Crosscutting CPS Needs in Industry

Janos Sztipanovits
ISIS, Vanderbilt University

“Transportation CPS Workshop”
November 19, 2008
Vienna, Virginia  22182
Outline

- A crosscutting CPS need in industry: System Integration
- Nature and scope of challenges
- Elements of a Science of System Integration
# Trends in Vehicles...

<table>
<thead>
<tr>
<th>Sectors</th>
<th>Goals</th>
</tr>
</thead>
</table>
| **Aerospace** | • Aircraft that fly faster and further on less energy.  
• Air traffic control systems that make more efficient use of airspace. |
| **Automotive** | • Automobiles that are more capable and safer but use less energy.  
• Highways that are safe, higher throughput and energy efficient. |
| **Defense** | • More capable defense systems  
• Better use of networked fleets of autonomous vehicles  
• Integrated, maneuverable, coordinated, energy efficient  
• Resilient to cyber attacks |
### Sectors | Goals
--- | ---
**Aerospace** | • Aircraft that fly faster and further on **less energy**.  
• Air traffic control systems that make more efficient use of airspace.

**Automotive** | • Automobiles that are more capable and safer but use less energy.  
• Highways that are safe, higher throughput and **energy efficient**.

**Defense** | • More capable defense systems  
• Better use of networked fleets of autonomous vehicles  
• Integrated, maneuverable, coordinated, **energy efficient**.  
• Resilient to cyber attacks

---

**Energy Internet: When IT Meets ET**

---

**THOMAS L. FRIEDMAN**

*Hot, Flat, and Crowded*
Networking and Information Technology (NIT) have been increasingly used as *universal system integrator* in human-scale and societal-scale systems.

Functionality and salient system characteristics emerge through the interaction of *networked physical and computational objects*.

Engineered systems turn into **Cyber-Physical Systems (CPS)**.
Why Integration?

NIT delivers unique precision and flexibility in interaction and coordination.

Cyber

- Rich time models
- Precise interactions across highly extended spatial/temporal dimension
- Flexible, dynamic communication mechanisms
- Precise time-variant, nonlinear behavior
- Introspection, learning, reasoning

Integrated CPS

- Elaborate coordination of physical processes
- Hugely increased system size with controllable, stable behavior
- Dynamic, adaptive architectures
- Adaptive, autonomic systems
- Self monitoring, self-healing system architectures and better safety/security guarantees.
Is the Integration Role Crosscutting?

- The share of value of embedded computing/networking components in different industries:

<table>
<thead>
<tr>
<th>Industry</th>
<th>2003</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automotive</td>
<td>52%</td>
<td>56%</td>
</tr>
<tr>
<td>Avionics/Aerospace</td>
<td>52%</td>
<td>54%</td>
</tr>
<tr>
<td>Health/Medical equipment</td>
<td>50%</td>
<td>52%</td>
</tr>
<tr>
<td>Industrial automation</td>
<td>43%</td>
<td>48%</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>56%</td>
<td>58%</td>
</tr>
<tr>
<td>Consumer electronics and Intelligent Homes</td>
<td>60%</td>
<td>62%</td>
</tr>
</tbody>
</table>

Outline

- A crosscutting CPS need in industry: System Integration
- Nature and scope of challenges
- Elements of a Science of System Integration
Dimensions of CPS Integration

Across each design concerns (functional, safety/security, physical platform):

- Components
- Layers
- System of Systems
**Component Integration**

**Functional: E.g.: Dynamics**

- Digital Controller
- D/A S/H
- Power Amp.
- Plant and Sensors

Component Integration Platform (e.g. TTP)

**Software: E.g. Timing**

- Comp-1
- Comp-2
- Comp-3

Component Integration Platform (e.g. TTP)

- Composability and compositionality are key concepts
- Defined for carefully selected properties (dynamics, latency, power,..)
- Decomposed into structure, interaction and behavior

**Challenges:**
- Composition frameworks that guarantee essential properties
- Heterogeneous composition
Multi-Layer System Integration

Roles

Cognitive processes
Social interaction
Command and control

Coordination
Data distribution

Component interactions
Component behaviors
Architecture

Resource management
Scheduling
Separation

Timing/performance
Fault management
Power management

Heat dissipation
Crossover
Radiation effects

Layers

Human Organization

System Operation Layer

SW/Component Layer

OS/Network Layer

HWSYSTEMS Layer

Materials & Devices

Characteristics

- Inter-layer interactions
- Effects propagate across the layers
- Efficiency and optimization drives toward intractability
- Inter-layer relationship:
  - mapping
  - refinement
  - synthesis

