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

Author: Rose Yntema (rose@intercax.com)
Date: Nov 11, 2015

Technote: Syndeia Excel Capabilities

Abstract

This technote presents the capabilities of Intercax products Syndeia[1] and ParaMagic[2] to connect system architecture models (SysML) with Excel spreadsheets. Three scenarios are demonstrated. In the first, Excel tables and their rows will be used to generate SysML blocks and instances, and create connections between the two. Then, these connections are used to compare and sync SysML block value properties (or instance slots) and Excel table rows bi-directionally throughout the development process. In the second, the reverse will be demonstrated, using SysML blocks and instances to generate a table in Excel. Finally, in the third scenario setup and execution of SysML parametric models for cost and mass rollups will be demonstrated, using the values synchronized from Excel tables.

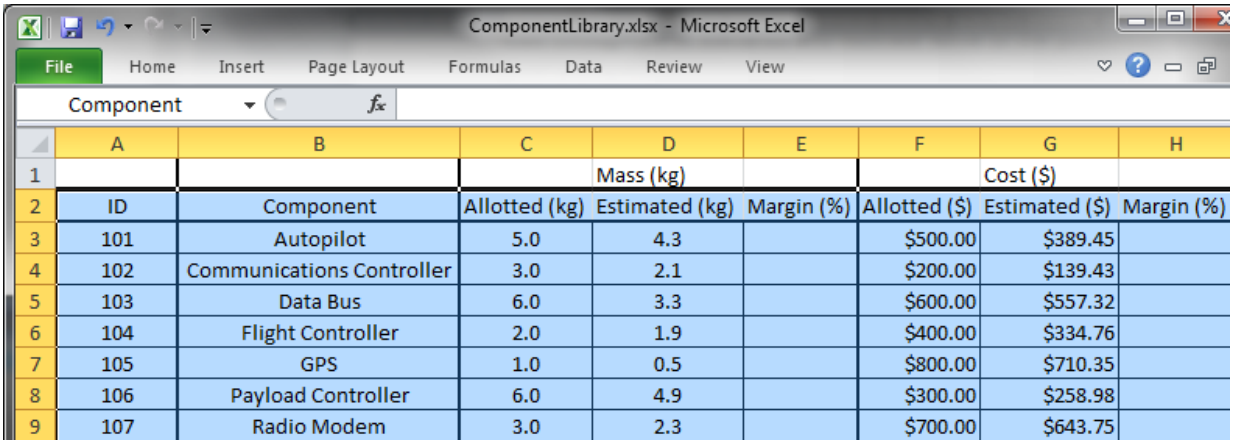
Introduction

Many engineering projects utilize tables in Excel that contain much of the same information that is needed by system engineers to model structure and interconnections in SysML. Syndeia from Intercax provides the ability to connect the two and ensure that values of key parameters such as mass and cost stay in sync, so that all team members are on the same page when using different software. Starting with a component library or equipment list in Excel table form, modelers may seed blocks and instances in SysML, or vice versa. Modelers may also build parametric mass and cost rollups of the components and values that were seeded from Excel. This can be done using SysML parametric modeling along with the ParaMagic solving tool from Intercax. This technical note will walk through these scenarios and illustrate some of the possibilities available for using Syndeia and ParaMagic with Excel and SysML.

[1] Syndeia – www.intercax.com/syndeia
[2] ParaMagic – www.intercax.com/paramagic

Scenario 1 – Creating SysML blocks and instances from a library of components in Excel and maintaining the two in sync

Part of a simplified UAV (Unmanned Aerial Vehicle) component library table in Excel is shown in Figure 1. The selected area of the workbook ComponentLibrary.xlsx has had a name defined, "Component," seen at top left in place of the selected cell range. Defining this name will help Syndeia "find" the table later.



	A	B	C	D	E	F	G	H
1				Mass (kg)			Cost ($)	
2	ID	Component	Allotted (kg)	Estimated (kg)	Margin (%)	Allotted ($)	Estimated ($)	Margin (%)
3	101	Autopilot	5.0	4.3		$500.00	$389.45	
4	102	Communications Controller	3.0	2.1		$200.00	$139.43	
5	103	Data Bus	6.0	3.3		$600.00	$557.32	
6	104	Flight Controller	2.0	1.9		$400.00	$334.76	
7	105	GPS	1.0	0.5		$800.00	$710.35	
8	106	Payload Controller	6.0	4.9		$300.00	$258.98	
9	107	Radio Modem	3.0	2.3		$700.00	$643.75	

