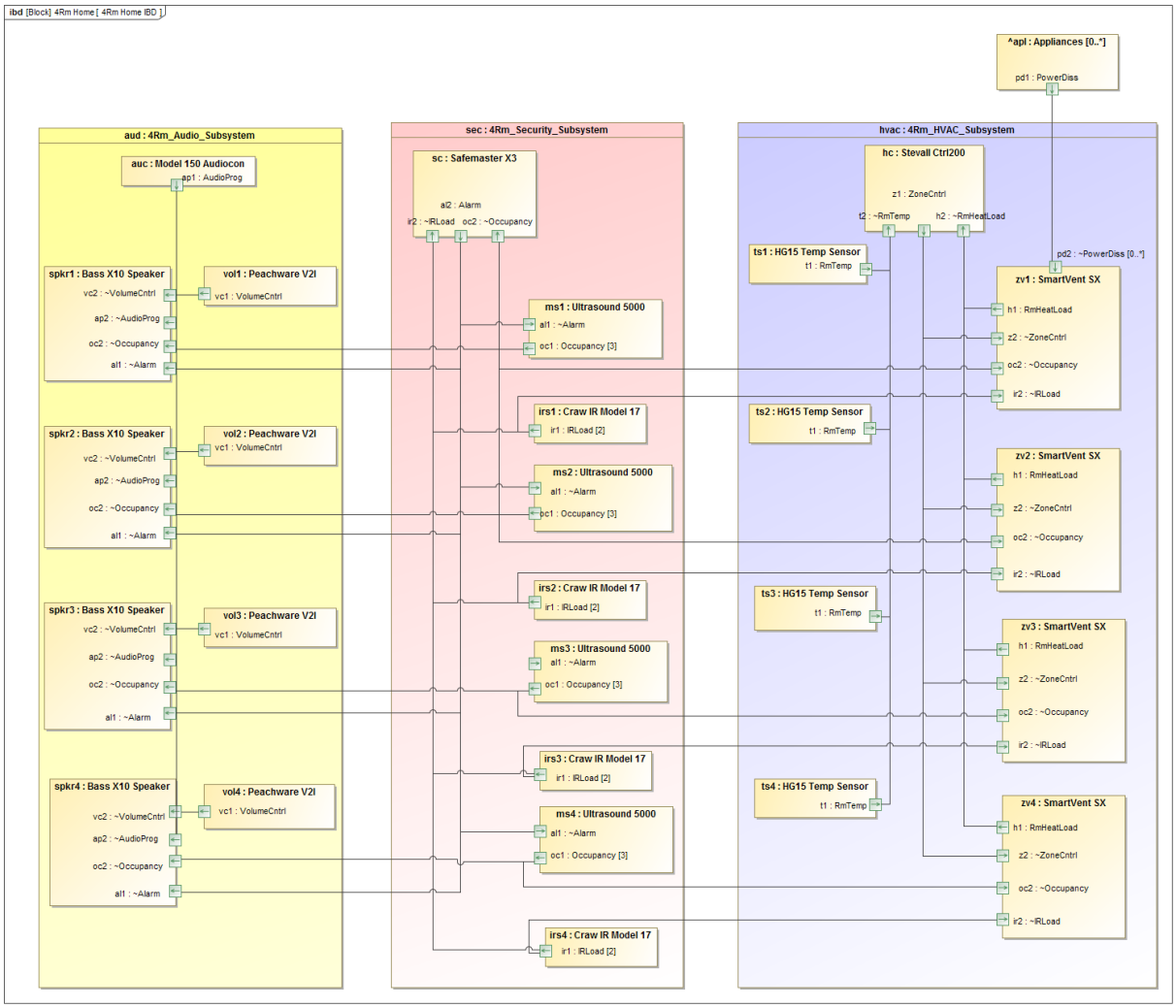

Figure 4 Internal Block Diagram for 4Rm Home IoT Connectivity

Figure 5 shows an enlarged detail from Figure 4. It shows that the parts are now connected through proxy ports. These ports were brought in using Syndeia from the MySQL database of component characteristics. Using ports for connections potentially allows better validation, since connected ports can be validated with respect to number, direction, and type, which further relates to communication medium and standard.

