# An Incremental Model-Based Design Methodology to Develop CPS with SysML/OCL/Reo

PhD student: Perla Tannoury
Supervisors: Dr. Samir Chouali & Dr. Ahmed Hammad
perla.tannoury@femto-st.fr, schouali@femto-st.fr, ahammad@femto-st.fr
Univ. Bourgogne Franche-Comté, FEMTO-ST Institute/CNRS, Besançon

April 2022

**Abstract**

Modeling Cyber-Physical Systems (CPS) remains a challenge due to their interconnected networks of heterogeneous embedded systems that operate in a physical environment. In this paper, we introduce a new modeling approach that relies on SysML, OCL, and Reo to capture the different aspects of CPS, including requirements, architecture, and interaction protocols. The novelty of our approach relies in the combination of SysML and Reo to handle the complexity of CPS architecture and protocols, in the design step by proceeding incrementally. Furthermore, we define OCL constraints to specify rules to be respected to model consistently CPS.

**Keywords:** CPS. SysML. Reo. OCL. Meta-Model. Incremental Design.

## 1 Introduction

Cyber-physical systems (CPSs) are a couple of computational systems (Cyber) with sensors and actuators (Physical), executing in varying spatial and temporal contexts while exhibiting diverse behaviors across runs. The relationship between the logical components and the physical ones is very complex. Therefore, many obstacles arise through their combination. To address these problems, a design methodology is needed on the one hand in a modeling language for modeling and analyzing all CPS facets and on the other hand in an appropriate tool to verify and simulate the modeled system.

The System Modeling Language (SysML), aims to support systems that exhibit hybrid phenomena, such as CPSs, by combining the continuous phenomena of physical systems with the discrete phenomena of software systems. SysML has many advantages in modeling a system's architecture, behavior, and requirements. However, CPS are usually developed by assembling components incrementally, which leads to complex behavior of the final system. SysML is limited to correctly specifying such behaviors, hence the need to use a powerful coordination language, to specify compositionally complex interactions between CPS components.

Reo [1], is a powerful coordination language that captures the foundations of the system. Reo helps the CPS designer to formally specify how, when, and under what conditions data can flow from a component's input to output ports. It makes it simple to design and analyze the dynamics of scalable systems, as well as to automatically validate conformance and ensure interoperability, which can be used to overcome the limitations of SysML in modeling CPSs. In addition, Reo has a graphical notation, which allows us to use it to enrich SysML models.

To the best of our knowledge, we are not aware of a comprehensive work on CPS modelization that combines SysML and Reo together. Most of previous works in [2, 4, 3] either focus on SysML or on Reo but never on both. SysML alone models the architecture and internal communication flow of components. Reo alone models the interaction protocols of the components. Meanwhile, the combination of SysML and Reo offers detailed and precise internal communication of components by modeling their architectures and protocols.

In this paper, we introduce a new incremental design methodology approach ”SysReo”, that combines SysML, OCL, and Reo for modeling CPS in the design step. We first define SysReo Meta-models allowing, on one hand, to define the extended SysML Block Definition Diagram (ExtBDD) to specify CPS hierarchical structure, on the other hand, to define Reo Internal Block Diagram (Reo IBD) by combining SysML and Reo notations to model interconnection between CPS components and their exogenous protocols. ExtBDD and Reo IBD allow to model CPS architecture and to capture their complex protocols. Then, we define OCL constraints on SysReo models to ensure the consistency of SysReo models between successive levels of modeling, in the context of our incremental modeling approach. The main purpose of our incremental model-based design methodology is to tackle CPSs design challenges and to express their interactions explicitly.

The paper is structured as follows. Section 2 describes the proposed modeling approach of SysReo models. Section 3 concludes the paper and presents the future works.

## 2 Modeling approach based on SysML/OCL/Reo

In this section, we present our model-based design methodology. First, we show the steps to follow to design incrementally CPS with SysML, OCL, and Reo. Then we define the meta-models that were exploited in the first step.

**Approach steps:** We present our modeling approach as shown in Figure 1. First, the modeler starts by specifying a requirement diagram (RD) to analyze and organize CPS requirements. The second phase is dedicated to specify CPS architecture, where the modeler defines the system's components as blocks using ExtBDD (discussed below). Then, in the third phase, the modeler uses an incremental approach starting from an abstract specification of the global internal system, described by Reo IBD. The latter is built gradually by selecting the appropriate components in such a way that their composition respects the constraints defined in the abstract specification. The fourth phase is dedicated to link requirement diagram to ExtBDD and Reo IBD. Finally, the modeler uses our OCL rules on ExtBDD and Reo IBD to precise and to detail the abstract diagrams, and to ensure the consistency of SysReo models.

**SysReo meta-models:** Figure 2 represents the meta-models of SysReo.The latter are described below.

*ExtBDD:* Figure 2A represents the meta-model of ExtBDD that is used to model the system's overall architecture. It is based on the block definition diagram, which provides an abstract representation of components and their connections. A block contains proxy ports and internal operations. Each proxy port is linked to an input/output interface block that contains provided (input) or required (output) services of a component. Operations that are defined in a block can be translated into functions which represent the behavioral aspect of the system.

