Prof. Delsing and the EISLAB group http://www.ltu.se/eislab has been a partner of major EU projects in the field, e.g.
- Socrades,
- IMC-AESOP,
- Arrowhead (coordinator),
- FAR-EDGE (WP lead),
- Productive4.0 (WP lead) and
- Arrowhead Tools (coordinator).

Delsing holds positions as vice president and board member of INSIDE (formerly ARTEMIS-IA) and board member of ProcessIT.EU and ProcessIT Innovations.
Engineering of complex System of Systems

Prof. Jerker Delsing
Luleå University of Technology
Sweden
Complex System of Systems - SoS

Complex Cyber Physical System of Systems

Automation

Digitalisation
Plethora of standards to support engineering of automation solutions

IEC 62264, based on ANSI/ISA-95.

Competing standards in similar areas e.g.
IEC 61850, IEC 61970 and IEC 61968, primarily associated with power systems management.
ISO TC 184, collaborating with “Machinery Information Management Open Systems Alliance” (MIMOSA)
ISO 15926 - Industrial automation systems and integration, and
ISO 18435 - Industrial automation systems and integration
RAMI4.0

For life-cycle and hierarchical structure:
IEC 62890 “Life-cycle management for systems and products used in industrial- process measurement, control and automation”
IEC 62264 (ISA-95) / IEC 61512 (ISA-88)

For end-to-end engineering:
AutomationML
ProSTEP iViP
eCl@ss

IEC 81346 “Industrial systems structuring principles”
IEC 62714, AutomationML
IEC 62541 OPC-UA
IEC 61131, IEC 61499, PLC coding
Basic engineering state of the art

IEC 81346
OT meets IT

Production SmartCity
Energy
Building
Life Science
Bank & Finance

2015+
The 4th industrial platform
Cloud IoT SoS Big Data Analytics Block Chain
AI UX

Technology Engineering Tools
IndTech
Market

1980+ Proprietor Automation & Industrial OT

OT meets IT
From enterprise to multi stakeholder operation
Current production automation

- Ridged pyramid
  - Inflexible automation
  - Cross layer dependencies
  - Low/No security

- Heterogeneous and incompatible networks
  - Industrial Ethernet
  - Fieldbus
  - Modbus
  - ASI bus
  - Hart/WirelessHart
  - 4-20 mA
  - ……
The automation technology transition

Hierarchical system implementation

Local automation cloud implementation
Digitised industry

- Dynamic digital industry
  - Changes in run-time
  - High security

- System of Systems - IoT based
  - Interoperable IoT’s
  - Functionality management
  - Security management
Scalability

• Digitalisation is pushing for integration of more systems than today
  • Moving beyond $10^5$ connected IoT’s

• Integration of today isolated systems
  • Preserving
    • Functionality
    • Real time
    • Security
    • Interoperability
    • …..
System of Systems integration to Cyber Physical System of Systems

• Service level integration
  • Descriptions of a plant
    • Physical functions
      • PI&D, ....
      • Control, ....
  • Electrical
    • Topology, logical
  • Communication, computation
    • Topology, Logical
• Wiring
• Layout
Digitalisation and Automation requirements

- Real time performance
- Engineering simplicity
- Interoperability
- Security and trust
- Safety
- Scalability
- System of Systems integration
- Flexibility
Scalability

• Digitalisation is pushing for integration of more systems than today
  • Moving beyond $10^5$ connected IoT’s
  • Integration of today isolated systems
    • Preserving
      • Functionality
      • Real time
      • Security
      • Interoperability
    • Enabling
      • Maintenance
      • Evolution
      • Lifecycle management
Model based engineering - MBE

Modelling complex Cyber Physical System of Systems

Languages

UML - Cyber space
SysML - Integrating cyber space and physics - CPS
AutomationML - Control
...

www.arrowhead.eu
What is SysML

The **Systems Modeling Language** (SysML) is a general-purpose modeling language for systems engineering applications. It supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems.

SysML was originally developed by an open source specification project, and includes an open source license for distribution and use.