In each subsystem, all components connect to a central controller, shown at the top. However, there are also communications between individual components in each room, including across subsystem lines. In this model, there are no communications links between the subsystem controllers, although such could be added easily with components designed for this purpose. These diagrams also do not show communication with systems outside the home; Part 4 will address some of the possibilities in this area.

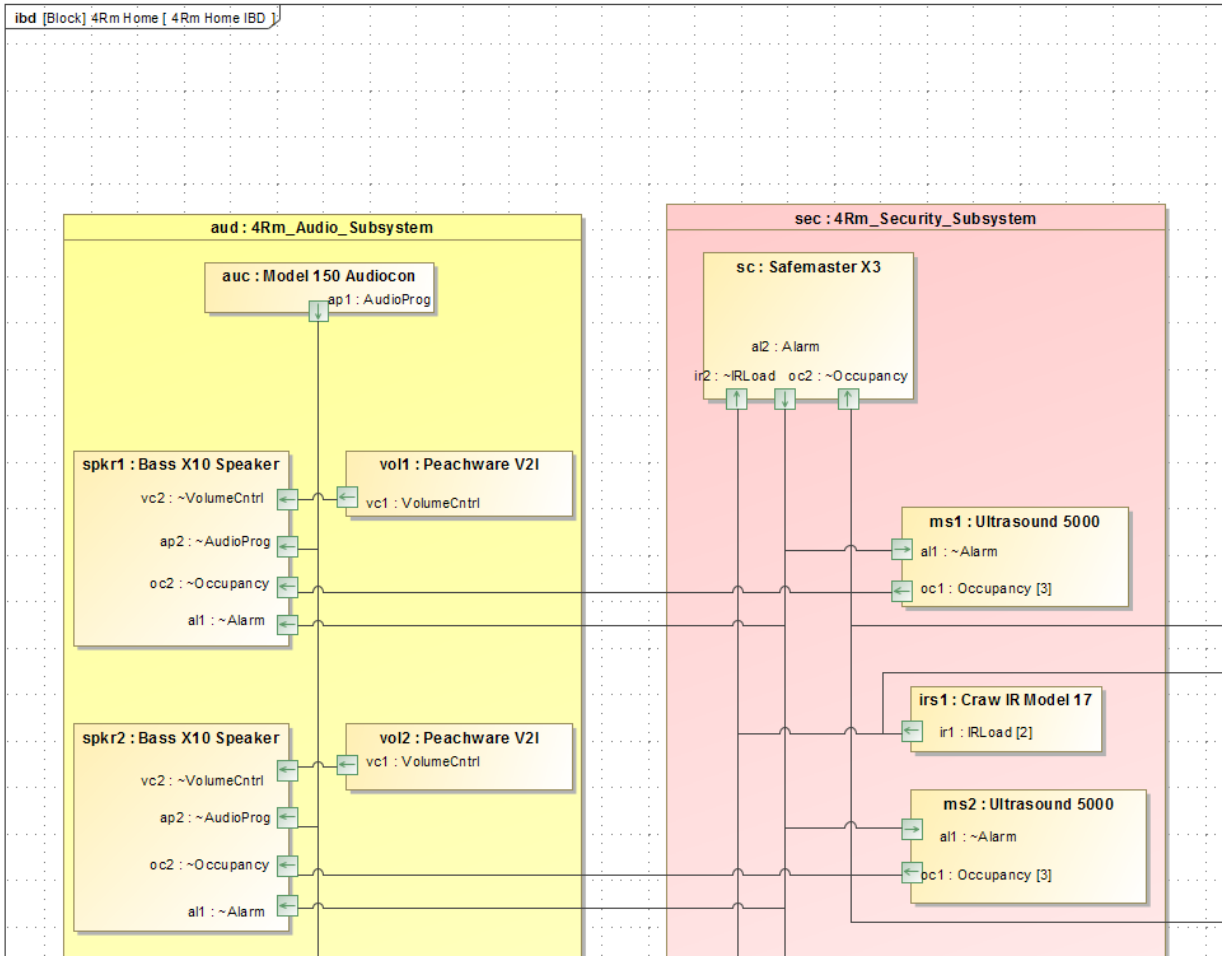

Figure 5 Internal Block Diagram for 4Rm Home IoT Connectivity (detail from upper left quadrant of Figure 4)

Evaluating the Cost of the System

Having designed a system, using commercial IoT components, it is also probably "smart" to evaluate cost and performance before purchasing and building. The parametric models we created in Part 2 can be used for this purpose. For the cost model, we considered only the capital cost of the components. This was entered into the SysML model using default values for the cost property brought in by Syndeia from the MySQL database. Additional costs for installing, operating and maintaining the network could be added to the parametric model, if needed.

