A multi-scale modelling perspective for SoS architectures

Ilhem KHLIF$^1$, Mohamed HADJ KACEM$^1$, Ahmed HADJ KACEM$^1$ and Khalil DRIRA$^2$

$^1$ Research Laboratory ReDCAD, University of Sfax, Tunisia
$^2$ LAAS-CNRS and Univ. Of Toulouse, France
The general Context of our study

Software architecture

Design and description Of software architecture

Development of large and complex Systems of Systems

Modeling software architectures for SoSs
The Problem

- How to handle software Systems of Systems complexity?
- How to describe software architectures to facilitate their validation at different description levels?

Our proposal:
Consider different architecture descriptions with different levels of modelling details: “the scales”
Our Work Objectives

- Provide solutions for modeling software architectures
  - To facilitate their validation at **different description levels**
  - To Validate through case studies for Systems of Systems (ERCMS, PMS, IAD)

- Propose a design approach
  - Organized around a set of **architectural transformation rules**
  - Based on **SysML** visual notations

- Present a multi-scale modeling approach for SoS architecture description for
  - Mastering the description details complexity
  - Validating through involving **intrinsic or mission-specific properties**
    - Allow the **validation while remaining tractable w.r.t. complexity**
    - Ensure the model **correctness w.r.t. SysML description**
The implemented Multiscale description approach (Eclipse Plugin)

Model transformation operations (Graph Matching & Transformation: GG semantics)

- «Require & Delete»
- «Require & Preserve»
- «Insert»

Validation (correctness of description, correctness of the system)

- Rule-oriented description technique
- Model transformation rules
- Vertical and horizontal refinement process
The Refinement Process

Start by modelling the first scale by a given coarse grain description using a SysML block diagram

This diagram is refined through model transformation operations (Horizontal and vertical refinements)
A model refinement executes the “Insert” transformation operation to add new blocks and connections

Transit from one scale to another by a refinement of composition enriching its architecture

Reach a fine-grain description representing the necessary details
What are Scales

- A « generic » description scale
  “GS\textsubscript{n}” (enriched with vertical refinements)
  - Is a model that provides additional details of design that pertain to “GS\textsubscript{n+1}”
  - Is a description level that allows the architect to:
    - Describe the necessary details to understand the SoS architecture and to validate the associated properties

- Under each generic description scale there are several specific description scales (Horizontal refinement)
  - Providing more details of a given SysML current description
What is an SoS?

- A set of collaboratively integrated systems that possess two additional properties:
  - Operational independence of the constituents
  - Managerial independence of the constituents

- A composition of systems in which its constituents, themselves systems, are separately discovered, selected and composed
  - To form a more complex system that performs a mission not possible by one of the constituent systems alone, i.e., it creates an emergent behavior
SoS intrinsic characteristics

- **Operational Independence**
  - The constituents of an SoS can execute independently

- **Managerial Independence**
  - The constituents of an SoS are separately integrated but manage their own resources independently

- **Evolutionary Development**
  - The SoS can evolve over time to respond to changing characteristics.

- **Emergent Behavior**
  - The SoS is capable to deliver new functions that are obtained from the composition of its constituents rather than from a singular constituent

- **Geographic Distribution**
  - The constituent systems are geographically extended in such a way interaction between them is limited to information exchange
Categories of SoSs

- **Directed SoS** *(e.g. The Integrated Air Defense System)*
  - A set of systems that operate subordinated to the central purpose
  - The constituent systems maintain an ability to operate independently
  - The operational mode is subordinated to the central managed purpose

- **Collaborative SoS**
  - A set of systems that collaborate to fulfill the agreed central purposes

- **Virtual SoS**
  - Lack a central management authority and a centrally agreed purpose for the SoS

- **Acknowledged SoS**
  - Recognized objectives,
  - A designated manager
A graphical modelling language
- Is based on UML
- Involves modelling blocks instead of modelling classes

The block definition diagram (bdd)
- Allows us to give a structural description of the system
- Describes the relationship among blocks (e.g., composition, association, specialization)

The internal block diagram (ibd)
- Is a white box view of a block
- Describes the internal structure of system in terms of parts, ports and connectors.
Overview