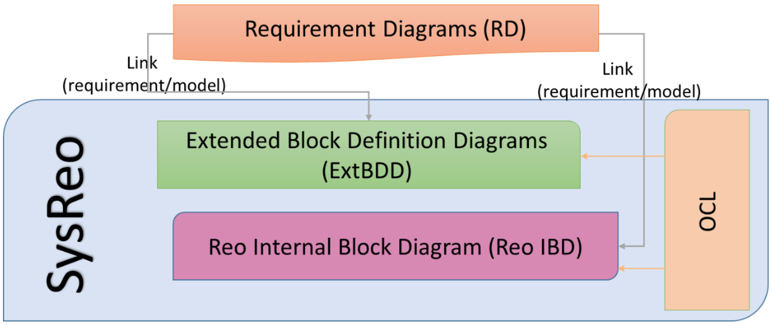
Requirement Diagrams (RD)
Link (requirement/model)
Link (requirement/model)
SysReo
Extended Block Definition Diagrams (ExtBDD)
Reo Internal Block Diagram (Reo IBD)
OCL

Figure 1: Modeling approach overview.

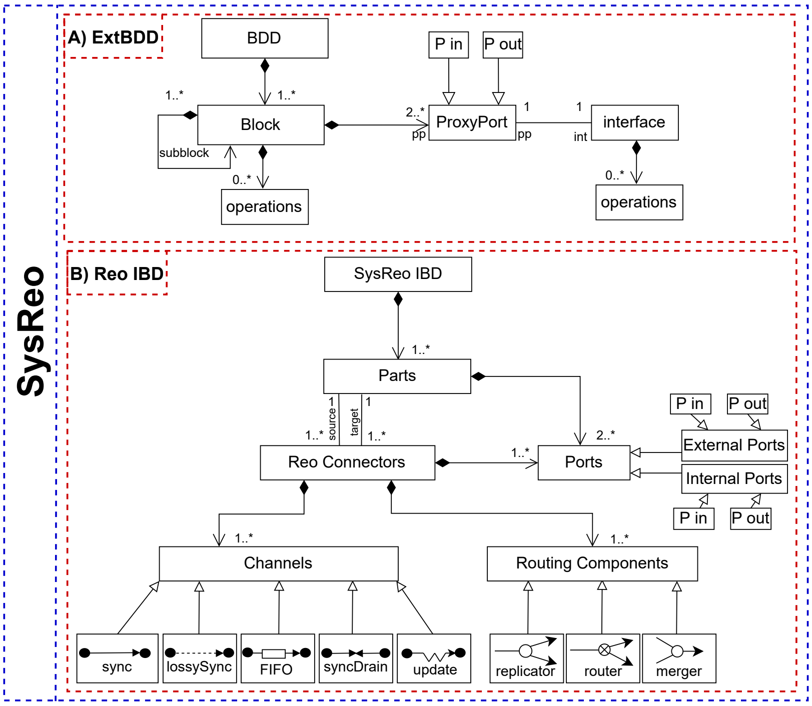

Figure 2: Architecture modeling approach of SysReo.

*Reo IBD:* Figure 2B captures the meta-model of Reo IBD that is based on ExtBDD to assemble the parts that compose the main block. It aims to model the second part of CPS architecture: connection between parts, specified as exogenous Reo protocols. Reo IBD is composed of Parts and Ports, where each part models a CPS component. The parts are "black-box" components composed of proxy ports. Proxy ports are used to exchange data between internal parts. The parts communicate with each other using one or many Reo connectors. These Reo connectors can be basic channels (sync, FIFO, syncDrain...) or routing components (replicator, merger, router...). Table 1 represents some constraints on ExtBDD and Reo IBD. For example, the first OCL in Table 1 is used to show that all the operations in a block must exist in the set of operations of its sub-blocks. The second one is used to precise that a Part (component) has at least one offered and one required services in the system (input/output), therefore the number of its ports should be at least ”2”.

Table 1: OCL Well-formedness constraints of SysReo.

| SysReo | Description | OCL |
|---|---|---|
| ExtBDD | All operations of a Block must exist in the set of operations of its subblocks | `context Block inv: self.pp.int.operations-> forAll (p:operations implies p in self. subblock.pp.int.operations)` |
| Reo IBD | Each Part must have at least two Ports (input , output) | `context Part inv : Part.Ports -> size()≥2` |

# 3 Conclusion and future works

In this paper, we first introduce a novel approach, ”SysReo”, which is a composition of SysML, OCL, and Reo to enable faithful modeling of a CPS. Then, we used OCL to precise constraints on SysReo models. Therefore, from the Reo level, the modeler can easily construct (possible) large scalable CPS. With SysML, the modeler can describe, on a single model, different aspects of a CPS and analyze the behavior, structure, and requirements of the components. As future works, we plan to: (1) Consider more SysML diagrams to enrich our methodology (parametric diagrams, sequence diagrams...). (2) Define formal semantics of SysReo models to verify formally OCL constraints and to verify safety properties on CPS. (3) Illustrate our work on the case study of the Communication-Based Train Control (CBTC) system to prove the feasibility of our approach.

# References

[1] Farhad Arbab. Reo: a channel-based coordination model for component composition. *Mathematical Structures in Computer Science*, 14(3):329–366, 2004.

[2] Ping Huang, Kaiqiang Jiang, Chunlin Guan, and Dehui Du. Towards modeling cyber-physical systems with sysml/marte/pccsl. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 264–269. IEEE, 2018.

[3] Esther Palomar, Xiaohong Chen, Zhiming Liu, Sabita Maharjan, and Jonathan Bowen. Component-based modelling for scalable smart city systems interoperability: A case study on integrating energy demand response systems. *Sensors*, 16(11):1810, 2016.

[4] Jian Xie, Wenan Tan, Zhibin Yang, Shuming Li, Linquan Xing, and Zhiqiu Huang. Sysml-based compositional verification and safety analysis for safety-critical cyber-physical systems. *Connection Science*, pages 1–31, 2021.