SysML is defined as an extension of a subset of the **Unified Modeling Language** (UML) using UML's profile mechanism. The language's extensions were designed to support systems engineering activities.
SysML Tools

MagicDraw - Cameo, commercial
  SysML v1.6
  Extensive graphical system modelling tool based on SysML

Papyrus - Eclipse Open source
  UML + SysML 1.6
System of Systems modelling

- Based on
  - Service Oriented Architecture
  - Micro-system producing and consuming micro-services
SOA/microsystem characteristics

- Look up
  - Requires a service registry
- Late binding
  - Requires orchestration capability
- Loose coupling
  - Autonomous exchange of services, push or pull based
- A micro system performs its function independently
- A micro system can
  - be stateful and is then responsible for and stores its own state
  - be stateless
- A micro system produces and/or consumes one or several services
SoS characteristics

**Operational independence/autonomy of the elements.** The constituent systems can operate independently in a meaningful way, and are useful in their own right.

**Belonging.** The autonomous constituent systems choose to belong to the SoS, and they do that because they see a value for themselves to give up some of the autonomy in order to get benefits from doing so.

**Connectivity.** To let the constituent systems interact, they must be connected, and unless they provide sufficiently generic interfaces, they need to be modified to provide such interoperability. Connectivity in an SoS is thus dynamic, with interfaces and links forming and vanishing as the need arises.

**Diversity - heterogeneity.** Whereas many other systems strive to minimize diversity to simplify the system, an increased diversity in an SoS gives it the ability to better deal with unforeseen situations during its life cycle.

**Managerial independence of the elements.** The constituent systems not only can operate independently, but they do operate independently even while being part of the SoS. They are acquired separately.

**Evolutionary development.** The SoS does not appear fully formed, and functions and purposes are added based on experience.

**Emergent behavior.** The principle purposes of the SoS are fulfilled by behaviors that cannot be localized to any individual constituent system. In an SoS, the emergent behavior is not restricted to what can be foreseen. Instead, it should have the capability to early detect and eliminate bad behavior that emerges.

**Geographical distribution.** The constituent systems only exchange information and not substantial quantities of mass or energy.

**Secure and safe.** Malicious behaviors in a SoS and its constituent systems need to be detected and mitigated to ensure information, system and SoS integrity.
Modelling of System of Systems, SoS

Based on Eclipse Arrowhead

A SOA/microsystem framework for creating automation and digitalisation solutions based on SoS

Key Arrowhead concepts to be modelled

- Network connecting
- Devices hosting
- SW-Systems constituting self contained
- Local clouds integrated to
- System of local clouds
SysML modelling basics

Requirement diagram/table
Use case diagrams
Activity diagram
Block definition diagrams
Internal block diagrams
Parametric diagram
State machine diagram
Sequence diagrams

...
SOA SysML support

Library

Eclipse Arrowhead core systems

Templates for

- Local clouds
- System of Local clouds
- Generic application systems
- Devices
- Network

www.github.com/eclipse-arrowhead
SOA support

SysML Profile
Based on Eclipse Arrowhead
Intend to support several engineering phases for a solution

• Requirement
• Design conceptual, black box,
• Design of implementation, white box,
• Procurement & Engineering
• Deployment
• Maintenance
• Evolution
Integration with the engineering process

Modelling the engineering process
Eclipse Arrowhead engineering

Engineering process
IEC 81346

SoS Requirements
• Functional,
• non-functional,
• security,
• commissioning,
• operations,
• management,
• maintenance,
• evolution
SoLCD

Functional design - black box
• Plant architecture and design,
• Functionality design
• Security design
• Local cloud sectioning
• Core system usage,
• Application system design,
  SoLCDD, LCD, SysD, SD

Procurement of:
• Application hardware, OS, Router,
• Installing OS
White box engineering of:
• Application systems and services code
• Orchestration and security policies
• Installing core and application systems to procured HW
• Configuration of network,
  LCDD, SysDD, IDD

Deployment of:
• HW with core and application systems in plant,
• Orchestration policies
• Security policies
Commissioning of:
• Local cloud functionality
• System of local cloud functionality

Devices and network

Physical deployment at site
• Devices
• Routers
• Power supply
• Network connection

www.arrowhead.eu
Engineering tools for cloud automation systems
Development support, documentation.

