Are we Ready to Unlash the Full Potential of Future Embedded Systems?

Annual Conference of the UTC Institute for Advanced Systems Engineering, University of Connecticut, Monday November 10, 2014

Holger Giese
System Analysis & Modeling Group,
Hasso Plattner Institute for Software Systems Engineering
University of Potsdam, Germany
holger.giese@hpi.uni-potsdam.de
Outline

1. Challenges Ahead
2. Available Options
3. Conclusions & Outlook
Outline

1. Challenges Ahead
   - Cyber-Physical Systems
   - System of Systems
   - Ultra-Large-Scale Systems

2. Available Options

3. Conclusions & Outlook
Envisioned Challenges for Future Embedded Systems

[Networked] Cyber-Pyhsical Systems

Smart Factory - E.g. Industry 4.0

Smart Logistic

Micro Grids

Internet of Things

Smart City

System of Systems

Ultra-Large-Scale Systems

Smart Home

E-Health

Ambient Assisted Living

RailCab Example: A Short Video ...

http://www.railcab.de/

Test shuttle

Test track

A shuttle system that builds convoys to optimize the energy consumption
A Selection of Critical Future Challenges

- **Operational and managerial independence**
  - operated independent from each other without global coordination
  - no centralized management decisions (may be conflicting)

- **Dynamic architecture and openness**
  - must be able to dynamically adapt/absorb structural deviations
  - subsystems may join or leave over time in a not pre-planned manner

- **Advanced adaptation**
- **Integration**
- **Resilience**
Challenge: Operational and Managerial Independence

“A system-of-systems is an assemblage of components which individually may be regarded as systems, and which possesses two additional properties:

- **Operational Independence** of the Components: If the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. That is, the components fulfill customer-operator purposes on their own.

- **Managerial Independence** of the Components: The component systems not only can operate independently, they do operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system-of-systems.”

[Maier1998]
“The sheer scale of ULS systems will change everything. ULS systems will necessarily be decentralized in a variety of ways, developed and used by a wide variety of stakeholders with conflicting needs, evolving continuously, and constructed from heterogeneous parts.”

“The vision of Cyber-Physical System (CPS) is that of open, ubiquitous systems of coordinated computing and physical elements which interactively adapt to their context, are capable of learning, dynamically and automatically reconfigure themselves and cooperate with other CPS (resulting in a compound CPS), possess an adequate man-machine interface, and fulfill stringent safety, security and private data protection regulations.”

Required capabilities:
- must be able to dynamically adapt/absorb structural deviations
- systems may join/leave over time in a not pre-planned manner

[Northrop+2006]
[Broy+2012]
“**Adaptation** is needed to compensate for changes in the mission requirements [...] and operating environments [...]”

[Northrop+2006]

“The vision of Cyber-Physical System (CPS) is that of open, ubiquitous systems of coordinated computing and physical elements which interactively adapt to their context, are capable of learning, dynamically and automatically reconfigure themselves and cooperate with other CPS (resulting in a compound CPS), possess an adequate man-machine interface, and fulfill stringent safety, security and private data protection regulations.”

[Broy+2012]

**Required kind of adaptation:**
- System level adaptation
- System-of-systems level adaptation
Challenge: Integration (1/2)

Model Integration?

- Problem to integrate models within one layer as different models of computation are employed
- Leaky abstractions are caused by lack of composability across system layers. Consequences:
  - intractable interactions
  - unpredictable system level behavior
  - full-system verification does not scale

[Sztipanovits2011]
Cross-Domain Integration:

Example: A convoy of fully autonomous cars abandons the premium track in order to give way to an ambulance (intersection of CPS specific for traffic and health care)

CPS of different domains have to be connected:

- According to social and spatial network topologies, CPS operate across different nested spheres of uncertainty
- CPS dedicated to different domains have to interact and coordinate.

[Broy+2012]
Challenge: Resilience

“The vision of Cyber-Physical System (CPS) is that of open, ubiquitous systems [...] which [...] and fulfill stringent safety, security and private data protection regulations.” [Broy+2012]

“Resilience[:] This area is the attribute of a system, in this case a SoS that makes it less likely to experience failure and more likely to recover from a major disruption.” [Valerdi+2008]

“Resilience is the capability of a system with specific characteristics before, during and after a disruption to absorb the disruption, recover to an acceptable level of performance, and sustain that level for an acceptable period of time.”

Required coverage of resilience:

- Physical and control elements (via layers of idealization)
- Software elements (via layers of abstraction)
- Horizontal and vertical composition of layers

Resilient Systems Working Group, INCOSE
Outline

1. Challenges Ahead

2. Available Options
   - Service-Oriented Architecture
   - Self-Adaptive & Self-Organization
   - Multi-Paradigm Modeling

3. Conclusions & Outlook
Service-Oriented Architecture:
- Dedicated services are offered by systems via defined service contracts can be offered, looked up, and bound at run-time
- Interoperability is provided by a service bus

Service oriented architecture Modeling Language (SoaML):
- a UML profile for modeling
- Support collaborations as first class elements (service contracts)
- Links collaborations with component-based models
### Observations:

- Service contracts permit to realize **operational and managerial independence**
- Offering, look up, and bin service and runtime supports **dynamic architectures** and openness (but not modeled)
- Under-specification in the service contracts preserves degrees of freedom for **adaptation** of the components (but not at the level of the collaboration)
- Service contracts can make cross-domain **integration** possible (but also required mapping concepts are not supported)
- No specific support for **resilience**