The next step requires the creation of an instance of the configured model. This can be done manually or using utilities created by the tool manufacturer. For the cost model, only instances of the structural blocks are required, but instances of the analysis blocks (see Part 2, Power Analysis Model) can be created at this time, as well. The slots representing part and reference properties must be accurately assigned. The causalities for the component costs are given; cost values at the Home and subsystem level are unknowns.

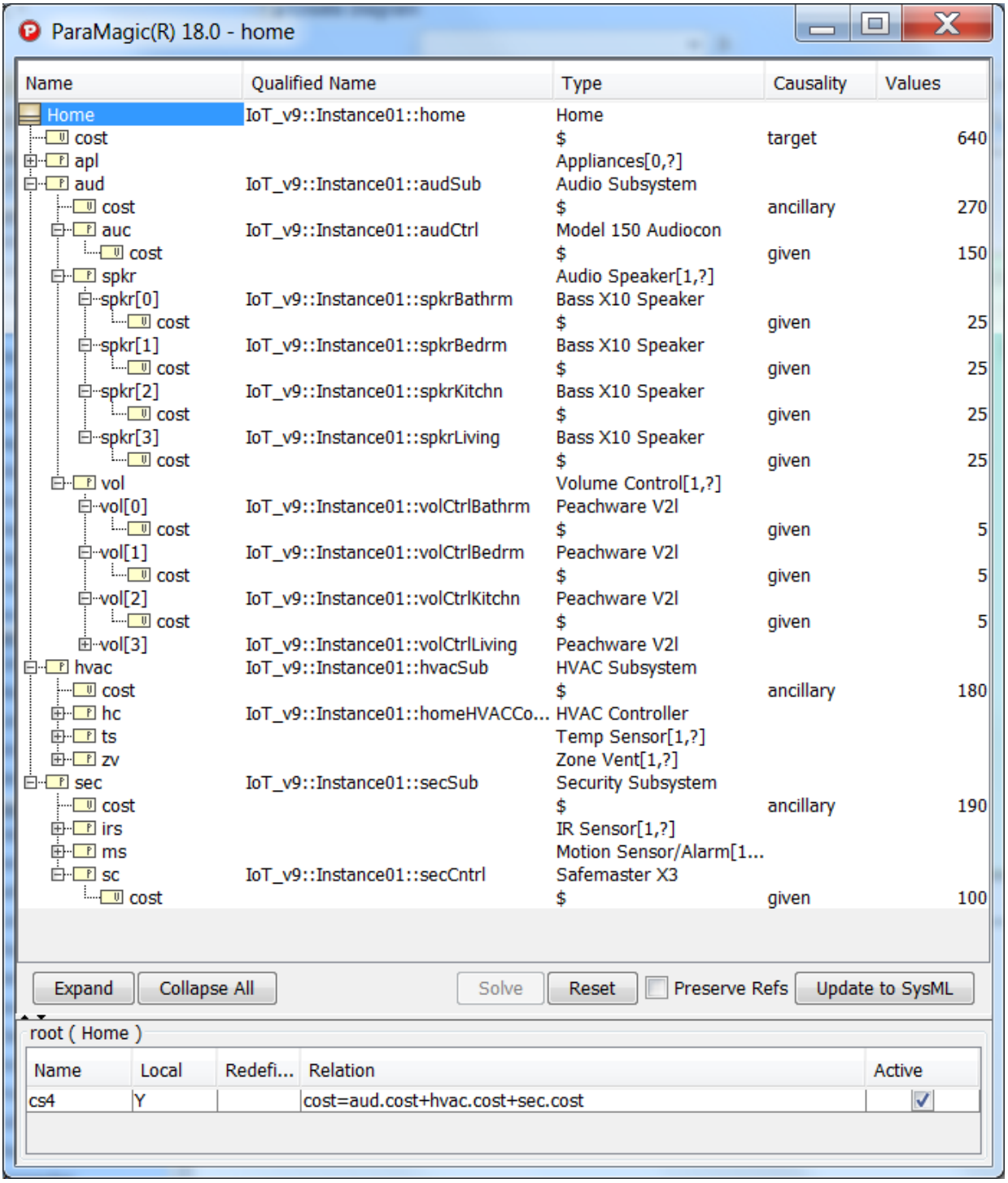

Figure 6 ParaMagic Browser for IoT System Cost, after solution

Figure 6 shows the ParaMagic browser after solving the instance. It has been partially expanded to show the given prices for the audio components. The rooms have been named kitchen, living room, bedroom and bathroom and the instances named accordingly. The total system cost is $640 for this specific configuration.