Figure 1 Partial component table in excel with name defined

Figure 2 shows the Syndeia dashboard, Connection Manager tab. A local file system repository that contains ComponentLibrary.xlsx has been loaded on the right hand side. When the workbook has been expanded, worksheets and named cells and tables can be viewed. Each row in the Excel table (Figure 1) shows up in the Syndeia Dashboard (Figure 2, RHS) in *column_name (value)* format. To begin the process of generating elements in SysML from this table using Syndeia, we select Model Transform as the connection type (middle panel) and drag the Component table (RHS) to a SysML package (LHS). This creates a Component block with value properties corresponding to the columns of the table, as can be seen on the left hand side of the figure below. The reason the Model Transform connection type is used here is so that later on structure-level changes, such as addition or removal of block value properties and Excel table columns, can be detected and synchronized bi-directionally.

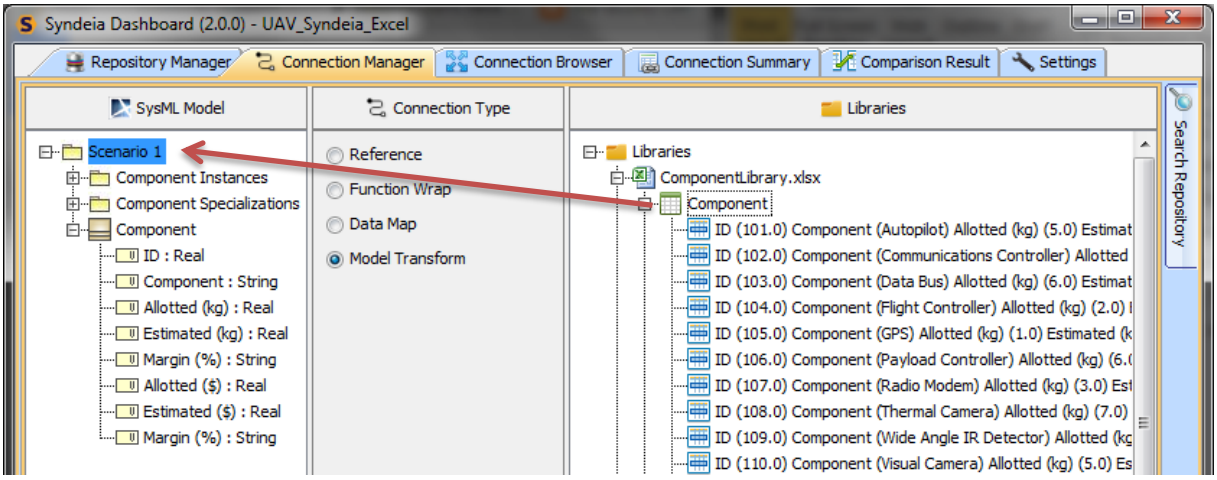


Figure 2 Syndeia dashboard Connection Manager view of Component table and corresponding block

We can now begin to use rows in the table to generate either block instances in SysML with populated slot values, or specializations of the Component block with default values for value properties, corresponding to the values in table rows. For this we select Data Map as the connection type and begin dragging the Excel table rows to the destination SysML packages, and select whether to generate an instance or a block specialization. The Data Map connection type is used at this point because it will facilitate compare and sync of attribute values later on. The instances generated in this way are shown below in Figure 3 and the block specializations are shown in Figure 4, expanded to show the values and generalization relations below. Data map connections have been created between these instances / blocks and the corresponding Excel table rows, and they can be renamed without losing this connection.