- Context
- Problematic
- Objectives
- Approach
- State of the art

Contribution
- SysML Metamodel
- Model transformation operations
- Verification rules for model traceability
- Use case: Application to the Integrated Air Defense SoS

Conclusion and perspectives
SysML Metamodel

- \( GS_0 \) defines the **whole system** by its name
- The beginning of traceability (System requirements are specified)

- Two horizontal refinements are associated with \( GS_1 \)
  - \( \{ GS_1, SS_0 \} \) shows **all blocks** that **compose** the system
  - \( \{ GS_1, SS_1 \} \) describes the **links** between those blocks

- Two specific scales are associated with \( GS_2 \)
  - \( \{ GS_2, SS_0 \} \) presents **composite** for blocks
  - \( \{ GS_2, SS_1 \} \) identifies the list of:
    - **Communication ports** for each composite block
    - **Interfaces** for communication ports, and
    - **Connections** between the system constituents
**Verification rules for model traceability (1/2)**

<table>
<thead>
<tr>
<th>GS₁</th>
<th>SS₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_{n}^{0}$</td>
<td>$B_{n}^{1}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GS₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_{n+1}^{0}$</td>
</tr>
<tr>
<td>$B_{n+1}^{1}$</td>
</tr>
<tr>
<td>$B_{n+1}^{2}$</td>
</tr>
<tr>
<td>$B_{n+1}^{2.1}$</td>
</tr>
<tr>
<td>$B_{n+1}^{2.2}$</td>
</tr>
</tbody>
</table>

**Approach with overlap between scales**

- **Traceability rule for block identification**
  - If we keep track of a block, the traceability is trivial, and the identification of the block is preserved.

- We note a block by $B_{n}^{m}$
  - $n$: the scale number ($n \geq 0$)
  - $m$: a cursor on the current block ($m \geq 0$)
  - $n, m$ are decomposed in the next scale

- If we have a block $B_{n}^{1}$, then its composing blocks in the next scale are named $B_{n+1}^{1.1}$ etc.
Verification rules for model traceability (2/2)

Approach with scale separation

- Rules for decomposing links between blocks
  - If a link is divided according to its identifiers (Producer Consumer), then a trace of the link decomposition is added

- Rule1: If $B_n^1$ is a Consumer, then the link between $B_n^1$ and $B_n^2$ in $GS_n$ will be transformed into an assembly connection in $GS_{n+1}$ extending from the source $B_{n+1}^{2.2}$ to the target $B_{n+1}^{1.1}$

- Property: $\forall B \in \{\text{Producers}\}, \exists C \in \{\text{Consumers}\}$ such that $\forall m:T$ (a message with a type $T$) sent by $B$ $m:T$ is received by $C$
Application to Integrated Air Defense (IAD)

- Application of the model transformation rules to the IAD SoS
- Verification of the model traceability properties during the refinement process
- Illustration of the block diagrams for generic and specific description scales
  - $G_S^0$ illustrates The Integrated Air Defense system as a directed SoS
  - $\{G_S^1, S_S^0\}$ presents the constituent systems as blocks (Ground Force, Air Force, and Maritime Force)
  - $\{G_S^1, S_S^1\}$ shows the internal communications between these systems as associations between blocks
  - $\{G_S^2, S_S^0\}$ represents composites of each constituent system.
    - Eg. Ground force is composed of Surveillance radars, command and control site, anti-aircraft artillery, and anti-aircraft weapons (Short Range Air Defense (SHORAD), and High to Medium Air Defense (HIMAD))
  - $\{G_S^2, S_S^1\}$ represents the internal connections to express the communication between all blocks that received orders from the Command and Control block.
Conclusion

- a multi-scale approach for software architectures at the **conceptual level**
- SysML notations at the **architectural style level**
- the state of the art on how SoS architecture modelling have been addressed.
- refinement rules through model transformation techniques
- verification rules for model traceability
- Implementation of an eclipse plugin
Ongoing/Future work

- Updating the state of the art in order to contribute to the area of SoS
- Considering the need and relevance of these large and complex systems
- Applying the multi-scale approach to other use cases for modeling Systems of Systems architectures
- Automated generation of Event-B specification for properties verification using theorem prover techniques
A multi-scale modelling perspective for SoS architectures

khalil.drira@laas.fr