Evaluating the Power Consumption of the System

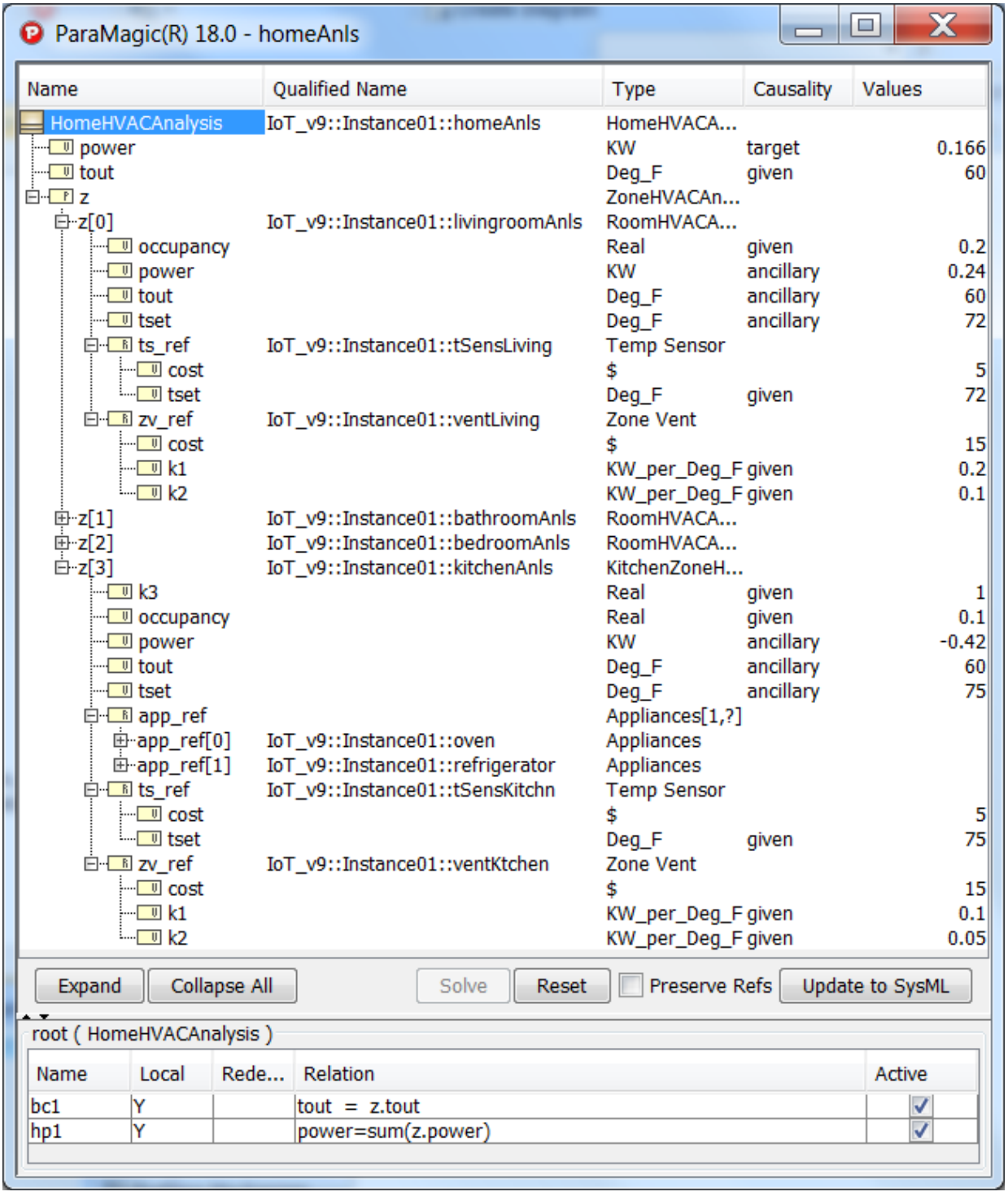

Figure 7 ParaMagic Browser for IoT System Power Consumption, after solution

The parametric model in Part 2 also calculates the power consumption of the HVAC system, as a function of

Temp Outside	Power Usage
Deg_F	KW
20	2.8
30	2.2
40	1.5
50	0.8
60	0.2
70	0.3
80	1.8
90	3.2
100	4.5

Table 2 Trade study results

- the outside temperature
- the setpoints for the individual rooms
- the heating and cooling efficiencies for those rooms (power consumed per degree difference between setpoint and external temperature), and
- the average occupancy of the room