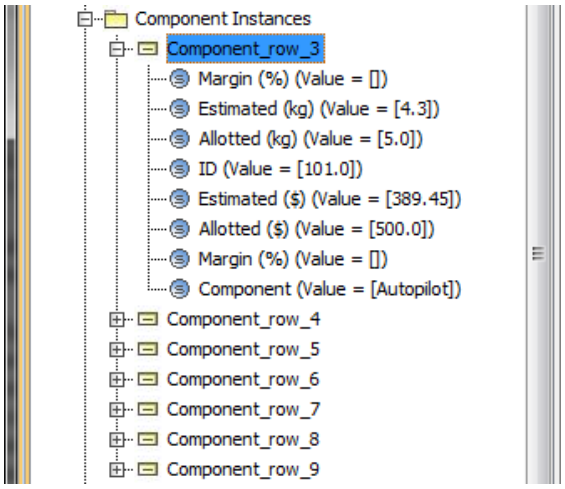


Figure 3 Instances of Component from excel rows

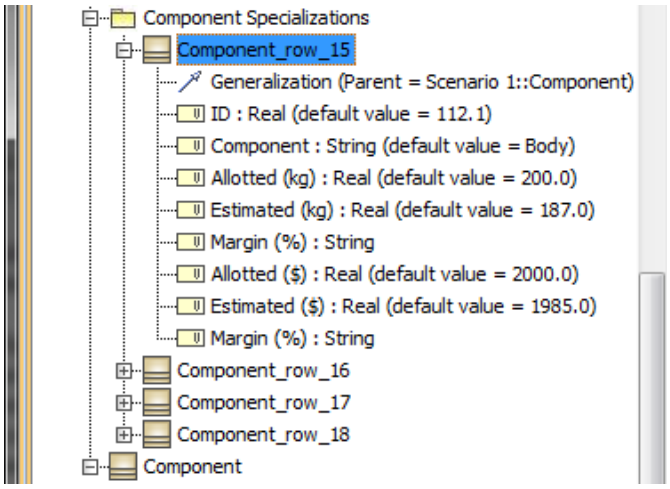


Figure 4 Specializations of Component from excel rows

Now we can exercise the power of such data map connections by making changes on one side or the other, and then using Syndeia to compare and update the other end. Consider the situation where new information from a supplier has been received that the Autopilot component has been upgraded, which has caused a change in estimated mass and cost, which has been duly updated in the ComponentLIbrary.xlsx workbook. The systems engineer, either during a routine check, or prompted by the spreadsheet manager, can compare SysML and Excel to see the changes highlighted in the results with a message as to what is different as shown in Figure 5.

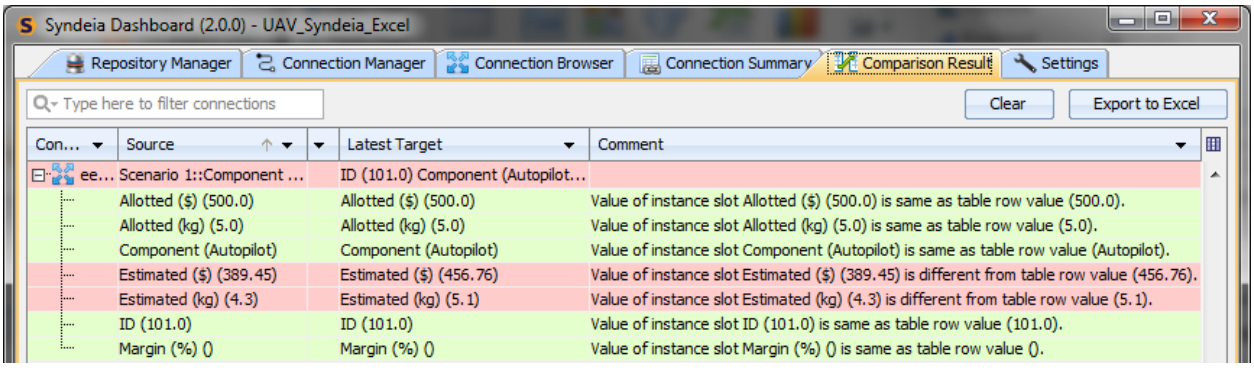


Figure 5 Compare SysML and Excel for a row with two values changed in Excel

Figure 6 shows the Syndeia Connection Browser before syncing values from Excel to SysML, and Figure 7 shows the instance in the MagicDraw containment tree after syncing. At this point, any parametric analyses that use these values (such as those that will be shown in Scenario 3) will be performed again.