SoSD: System-of-Systems Description
SoSDD: System of Systems Design Description
SysD: System Description
SysDD: System Design Description
SD: Service Description
IDD: Interface Design Description
CP: Communication Profile
SP: Semantic Profile
Architecture modelling

- Large scale SoS
  System of Local clouds

- Local scale SoS
  Local Cloud

- System producing and/or consuming services

- Device hosting one or several Systems

- Network integrating hardware

- System-of-LocalClouds-Requirements
  [Class]

- LocalCloud-requirements
  [Class]

- System-Requirements
  [Class]

- Device-Requirements
  [Class]

- Network-Requirements
  [Class]
Eclipse Arrowhead documentation structure

SoSD: System-of-Systems Description
SoSDD: System of Systems Design Description
SysD: System Description
SysDD: System Design Description
SD: Service Description
IDD: Interface Design Description
CP: Communication Profile
SP: Semantic Profile
SoS architecture and engineering in SysML
Let's start with a use case
Use case diagram
Requirements

1. **Id**: "17"
   **Text**: "Tank level control function. Based on level sensor, flow measurement and"

2. **Id**: "18"
   **Text**: "Level measurement accuracy: +/- 1cm
Flow measurement accuracy of actual flow: 1%
Valve flow control: linear
Tank level max: 90%
Tank level min: 10%"

3. **Id**: "19"
   **Text**: "Controller cycle time: 1s"
SoS architecture and engineering in SysML
Functional system and service design

Micro-systems
Micro-services
Functional SoS design - black box

Local cloud functional orchestration
Service exchange functionality
SoS architecture and engineering in SysML
White box engineering

SysDD and IDD
Functional system and service design & design description/implementation 
black box & white box + code

Micro-systems

Micro-services
Functional LC design description - white box

Local cloud functional design description model
Orchestration policies - rules and conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Role (Connector End A)</th>
<th>Role (Connector End B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Orchestration</td>
<td>inout p4 : ServiceDiscovery_HTTP</td>
<td>inout p1 : ServiceDiscovery_HTTP</td>
</tr>
<tr>
<td>2</td>
<td>Orchestration</td>
<td>inout p2 : Orchestration_HTTP</td>
<td>inout p1 : Orchestration_HTTP</td>
</tr>
<tr>
<td>3</td>
<td>Orchestration</td>
<td>inout p2 : GetPublicKey_HTTP</td>
<td>inout p1 : GetPublicKey_HTTP</td>
</tr>
<tr>
<td>4</td>
<td>Orchestration</td>
<td>inout p1 : Flow-CoAP</td>
<td>inout p2 : Flow-CoAP</td>
</tr>
<tr>
<td>6</td>
<td>Orchestration</td>
<td>inout p2 : Orchestration_HTTP</td>
<td>inout p1 : Orchestration_HTTP</td>
</tr>
<tr>
<td>7</td>
<td>Orchestration</td>
<td>inout p2 : GetPublicKey_HTTP</td>
<td>inout p1 : GetPublicKey_HTTP</td>
</tr>
<tr>
<td>8</td>
<td>Orchestration</td>
<td>inout p2 : Configuration-CoAP</td>
<td>inout p1 : Configuration-CoAP</td>
</tr>
<tr>
<td>10</td>
<td>Orchestration</td>
<td>inout p2 : GetPublicKey_HTTP</td>
<td>inout p1 : GetPublicKey_HTTP</td>
</tr>
<tr>
<td>12</td>
<td>Orchestration</td>
<td>inout p2 : Orchestration_HTTP</td>
<td>inout p1 : Orchestration_HTTP</td>
</tr>
<tr>
<td>13</td>
<td>Orchestration</td>
<td>inout p2 : GetPublicKey_HTTP</td>
<td>inout p1 : GetPublicKey_HTTP</td>
</tr>
<tr>
<td>14</td>
<td>Orchestration</td>
<td>inout p2 : SetPoint-CoAP</td>
<td>inout p1 : SetPoint-CoAP</td>
</tr>
</tbody>
</table>
Security policies - rules and conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Role</th>
<th>Role</th>
<th>Security constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Orchestration</td>
<td>inout p4 : ~ServiceDiscovery_HTTP</td>
<td>inout p1 : ServiceDiscovery_HTTP</td>
<td>Security p=authorisation == system certificate: certificate-1...</td>
</tr>
<tr>
<td>2</td>
<td>Orchestration</td>
<td>inout p5 : ~ServiceDiscovery_CoAP</td>
<td>inout p2 : ServiceDiscovery_CoAP</td>
<td>Security-3=authorisation == system certificate</td>
</tr>
<tr>
<td>3</td>
<td>Orchestration</td>
<td>inout p3 : ~Orchestration_HTTP</td>
<td>inout p1 : Orchestration_HTTP</td>
<td>Security 3=authorisation == system certificate</td>
</tr>
<tr>
<td>4</td>
<td>Orchestration</td>
<td>inout p1 : How-CoAP</td>
<td>inout p2 : ~How-CoAP</td>
<td>Security policy=authentication == system certificate...</td>
</tr>
</tbody>
</table>
Functional SoLC design
System of local clouds functional design model
SoS architecture and engineering in SysML
Implementation