Challenges:
  - modeling,
  - constraining
  - composing
System of System Integration

Future Military Systems in the Field

- Heterogeneous CPS
- Open Dynamic Architecture
  - heterogeneous networking
  - heterogeneous components
- Very high level concurrency with complex interactions

**Challenges:**
- understanding and
- predicting behavior
Real-Life CPS Development

- All integration dimensions are present
- Systems are evolving along “spiral-outs”
- New technical challenges are emerging and potential solutions need to be rapidly explored
- All layers of the system are subject to modifications, there are no well defined synchronization points in the development process
- Integration is inherently incremental; deployed systems need to be integrated with components on different level of maturity: prototypical and with simulated systems/components.
How Is It Solved Today?

- Systems are integrated when all components are delivered
  - Acquisition pushes in this direction
- Integration means: “Make it working somehow”
- System Integration Labs do not offer support for spiral development
  - Neither acquisition practices
- There is no approach to deal with incomplete specifications and components

System Integration is the highest risk, most expensive, least predictable step in CPS design
Outline

- A crosscutting CPS need in industry: System Integration
- Nature and scope of challenges
- Elements of a Science of System Integration
Components

- Model-based design
  - Foundation for convergence across disciplines
- Composition theories for heterogeneous systems
  - Decoupling
  - Orthogonalization
- Agile System Integration
  - Extensive use of modeling and model evolution
  - Multi-model simulation
Convergence: Model-Based Design

- **Systems Engineering**: Model-based design has been the state of practice
- **Control Engineering**: Wide acceptance due to popular tools like MathWorks Simulink/StateFlow
- **Software Engineering**: Increasing acceptance due to OMG’s MDA push and wider availability of tool suites
Key Idea: Manage design complexity by creating abstraction layers in the design flow.
(Platform-based design: Alberto Sangiovanni-Vincentelli)

Abstraction layers define platforms.

Abstractions are linked through mapping.

Abstraction layers allow the verification of different properties.

Dynamics models define the composition of functions that describe system behavior.

Software architecture model defines the software components and their interaction.

System-level architecture defines a set of concurrent functional units, where the software architecture can be deployed.
Composition in Heterogeneous Systems

How to achieve composability and compositionality?

- **Orthogonality among the design layers**
  - Controller design depends on assumptions about implementation
  - Orthogonalization removes assumptions
    - E.g. Passivity

- **Decoupling across design layers**
  - E.g., Time Triggered Architecture
  - Timing specification drives execution
  - Static structure

- **Fundamental change in design flows**
  - Cross-layer abstractions
  - New specification/implementation interface
  - Redefining testing and verification

---

<table>
<thead>
<tr>
<th>Plant Dynamics &amp; Requirement Models</th>
<th>Controller Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller design</td>
<td></td>
</tr>
<tr>
<td>Specification/implementation interface</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software Architecture Models</th>
<th>System-Level Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>System-level design</td>
<td></td>
</tr>
<tr>
<td>Specification/implementation interface</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Deployment Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation Platform Design</td>
<td></td>
</tr>
</tbody>
</table>
Agile tools and processes for:

- Rapid modeling of Components and Systems
- Rapid synthesis of adapters and wrappers to integrate systems and components at three levels of fidelity
  - deployed, fully qualified
  - experimental prototype
  - simulated
- Rapid configuration of experiments
  - deployment of services,
  - configuration of environment simulators,
  - config of instrumentation
- Rapid experimental runs
  - control launch, run, post run data collection
  - virtualization of experimental platforms to enable large-scale experiments
- Analysis
  - mapping results back to requirements
- Iterate
How can we integrate the models?
How can we integrate the simulated heterogeneous system components?
How can we integrate the simulation engines?
## Benefits

### Challenges

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Model-Based Design</th>
<th>Composition Theories</th>
<th>Agile System Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>All integration dimensions are present</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>The system is evolving along “spiral-outs”</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>New technical challenges are emerging and potential solutions need to be rapidly explored</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>All layers of the system are subject to modifications, there are no well defined synchronization points in the development process</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Integration is inherently incremental; deployed systems need to be integrated with components on different level of maturity: prototypical and with simulated systems/components</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
CPS-s represent the “center of gravity” in NIT applications in the future

System Integration for CPS is a crosscutting CPS need

An important challenge: Science of System Integration