Modeling for Systems Engineering: How and when do we need it

Laurent Balmelli, Ph.D.
Guest Professor, Keio University, Tokyo, Japan
Director, Sodius Corp.
Author presentation

Laurent Balmelli, M.S. & Ph.D. (EPFL)

• From 2000 to 2012 - 12 years of experience in Systems Engineering as
  – Manager at IBM Software Group Japan (2006-2009)
  – Member of IBM US CTO team for Systems (2009-2012)
  Focusing on the A&D and automotive industries

• Since 2006, guest professor at Keio University, in Tokyo Japan
  – Teach a class on model-driven systems engineering every year

• From 2012 to present
  – Founder of the Sana Elias Group (with my wife)
  – Lecturer on product design innovation at HEIG-VD (Innokick Master)
  – Lecturer at the EPFL Master of Technology on products and services design
  – Advisor to technology companies
What types of models are we dealing with?

- Visual model are a graphical representations of formal semantics.
  A bit of technicalities….
  - The model semantics are represented by a meta-model (i.e. a model of model)
  - The model is an instance of the meta-model
  - The graphic representation is defined along with the meta-model definition

- Examples of visual models are: the Business Process Modeling Language (BPML), the Unified Modeling Language (UML), the Systems Modeling Language (SysML), Integrated Architecture Frameworks, etc.
Visual modeling has penetrated many industries, in particular during the last 15 years.

- In business process modeling, BPML is a means to formalize the operations of companies in order to optimize them. Processes such as procurement, customer service, product design, etc. are frequently modeled.

- In software engineering, modeling is a means to design software architectures before starting coding activities. Class and library skeletons can be generated from the model.

- In defense, Integrated Architecture Framework are used to represent entire organization and their functioning. The models are frequently used to assess the impact of acquiring new resources or developing new operational capabilities.

- In product design, modeling is a means to create a multi-disciplinary blueprint of products in order to harmonize the coordinated effort of a diverse team of engineers.

Across industries, modeling is a means to formalize development and operational activities in a business setting.
The defense industry and the use of Integrated Architecture Frameworks

- Integrated Architecture Frameworks are the center of modeling activities in the defense industry.
- They allow the description of the defense organizations with a common (visual) language, in addition non-visual (e.g. tables) entities.
- The purpose is to allow a rigorous and consistent linkage of complex information, as well as normalized means for describing systems.
- Business goals include:
  - Ensure that the next acquisitions properly inter-operate with existing resources and operations
  - Derive an understanding of 2nd and 3rd order effect with existing systems when acquiring a new entity (i.e. impact analysis)
Figure: NCV-2 “Grand Maneuver with Enterprise”
**Left:** model (concrete syntax), **right:** meta-model
What are the business goals when using models?

• The (correct) use of models is leveraged as a business process improvement mechanism with the following goals:

  1. **Documentation mechanism**: models of business process, product and service structure, organization structure, etc. are better suited to analysis and efficient information gathering.

  2. **Decision mechanism**: models represent an abstracted view of complex information, and allow decision making based on a simplified setting. For example: trade-off analysis, impact analysis, etc.

  3. **Integration mechanism**: in the field of product development, models can be used as a multi-disciplinary representation of design, i.e. mixing multiple engineering disciplines in the single combination of entities.

  4. **Simulation mechanism**: models representing business processes or products can be simulated to assess functional (operational) behavior or intrinsic properties, for example physical.

  5. **Collaboration mechanism**: models can be used as a common vocabulary to communicate across development stages, across organizations, across stakeholders, across the value chain, etc.

  6. **Traceability mechanism**: models can be used to store relationships between artifacts (e.g. used across the development lifecycle) and allow users to characterize their inter-relationships with strong semantics.
Foster decision-making ability at early stages of the development

- In product development, the availability of relevant information at early stages enables better decision-making eventually leading to shorter development cycles (and shorter time-to-market)
- The (art of) modeling approach provides a semantic framework for information gathering (e.g. based on a meta-model) and allows for computer processing and analysis
  - For example, behavior modeling allows designers to test the functioning of products before they are built
  - Business process simulation allows managers to validate operational assumptions before their implementation in the organization.
  - In Defense, an architecture model of an organization allows managers to assess the impact of a new policy by considering the instantiated entities in the model.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Analysis/Design</th>
<th>Development</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present</td>
<td>3%</td>
<td>27%</td>
<td>55%</td>
</tr>
<tr>
<td>Leading Practice</td>
<td>20%</td>
<td>13%</td>
<td>22%</td>
</tr>
</tbody>
</table>

University of West Virginia and The United States Air Force Academy 2001

Time saving

30-50%
In Systems Engineering, the use of a system model provides a repository for the semantic description of development artifacts across engineering disciplines.