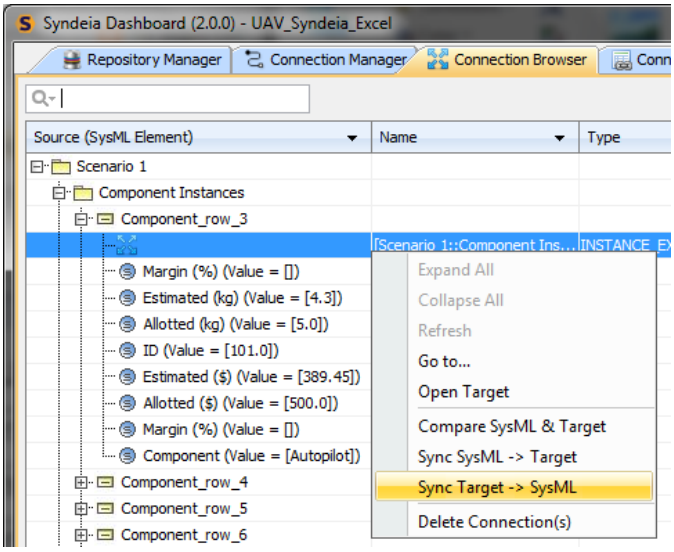


Figure 6 Instance values before syncing Target (Excel) -> SysML

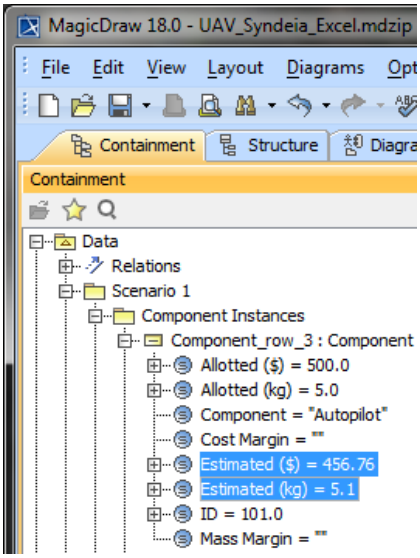


Figure 7 Instance after syncing Target (Excel) -> SysML

Now consider the case where the systems engineer changes some values, perhaps by calculating the margins using parametrics, and then wishes to compare and sync them to the excel workbook ComponentLibrary.xlsx. The steps are the same as above, except reversed in direction. The system engineer updates the values of Cost Margin and Mass Margin of a component in the SysML model, and then uses Syndeia to compare with the Excel spreadsheet, which may have also changed as is often the case in concurrent development. The comparison results are shown in Figure 8.

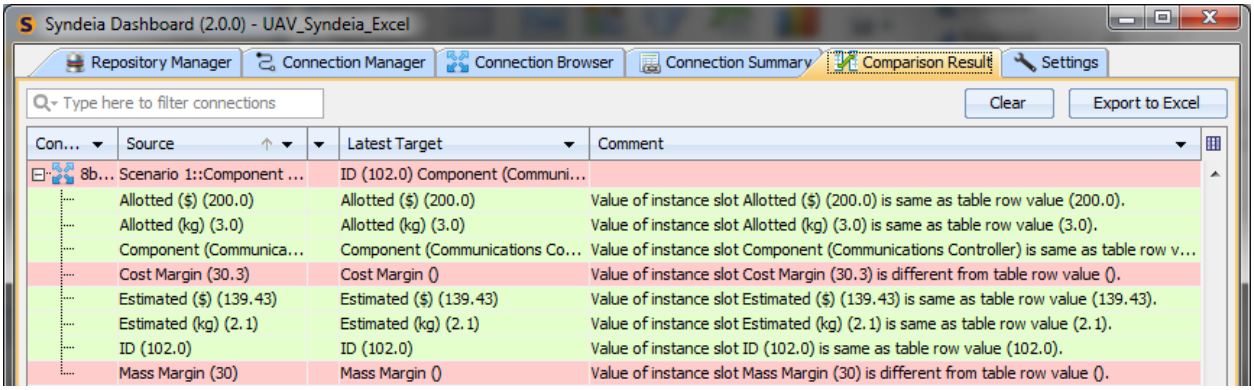


Figure 8 Compare SysML and Excel for a row with two values changed in SysML

The system engineer uses Syndeia to sync from source (SysML) to target (Excel). The Excel table after syncing new margin values is shown in Figure 9. When syncing SysML -> Excel, the excel workbook needs to be closed so that new values may be written, and then it may be opened again to view the updated results.

	A	B	C	D	E	F	G	H
1				Mass (kg)			Cost ($)	
2	ID	Component	Allotted (kg)	Estimated (kg)	Mass Margin	Allotted ($)	Estimated ($)	Cost Margin
3	101	Autopilot	5.0	5.1		$500.00	$456.76	
4	102	Communications Controller	3.0	2.1	30	$200.00	$139.43	30.3
5	103	Data Bus	6.0	3.3		$600.00	$557.32	

Figure 9 Excel table row after syncing SysML -> Target

New rows may also be added to the spreadsheet, corresponding to new components. As long as the connection is maintained between the Component block and the Component table, these new rows can be dragged from Excel to SysML as needed to create new instances or specializations

Scenario 2 – Generating a table in Excel from blocks and instances or specializations in SysML

Our second scenario begins with a Component block in SysML with value properties of either type Real or String, as shown at the top of the BDD in Figure 10, specializations of this block that have the same value properties redefined with default values added where applicable, as shown in the middle of the BDD, and instances of the same block with some slot values populated, as shown at the bottom of the BDD.