We also need

- Devices
- Network
Device implementation

Devices with
- Mandatory core systems
- Support core systems
- Application systems

Router
SoS architecture and engineering in SysML
Functional SoS/
Local cloud
implementation engineering
Functional SoLC implementation engineering

System of local clouds functional implementation model
### Extraction of code

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Specification</th>
<th>Constrained Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Implementation</td>
<td>Authorization v4.3.0 == <a href="https://github.com/eclipse-arrowhead/core-java-spring/tree/master/authorization">https://github.com/eclipse-arrowhead/core-java-spring/tree/master/authorization</a></td>
<td>Authorization-implementation</td>
</tr>
<tr>
<td>2</td>
<td>Implementation</td>
<td>CertificateAuthority v4.3.0 == <a href="https://github.com/eclipse-arrowhead/core-java-spring/tree/master/certificate-authority">https://github.com/eclipse-arrowhead/core-java-spring/tree/master/certificate-authority</a></td>
<td>CertificateAuthority-implementation</td>
</tr>
<tr>
<td>3</td>
<td>Implementation</td>
<td>v4.3.0 == <a href="http://github.com/eclipse-arrowhead/core-java-spring/datamanager">http://github.com/eclipse-arrowhead/core-java-spring/datamanager</a></td>
<td>DataManager-implementation</td>
</tr>
<tr>
<td>4</td>
<td>Implementation</td>
<td>DeviceRegistry v4.3.0 == <a href="https://github.com/eclipse-arrowhead/core-java-spring/tree/master/deviceregistry">https://github.com/eclipse-arrowhead/core-java-spring/tree/master/deviceregistry</a></td>
<td>DeviceRegistry-implementation</td>
</tr>
<tr>
<td>5</td>
<td>Implementation</td>
<td>v4.3.0 == <a href="https://github.com/eclipse-arrowhead/core-java-spring/tree/master/eventhandler">https://github.com/eclipse-arrowhead/core-java-spring/tree/master/eventhandler</a></td>
<td>Eventhandler-implementation</td>
</tr>
<tr>
<td>6</td>
<td>Implementation</td>
<td>Gateway v4.3.0 == <a href="https://github.com/eclipse-arrowhead/core-java-spring/tree/master/gateway">https://github.com/eclipse-arrowhead/core-java-spring/tree/master/gateway</a></td>
<td>Gateway-implementation</td>
</tr>
<tr>
<td>7</td>
<td>Implementation</td>
<td>v4.3.0 == <a href="http://www.arrowhead.eu/api/arrowhead/mqtt-broker">http://www.arrowhead.eu/api/arrowhead/mqtt-broker</a></td>
<td>MQTT-broker-implementation</td>
</tr>
<tr>
<td>8</td>
<td>Implementation</td>
<td>Onboarding v4.3.0 == <a href="https://github.com/eclipse-arrowhead/core-java-spring/tree/master/onboarding">https://github.com/eclipse-arrowhead/core-java-spring/tree/master/onboarding</a></td>
<td>Onboarding-implementation</td>
</tr>
<tr>
<td>9</td>
<td>Implementation</td>
<td>Orchestrator v4.3.0 == <a href="https://github.com/eclipse-arrowhead/core-java-spring/tree/master/orchestrator">https://github.com/eclipse-arrowhead/core-java-spring/tree/master/orchestrator</a></td>
<td>Orchestrator-implementation</td>
</tr>
<tr>
<td>10</td>
<td>Implementation</td>
<td>ServiceRegistry v4.3.0 == <a href="https://github.com/eclipse-arrowhead/core-java-spring/tree/master/service-registry">https://github.com/eclipse-arrowhead/core-java-spring/tree/master/service-registry</a></td>
<td>ServiceRegistry-implementation</td>
</tr>
<tr>
<td>11</td>
<td>Implementation</td>
<td>SystemRegistry v4.3.0 == <a href="https://github.com/eclipse-arrowhead/core-java-spring/tree/master/system-registry">https://github.com/eclipse-arrowhead/core-java-spring/tree/master/system-registry</a></td>
<td>SystemRegistry-implementation</td>
</tr>
<tr>
<td>12</td>
<td>Implementing code pak</td>
<td>v4.3.0 == <a href="https://www.github.com/eclipse-arrowhead/core-java-spring/system-registry">https://www.github.com/eclipse-arrowhead/core-java-spring/system-registry</a></td>
<td>Translation-implementation</td>
</tr>
</tbody>
</table>