For a particular set of temperatures and occupancy rates, Figure 7 shows the solution with an average HVAC power consumption of 0.166 kilowatts. Table 2 shows this value as a function of outside temperature.

Creating a PLM Bill of Materials

Having created and validated the design in SysML, one possible next step is to use this model to seed a bill of materials (BoM) in a PLM (Product Lifecycle Management) repository. With Syndeia (InterCAX), this is a single step drag-and-drop operation.

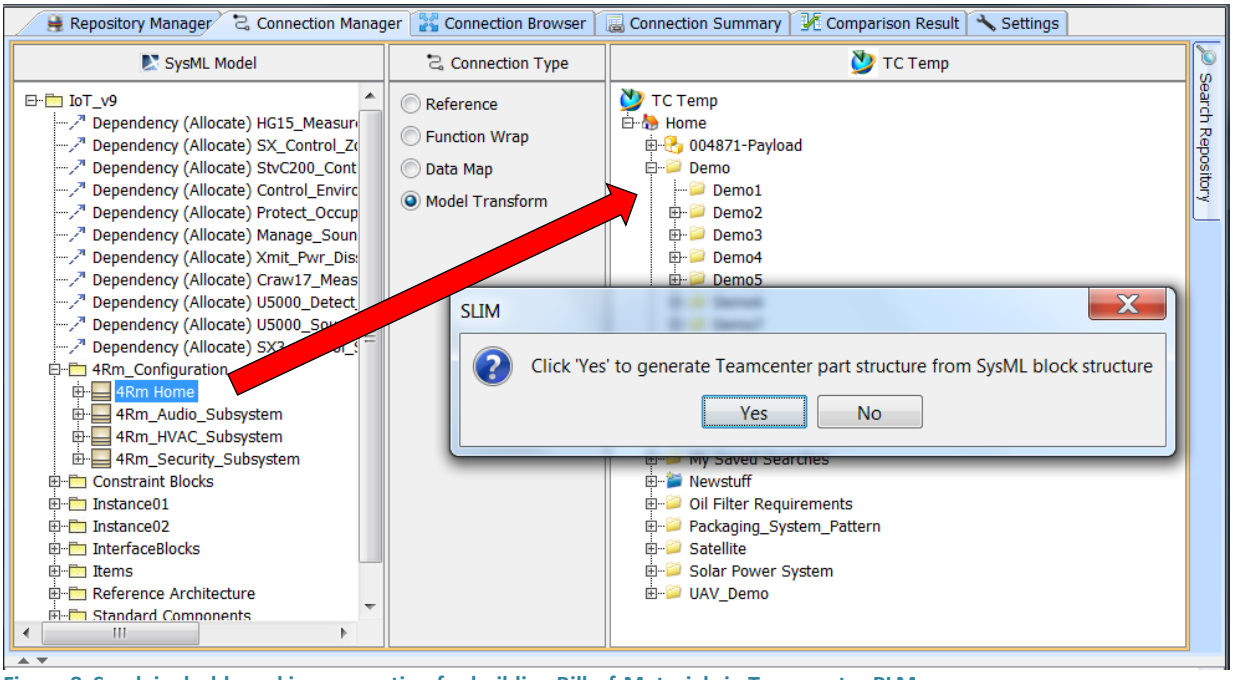

Figure 8 Syndeia dashboard in preparation for building Bill-of-Materials in Teamcenter PLM