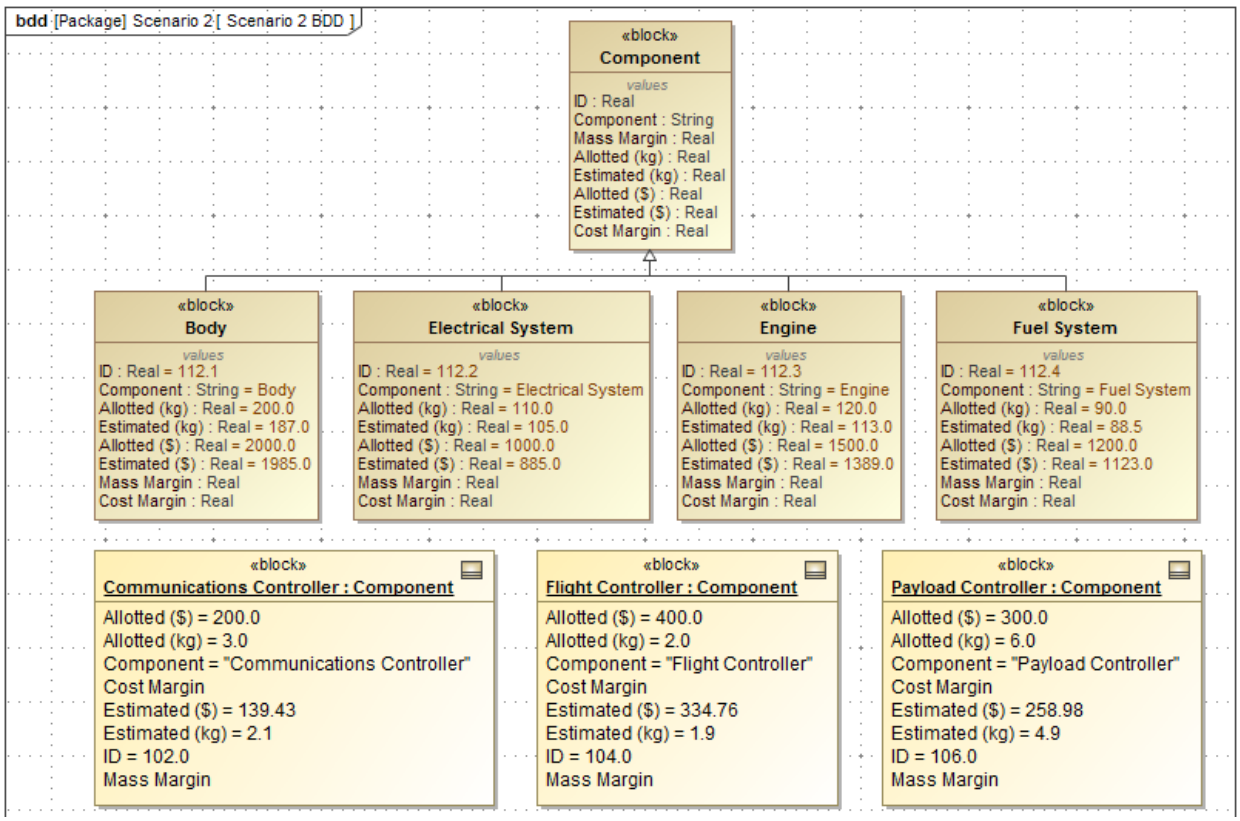


Figure 10 Component block with block specializations and instances for generation in Excel

Making sure the Excel workbook is closed before attempting to make changes through Syndeia, the Component block can be dragged to an empty sheet of the workbook (Library from SysML), creating a new named reference (specifically a table) in Excel as shown in Figure 11. Other options include: (1)

Dragging the block to a workbook will create a new sheet and a table in that sheet, and (2) Dragging a block to a folder will provide the option to create a workbook, sheet, and table.