Move from here to Docker containers for deployment to
Selected HW and OS
- Server - Linux - Ubuntu 20.10
- Desktop computer, Windows 10.xx, OSX 11.2.1
- Embedded system
  - Raspberry PI
SoS architecture and engineering in SysML
Deployment engineering
SoS/Local cloud implementation engineering
Network deployment

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Role</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ethernet connections Dev 1</td>
<td>inout p2 : Ethernet</td>
<td>inout p2 : Ethernet</td>
</tr>
<tr>
<td>2</td>
<td>Ethernet connections Dev 2</td>
<td>inout p2 : Ethernet</td>
<td>inout p2 : Ethernet</td>
</tr>
<tr>
<td>3</td>
<td>Ethernet connections Dev 3</td>
<td>inout p2 : Ethernet</td>
<td>inout p2 : Ethernet</td>
</tr>
<tr>
<td>4</td>
<td>Ethernet connections Dev 4</td>
<td>inout p2 : Ethernet</td>
<td>inout p2 : Ethernet</td>
</tr>
<tr>
<td>5</td>
<td>Ethernet connections Router</td>
<td>inout p2 : Ethernet</td>
<td>inout p2 : Ethernet</td>
</tr>
</tbody>
</table>

ToDos

- ID of instances to be made according to standards
- Automatic naming of instances based of standards

Applicable standards

- ISO 15926
- ISO 10303
- ISO 19650 - BIM v5
Deployment of policies

Orchestration policies

PlantDescription system
Management system

{ Orchestration system

Authorisation rules

PlantDescription system
Management system

{ Authorisation system
SoS architecture and engineering in SysML
Adding functionality

Making use of

Support core systems models
  Translator
  DataManager
  TimeManager
  ...

Adaptor systems models
  OPC-UA -> Arrowhead
  Modbus TCP -> Arrowhead
  Z-wave -> Arrowhead

Application function systems models
  Code generation from models to executable code
Engineering automation

Move from SysML models of complex SoS to Docker containers for deployment to Selected HW and OS
Desktop computer,
Embedded system e.g.
SoS solution generation

From SysML model of complex SoS
Integration with Eclipse IDE
Plug-ins
Code generation
Output
Containers of working code
Deployable code to selected hardware devices and physical network
Conclusions

• SoS solution will rapidly become very complex
• MBE is a time and cost effective approach
  - Automating SoS solution code creation and code reuse
  - Automated extraction of orchestration and security management policies
• Based on open source Eclipse
  - architecture
  - integration framework
  - code and
  - tools
• Github
  www.github.com/eclipse-arrowhead
Availability

Github

www.github.com/eclipse-arrowhead
Comments! Questions?

jerker.delsing@ltu.se