- Each “box” in the system model is attached to an artifact or set of artifacts in an engineering discipline.
- Each relation in the system model represents a dependency between the connected artifacts that is relevant (by definition) at the system level. In other words, it is a dependency between the designated engineering disciplines.
The use (and re-use) of models allows designers to test product assemblies to assess parameter values (e.g. measures of merit)

- The scope and accuracy of early stage analysis depends on the granularity of information embedded in the model
- Trade-off analysis can be performed in the computer-assisted modeling environment and include access to proprietary simulations dedicated to assess key aspects of design.
Effective systems engineering in five scenarios

Five tenets for deploying effective systems engineering processes:

1. An efficient infrastructure of systems engineering allows for orchestrating activities across disciplines, such as *(integrated)* Design Change Management requests.

2. Models for systems engineering *(requirements, architecture, etc)* ought to be accessible across organizational boundaries, e.g. to gather all stakeholder feedback.

3. Models of multi-disciplinary products combining hardware and software ought to provide a mechanism for simulation allowing early validation of design trades.

4. Models of product requirements ought to be shared across the supply chain to allow efficient inter-organizational product design.

5. System models ought to provide a mechanism to deploy design information across disciplines, for example using model transformation.
Scenario 1: Coordinating change management across ALM/PLM boundaries

An effective systems engineering infrastructure satisfies the following requirements

- Tools used across development gain access to relevant datasets via links managed by protocols built on open standards such as the Open Standard for Lifecycle Management (OSLC)
- In turn, the management of linked artifacts across the development allows for the creation of inter-disciplinary dependencies
- The ability to effect the coordinated change of hardware and software components is a basic requirement of modern systems engineering infrastructures

![Diagram showing coordination across ALM/PLM boundaries]
Demo  Integrated change request management
Scenario 2: Model publication for review

- Proving access to model information across organizational boundaries allows managers to gather efficient stakeholder feedback
  - Model publishing and review is a powerful collaborative capability that fosters the efficacy of the development process.
Demo Collaborative process for model review and feedback
Scenario 3: Transformation from SysML to SystemC

- By using the modeling language SysML as a modeling formalism, we allow designers to explore system-level design space tradeoffs
  - SystemC code generation to from IBM Rhapsody enables a model-based code development flow for HW/SW co-design
Demo Code generation from SysML model for hardware simulation
Scenario 4: Requirement synchronization across the supply chain

- In the product development industry, OEMs and suppliers are continuously exchanging requirements
  - Requirement management tools and business processes vary considerably between the business units of the same organization, as well as between organizations.
  - An efficient requirement model exchange mechanism is necessary to enable efficient collaborative processes
Web-services data synchronization

- Stakeholder can define an inter-company collaborative process based on web-standards.
DOORS to Integrity exchange platform (1)
DOOR to Integrity exchange platform (2)
DOOR to Integrity exchange platform

• A web application manages all synchronization tasks, as specified in the collaboration process between the two or more organization on a dedicated server.
Example: DOORS input data screen
Example: Exchange result screen
**Scenario 5: Systems engineering to software engineering**

- The system model is leveraged across stages of the development as a means to deploy design information
  - Mechanism to automate the transition from systems modeling to software modeling
  - Systems engineering artifacts are linked to software models
  - Model transformation between the system model and the software model can be done incrementally: previously transformed artifacts are updated and a change report is generated

---

**System Architect**

**Systems Engineering Models**

**SE models are viewed as «requirement» models for software artifacts**

**UML Software Models**

- Dependency
- Read-only

---

9/1/2015
Example: DoDAF model to UML model

System Architect
SE Models

OV2, OV5, OV7, SV11, SV1, SV2, SV4

UML
Software Models
How to achieve traceability between systems engineering and engineering disciplines

- Incremental update of systems engineering models can be achieved, producing traceability and change report for software teams.
Contact

Laurent Balmelli, Ph.D.
Director
Strategic Business Development
+41 76 348 38 61
lbalmelli@sodius.com

accelerating engineering processes