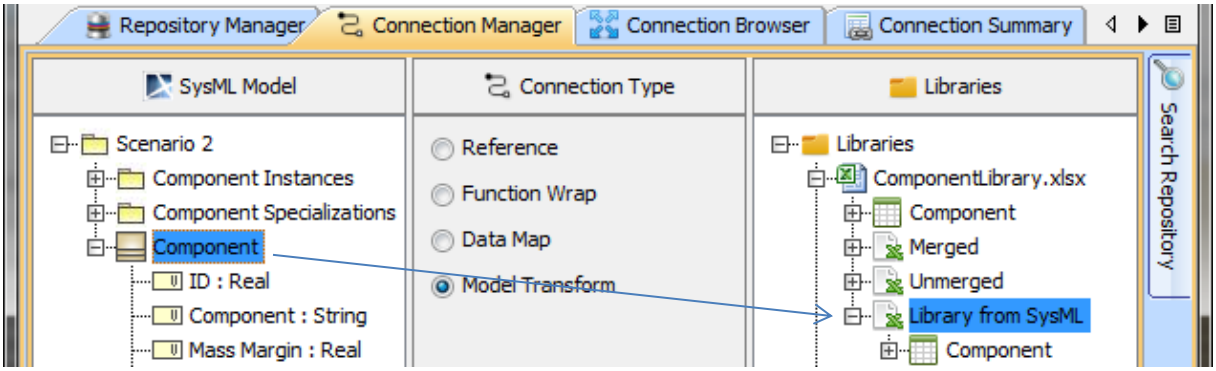


Figure 11 Creating component table in Excel from Component block in SysML

From here, selecting Data Map connection type in Syndeia, we can drag both instances and block specializations to this new table. This will create a new table row in Excel that is connected to the corresponding block instance or specialization for downstream sync, as shown in Figure 12.

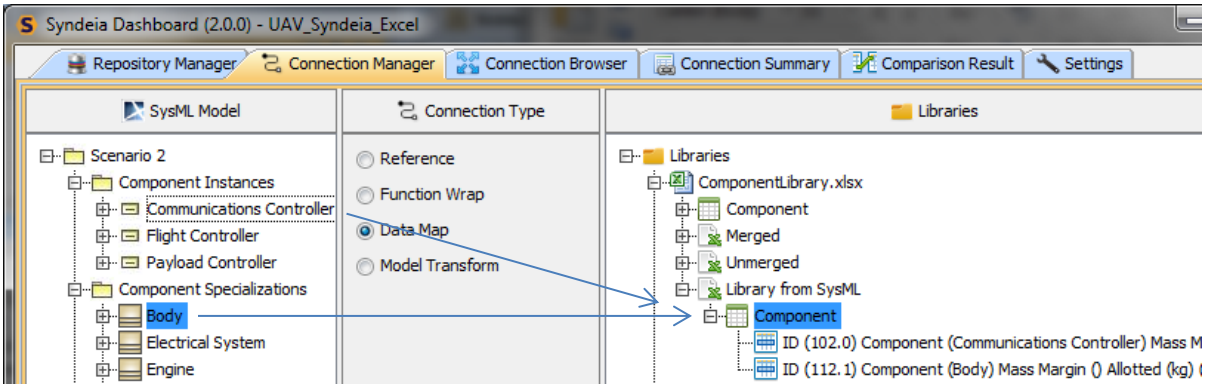


Figure 12 Creating table rows in Excel from instances and block specializations in SysML

The connections created by dragging SysML -> Excel are the same as connections created by dragging Excel -> SysML, so the same compare and sync operations shown in Scenario 1 may be repeated for Scenario 2.

Scenario 3 – Mass and Cost rollups for components taken from Excel library in Scenario 1

In this last scenario, we will build on the Component block and instances created in Scenario 1 to run some simple mass and cost rollup calculations. First of all, we create a UAV System block and add a directed composition from UAV System to Component block to create a *component* part property, and set the multiplicity of this property as one-to-many (1..*). Then, we create an instance of the UAV System (UAV config 1) and link it with the Component block instances generated from Excel in Scenario 1. Both the block and instance structure may be seen in Figure 13.