Figure 8 shows the SysML model in the left column with the top-level block of the configured system 4Rm Home. In the right column is a Siemens Teamcenter PLM repository to which the system engineer has write access privileges. The 4Rm Home block is dragged in the dashboard on top of an empty product folder Demo1 in the Teamcenter repository.

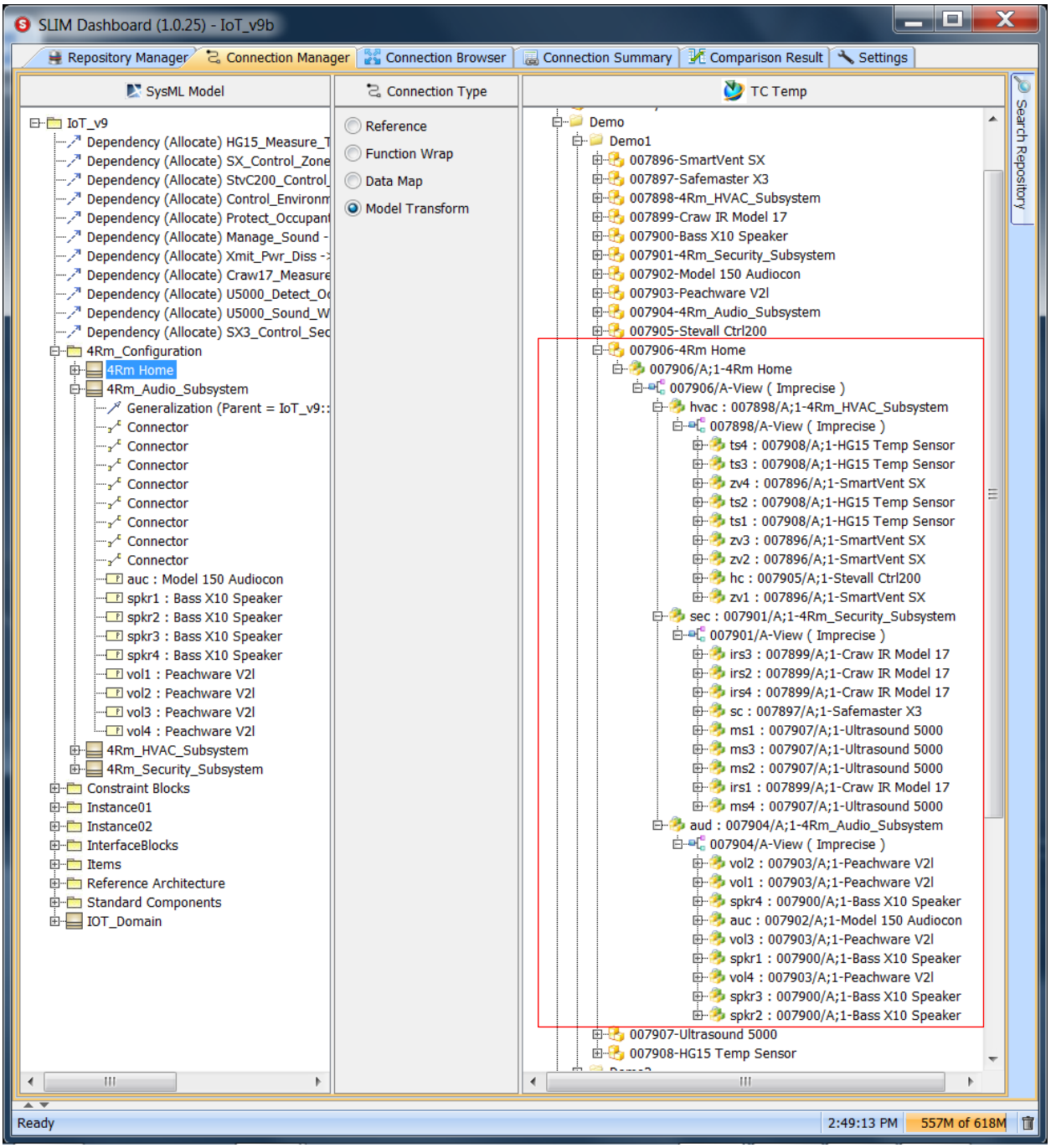

Figure 9 Home BoM Version A;1 (right column) generated from SysML model

After the operation, a complete PLM structure (outlined in red in Figure 9) is created, with its three subsystems and many IoT components. This structure could become the starting point for other engineering work processes based on conventional PLM tools. One advantage of the Syndeia software approach to System Lifecycle Management is the each of the SysML blocks retains a persistent connection to the equivalent item in the PLM repository, as shown in Figure 10. Changes at either end of the connection can be detected, compared and updated in either direction as PLM or SysML models change.

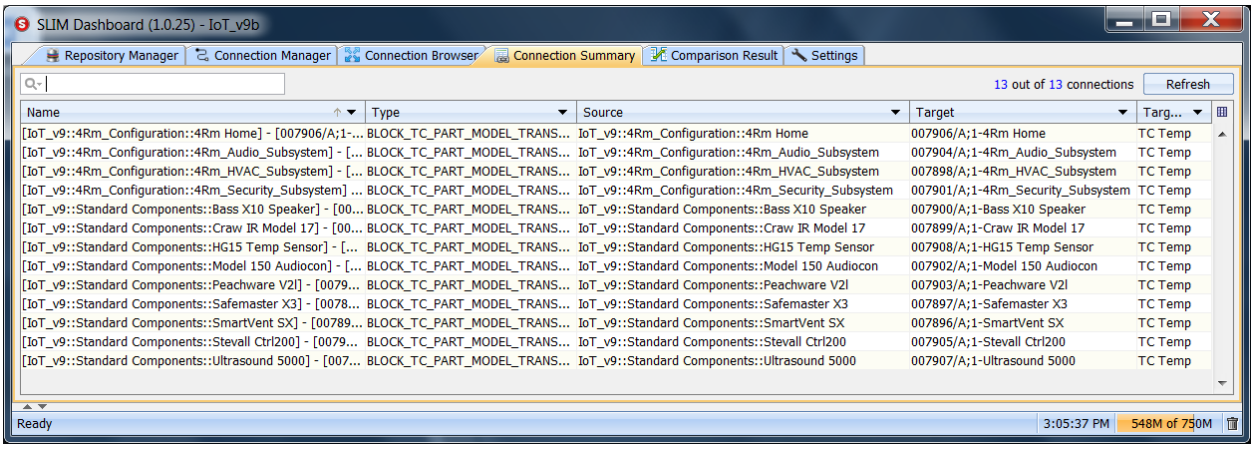

Figure 10 Syndeia Connection Summary showing persistent connections between SysML and PLM parts

Summary

Our objective in the first three notes of this series has been to outline a base process creating an Internet-of-Things model for a Smart Home. The sequence has been

- Specify
- Design
- Procure
- Evaluate

This base process relied primarily on SysML, supported by links to federated models, tools and data sources. We began with requirements, which drove a Reference Architecture, first functional, then structural, then parametric. We linked the model to a database with "real" product information and used that data to build and evaluate an actual system configuration. Throughout the process, we looked back to provide traceability in the development, allocating requirements to functions, functions to structures, activities to operations.

In future notes in this series, we will show how a SysML IoT model can be federated to other models and other tools. This includes acting as a building block in larger SysML models, e.g. the Smart Grid. It will also include linking to CAD models, e.g a CAD model of the Smart Home itself, and more sophisticated simulation models, e.g. a Simulink model of power consumption variation over a daily cycle.

About the Author

Dr. Dirk Zwemer (dirk.zwemer@intercax.com) is President of InterCAX LLC, Atlanta, GA and holds OCSMP certification as Model Builder - Advanced.

For further information, visit us at www.intercax.com or contact us at info@intercax.com.