#### Challenges \ Approaches vs SOA

<table>
<thead>
<tr>
<th>Challenges \ Approaches</th>
<th>SOA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational and managerial independence</td>
<td>✓</td>
</tr>
<tr>
<td>Dynamic architecture and openness</td>
<td>✓</td>
</tr>
<tr>
<td>Advanced adaptation</td>
<td>✓</td>
</tr>
<tr>
<td>Integration</td>
<td>✓</td>
</tr>
<tr>
<td>Resilience</td>
<td>X</td>
</tr>
</tbody>
</table>
Option: Self-Adaptive & Self-Organization

- **Self-Adaptive Systems:**
  - Make systems self-aware, context-aware, and requirements-aware using some form of reflection
  - Enable systems to adjust their structure/behavior accordingly

- **Self-Organization:**
  - The capability of a group of systems to organize their structure/behavior without a central control (emergent behavior)

- **Engineering perspective:**
  - A spectrum from centralized top-down self-adaptation to decentralized bottom-up self-organization with many intermediate forms (e.g. partial hierarchies) exists
Option: Self-Adaptive & Self-Organization

Observations:
- Can **co-exist** with managerial and operational independence as well as dynamic architecture and openness, but both make the problem considerably harder.
- Self-adaptive systems enable advanced adaptation at the **system-level** while self-organization covers the **system-of-systems-level**.
- Cross-domain **integration** is possible (but there is no support for adaptation across the domains).
- While both self-adaptive behavior as well as self-organization can **contribute to resilience**, it also makes the problem considerably harder.

<table>
<thead>
<tr>
<th>Challenges \ Approaches</th>
<th>Self-Adaptive / Self-Org</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational and managerial independence</td>
<td>✓</td>
</tr>
<tr>
<td>Dynamic architecture and openness</td>
<td>✓</td>
</tr>
<tr>
<td>Advanced adaptation</td>
<td>✓</td>
</tr>
<tr>
<td>Integration</td>
<td>✓</td>
</tr>
<tr>
<td>Resilience</td>
<td>?</td>
</tr>
</tbody>
</table>
Multi-Paradigm Modeling:

- Enable to use different domain-specific models with different models of computation for different modeling aspects.
- Can be employed at the system-level to combine all necessary models for a system.
- Can be employed at the system-of-systems-level to combine all necessary models for a system-of-systems.
- Requires that for employed model combinations a suitable semantic integration is known (and supported by the tools).
Option: Multi-Paradigm Modeling

### Challenges \ Approaches vs. Multi-Paradigm Modeling

<table>
<thead>
<tr>
<th>Challenges \ Approaches</th>
<th>Multi-Paradigm Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational and managerial independence</td>
<td>✓</td>
</tr>
<tr>
<td>Dynamic architecture and openness</td>
<td>✗</td>
</tr>
<tr>
<td>Advanced adaptation</td>
<td>✓</td>
</tr>
<tr>
<td>Integration</td>
<td>✓</td>
</tr>
<tr>
<td>Resilience</td>
<td>✓</td>
</tr>
</tbody>
</table>

### Observations:

- Can co-exist with managerial and operational independence as well as advanced adaptation, but both make multi-paradigm modeling considerably harder.
- The multi-paradigm modeling approaches assume a fixed hierarchical structure and therefore do not fit to dynamic architectures and openness (exceptions: [Giese+2011] for a specific case and [Pereira+2013] for a MoC).
- Integration is well supported for the models and also across domains.

---

[Brooks+2008]
1. Challenges Ahead
2. Available Options
3. Conclusions & Outlook
Some Conclusions Concerning the Options

Besides Resilience all challenges can be covered by one of the available options.

However, currently **no general combination** of the three options exists.

Are we Ready to Unlash the Full Potential of Future Embedded Systems? **not yet!**
RailCab Example: Challenges & Options

**MechatronicUML:**
- System level: complex graph of models with strict containment hierarchies
- System of systems level: collaboration of a graph of systems

But for the **MechatronicUML** approach we had to develop tool support
- that integrates existing tools in a particular way [Burmester+2008],
- that allows simulating the highly dynamic models [Giese+2011], and
- that ensure the safety at the system-of-systems level [Giese&Schäfer2013].

**Problem:**
- The high effort is required for each specific approach (limited coverage!)
Some Observations Concerning the Options

- **Service-Oriented Architecture** can be described by a graph of links between the systems that evolve.
- **Self-Adaptive** and **Self-Organization** can be described by a graph of links between the components resp. systems that evolve/reconfigure and in case of reflection most models can be described by such a graph as well.
- **Multi-Paradigm Modeling with evolving structures** can be described by a dynamic graph of models and links between them.

**Open question:** Can we exploit that observation to combine the options?
My thesis: **Dynamically changing graphs** encoding models and their linking would allow to combine Service-Oriented Architecture, Self-Adaptive / Self-Organization, and Multi-Paradigm Modeling with evolving structures and could be the basis for a **new solid foundation** to Unlash the Full Potential of Future Embedded Systems ... 

But
- We first have to develop such a generic **new solid foundation** that allows to capture the relevant dynamic architecture and openness!
- A simple **evolution with only minor extensions** to existing tool chains alone will probably not do the job (what is realistic here?)
- The resilience will become an even harder problem (can we justify the **risks** resulting from the envisioned systems of systems at all?)

*But it will get even worse, if we continue in this direction without any such **new solid foundation** (graph-based or any other one)!!*
### Bibliography

<table>
<thead>
<tr>
<th>Reference</th>
<th>Title</th>
</tr>
</thead>
</table>