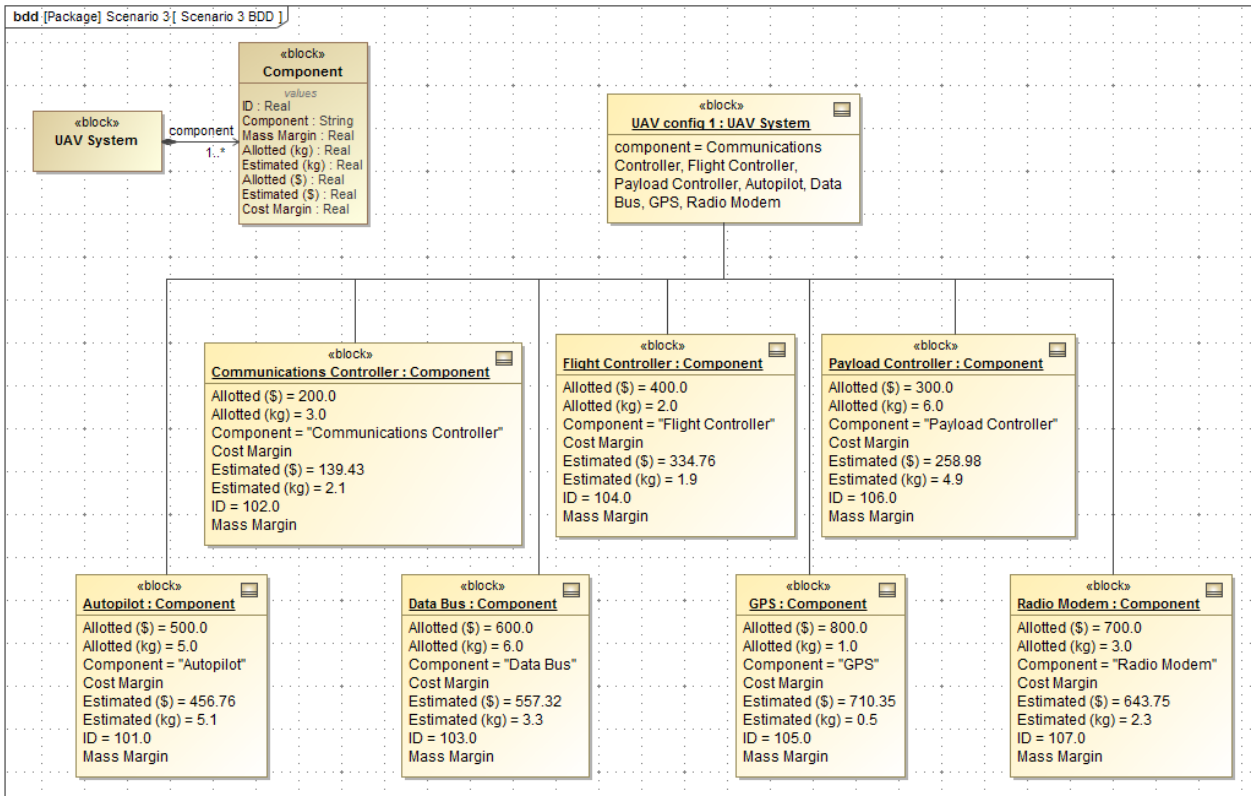


Figure 13 Enhanced block and instance structure with a top-level UAV System block and UAV config 1 instance

Next we create a parametric diagram for the UAV System block, which uses the component part property and the mass and cost value properties of the Component block (Figure 14). Value properties are created in the UAV System block to contain the rolled-up values. Then we use the *Rollup* constraint block with equation "y = sum(x)" four times to roll up allotted and estimated cost and mass.

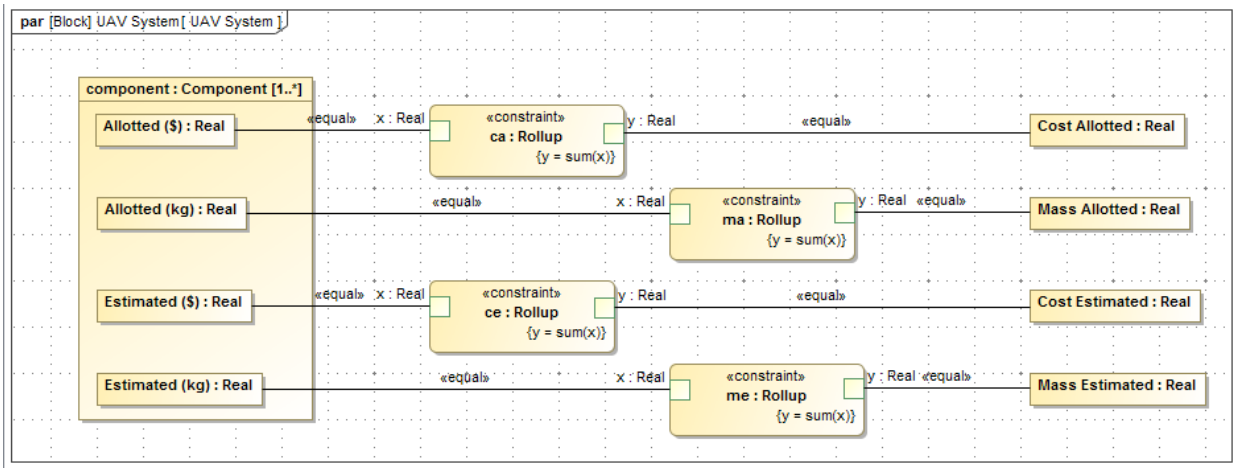


Figure 14 Parametric diagram for the UAV System block, showing rollup constraint properties

