

To Do Better MBSE with Enterprise Architect, What Do We Need?

Takeshi Kouno
Sparx Systems Japan

Conclusion: We Need Two Keys to Success in MBSE with Enterprise Architect and SysML

- Create and maintain models efficiently
- Detect inconsistencies and errors during modelling

But how?

With my free SysML add-in!

Introduce Myself

Takeshi Kouno

- CEO and Founder of Sparx Systems Japan (2003-)
- 'Takeshi K' in the Sparx Systems Forum (<https://sparxsystems.com/forums/>)



Takeshi K
EA User
■■
Posts: 571
Karma: +35/-1

More Efficient Modeling Example in MBSE using SysML

When I decompose an Action in a system-level Activity diagram:

1. Create an Activity for a subsystem Action.
2. Drag it onto the diagram as a Call Behavior Action.
3. Switch the connected flows to the new action.
4. Delete the old Action from model.
5. Create the Activity diagram. (then add Partitions and Actions)

How we decompose, by using two Enterprise Architect:

- **Blue: original Enterprise Architect**
- **Green: Enterprise Architect + the SysML Add-in**

The SysML Add-in

You can download and install from the following webpage.

<https://www.sparxsystems.jp/en/SysML/>

or

<https://bit.ly/easysml>

You can use as free!

(You can support the add-in development by purchasing the 'Supporters edition', but the features are basically the same.)

Current Situation in Japan

- MBSE with SysML is a major topic of discussion, and SysML modelling tools for MBSE are also of significant interest.
- Two points to choose a tool:
 - not only features (what we can do with each tool)
 - but also usability (how many steps we can do for each operation)

This is why I have offered the SysML add-in for Japanese users for several years.

- more useful features
- more efficient modelling

Demo: Typical Operations in MBSE

I will have a demo with following diagrams in the MagicGrid framework model.

1. Requirement diagram
2. Block Definition diagram
3. Interface definition
4. Internal Block diagram

Useful Features for SysML

- More drag and drop for typical SysML modelling
- Dedicated properties dialogs for SysML main elements
- Consistency based suggestion for conveyed items
- Various types of search rules with Excel export
- Smart Auto Layout for ports and connectors

Useful Features for SysML (cont.)

- Other 'small' nice features for typical SysML modelling
 - reverse Containment (Nesting) for Requirements
 - set default compartments visibility
 - generate a Block or an Activity from instances
 - automatic Port adding
 - synchronize type of Ports and Action Pins
 - apply Primitive type Value Types if they are in a model
 - and much more...

The SysML Add-in (same as page 5)

You can download and install from the following webpage.

<https://www.sparxsystems.jp/en/SysML/>

or

<https://bit.ly/easysml>

You can use as free!

(You can support the add-in development by purchasing the 'Supporters edition', but the features are basically the same.)

Conclusion: Two Keys to Success in MBSE with Enterprise Architect and SysML

- Create and maintain models efficiently
 - automatic ID generation for Requirements
 - automatically placed Partitions
 - creating ports etc. by drag and drop
 - better routing for connectors
- Detect inconsistencies and errors during modelling
 - selectable types in the SysML specification
 - selectable items for conveyed items
 - coloring conveyed items
 - checking by different views - e.g. the ICD (Interface Control Document)

We Need More Features for Large and Complex Model!

More advanced features for MBSE are offered as the Traceability Suite add-in. (paid add-in)

<https://www.sparxsystems.jp/en/trace/>

Example: MatrixEx Features for MBSE

The screenshot displays the MatrixEx MBSE tool interface. It features a 'Source' tree on the left, a 'Target' tree at the top, and a central matrix table. Annotations highlight specific features: 'Two or more Types' points to the SysML Derive/Refine icons; 'Two or more Packages' points to the package headers; 'Hide any objects from Matrix' points to the hide icon; 'Implied Relationships' points to the upward arrow icon; 'Show Req ID + Name' points to the requirement name in the source tree; 'Relationship Counter' points to the counter icon; and 'Zoom by mouse wheel' points to the zoom icon.

Source	1 Subsystem Requirements	2 System Structure
1 Implementation Requirements	2	25
PR-1 Implementation Requirements Specification	2	25
PR-1.1 Electronics & Electric	1	0
PR-1.1.1 Electric Relays	1	0
PR-1.2 Mechanics	0	0
PR-1.2.1 Refrigerant Pipe	0	0
PR-1.2.1.1 Pipe Diameter	0	0
PR-1.2.1.1 Pipe Material	0	0
PR-1.3.1 Refrigerant	0	0
PR-1.4 Generic Characteristics	1	0
PR-1.4.1 Max Energy Consumption by ...	1	0

Thank You!

I hope that this session will provide you with some hints on how to make better use of Enterprise Architect.

If you have any questions, comments and/or ideas, please let me know or send a message:

- I will join tomorrow as an attendee, feel free to talk to me directly today or tomorrow
- Email: tkouno@sparxsystems.jp
- LinkedIn: takeshikouno