Deployment of SysML in Tools and Architectures: an Industry Perspective

Rick Steiner
Raytheon IDS, San Diego
fsteiner@raytheon.com
4 Pillars of SysML – ABS Example

1. Structure

2. Behavior

3. Requirements

4. Parametrics
Cross Connecting Model Elements

1. Structure

2. Behavior

3. Requirements

4. Parametrics
Key Considerations for SysML Tool Selection

- The specific MBSE method employed may leverage specific SysML features, but may not require other features. It is appropriate to ask the following questions to emphasize the features of SysML that a successful tool deployment will need to support.
  - Which behavior representations are most important? Activity diagrams? State machines? Sequence diagrams?
  - Will there be a need for item flow representation?
  - What kind of need will there be for detailed performance analysis and parametric modeling? Expression of mathematical equations relating parameters of system elements may be a very important part of the system development process/method employed.
  - Will there be a need for algorithm specification & development? It may be important to express information processing algorithms explicitly in mathematical form, using constraint blocks and eventually relating them to specific blocks representing software code.
  - Which architecting principles need to be supported by the tool?
  - How will allocation be used? The manner in which allocation is used to guide the development process may dictate a set of constraints & rules associated with allocation relationships. By enforcing or enabling these rules, a toolset can improve the efficiency of the modeling process.
Water Distiller Example

- Functional Analysis based, not OOA
  - Relies heavily on activity diagrams and functional allocation
- Solution to problem focused on activity modeling, flow allocation, item flows & parametrics
  - Heat balance of distiller relies on properties of water flowing through system
- Traditional UML tools just don’t do these things
Tool Comparison
For Distiller Example

- No tool “fully” implements SysML
- Clearly, each tool has strengths & weaknesses
  – Make sure tool is compatible with your method
- Other tools exist, but not evaluated
- RS(X) is tool I’m least familiar with

<table>
<thead>
<tr>
<th>Feature</th>
<th>Enterprise Architect ver 7.1</th>
<th>Magic Draw ver 15.1</th>
<th>Rhapsody ver 7.2</th>
<th>RS(X) ver 7.0.5</th>
<th>E+ SysML ver 2.0.5.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Modeling</td>
<td>full</td>
<td>full</td>
<td>limited</td>
<td>full</td>
<td>full</td>
</tr>
<tr>
<td>Structural Modeling</td>
<td>full</td>
<td>full</td>
<td>full</td>
<td>full</td>
<td>full</td>
</tr>
<tr>
<td>Item Flows</td>
<td>limited</td>
<td>full</td>
<td>full</td>
<td>limited</td>
<td>limited</td>
</tr>
<tr>
<td>Ports/Interfaces</td>
<td>full</td>
<td>limited</td>
<td>full</td>
<td>full</td>
<td>full</td>
</tr>
<tr>
<td>Functional Allocation</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Flow Allocation</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Parametrics</td>
<td>full</td>
<td>full</td>
<td>full</td>
<td>full</td>
<td>full</td>
</tr>
<tr>
<td>Code Gen/Animation</td>
<td>none</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Requirements</td>
<td>full</td>
<td>full</td>
<td>full</td>
<td>full</td>
<td>full</td>
</tr>
<tr>
<td>Distiller Model Source</td>
<td>Steiner</td>
<td>Steiner</td>
<td>Lussier</td>
<td>Steiner</td>
<td>Steiner</td>
</tr>
<tr>
<td>UML4SysML 2.1</td>
<td>most</td>
<td>all</td>
<td>most</td>
<td>most</td>
<td>most</td>
</tr>
</tbody>
</table>
Allocate activity partitions work well, allocation tables are fast & easy
Flow allocation not possible (object flow to item flow)
Allocate activity partitions work
Flow allocation works
Flexible tabular view
- Action nodes do not invoke activities (no activity hierarchy)
- No activity parameter nodes (on diagram frame, or otherwise)
- Action pin notation is awkward, pins not reused when action referenced
- Can’t distinguish control flow from object flow
- Tabular view & reports of allocation are available
Non-standard diagram frame/label
No unique action names (must be same name as activity), but allocation is unique
Allocation partitions work (automatically create allocation relationships) to blocks or parts.
- Allocation works, but compartments not supported
- Can’t access value properties of item properties (e.g. temp of water into Heat Exchanger) -> can’t do parametric analysis of distiller example.
- Diagram frame uses incorrect nomenclature
- Allocation compartment incorrect format
- DOES allow full access to item properties
- Item flows and item properties fully allocable
  - Item flows look weird, but work fine
  - ObjectFlows can’t be allocated, but ObjectNodes can.
- Full allocation compartments & callouts
- ItemFlows incorporated in RSD 7.0.5/E+ 2.0.5.1, but
  - no icon or name/ItemProperty on diagram, ItemFlow not associated with Connector
- Non-standard diagram frame/label
- Allows Allocation of ObjectFlow to ItemProperty, but not to ItemFlow
  - no allocation compartment/callouts on parts
- Item properties, value types, units and dimensions fully supported
Rhapsody Parametric Diagram

- Item properties, value types, units & dimensions fully supported
EA & RS(X)/E+ Parametrics

- Rendering issues on cut & paste

![Diagram of Isobaric Heat Balance](image-url)
EA & RS(X)/E+ Parametrics

- Both support units, dimensions, value types, constraint blocks, and parametric diagrams
- Neither support value properties of item properties on item flows
  - Item Flows incorporated in RSD 7.0.5/E+ 2.0.5.1
SysML Diagrams—a Method for Model Integration

- 3 separate hierarchies of Structure, Behavior, and Data
  - Usage (internal connection) is documented with separate diagrams
- These 3 hierarchies maintained at Operational and System level

<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>Usage</th>
<th>Cross-Connect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>bdd</td>
<td>act (swimlane), seq (lifeline, op)</td>
</tr>
<tr>
<td>Behavior</td>
<td>bdd</td>
<td>ibd (itemFlow), seq (msgType)</td>
</tr>
<tr>
<td>Data</td>
<td>bdd</td>
<td>act (objFlow), seq (msg,op), stm</td>
</tr>
</tbody>
</table>

bdd = Block Definition Diagram (no DoDAF)
ibd = Internal Block Diagram (OV-2, SV-1, SV-2)
act = Activity Diagram (OV-5, SV-4)
seq = Sequence Diagram (OV-6c, SV-10c)
stm = State Machine Diagram (OV-6b, SV-10b)
DoDAF Views Horizontally Cross-Connecting a Complex SoS Model

I. Operational
   Command, OpNode

II. System
   IIA. Conceptual
       Multi-Node System

   IIB. Logical
       Generic Systems (C2, Sensor…)

   IIC. As-Is
       POR 1
       POR 2
       POR 3

   IID. To-Be
       Future System/Standard 1
       Future System/Standard 2

Triangles represent hierarchy diagrams (no DoDAF equivalent)
Allocation Vertically Cross-Connecting a Complex SoS Model

I. Operational
   Command, OpNode

II. System
   IIA. Conceptual
      Multi-Node System
   IIB. Logical
      Generic Systems (C2, Sensor…)
   IIC. As-Is
      POR 1
      POR 2
      POR 3
   IID. To-Be
      Future System/Standard 1
      Future System/Standard 2

Triangles represent hierarchy diagrams (no DoDAF equivalent)