From here we simply browse the UAV config 1 instance in ParaMagic, set the Allotted and Estimated Cost and Mass values as targets for solving, and solve, as shown in Figure 15.

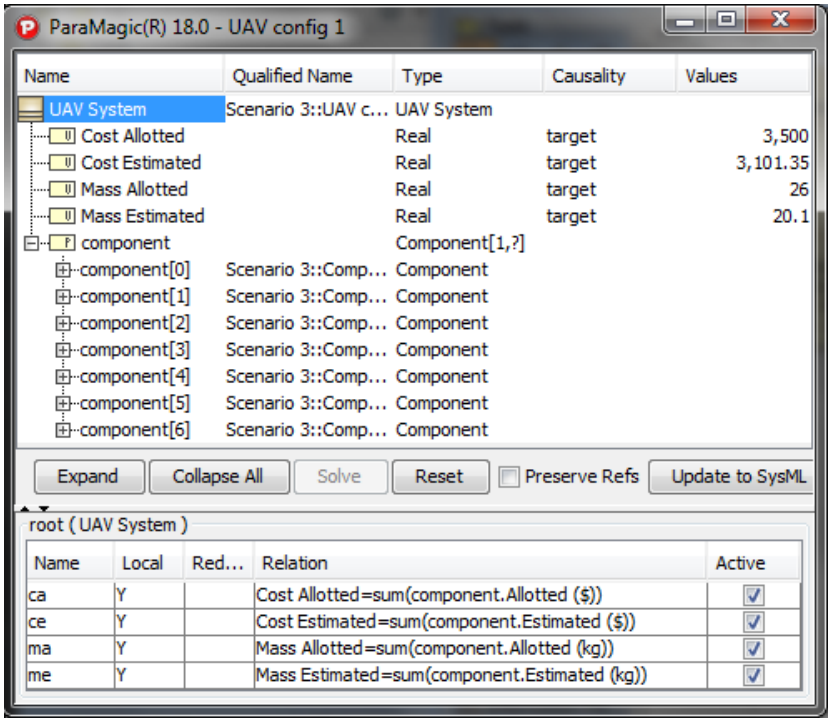


Figure 15 Solutions for cost and mass rollups using ParaMagic

These same general approaches can be extended to additional roll-up calculations, including additional layers of inheritance and multiple inheritance. It is important to evaluate the parametric solver tool capabilities to handle inheritance, recursion and complex aggregates for the class of problems the modeler needs to solve.

Summary

The intent of this technical note has been to demonstrate three basic scenarios for integrating Excel tables with SysML models using the Syndeia and ParaMagic plugins for MagicDraw & SysML. In the first scenario, Excel tables and their rows were used to generate SysML blocks and instances while simultaneously creating connections between the two. Then, changes were made to values both in rows of the Excel table and the block specializations / instances and synced in both directions. In the second scenario, the reverse was demonstrated, where SysML blocks and instances were used to generate a table in Excel. In the third and final scenario, the instances generated from Excel rows in the first scenario were used to perform mass and cost rollup calculations using SysML parametrics and the ParaMagic solver plugin. This Technote has demonstrated the ability to connect equipment lists in Excel and systems engineering models in SysML, and ensure that values of key parameters such as mass and cost stay in sync between different software environments during concurrent development.

About the Author

Rose Yntema (rose.yntema@intercax.com) is Applications Engineer for Intercax LLC, Atlanta, GA, where she applies MBSE techniques to complex systems in areas such as aerospace, energy, defense, and telecommunications. She is actively involved in the development of SysML parametric modeling and simulation software. Rose earned her M.S. (2012) in Electrical and Computer Engineering from the Georgia Institute of Technology, and Sc.B. (2010) in Electrical Engineering from Brown University.
For further information, visit us at www.intercax.com or contact us at info@intercax.com.