Integrating SysML and Timed Reo for Modeling Interactions in Cyber-Physical Systems Components

Perla Tannoury and Ahmed Hammad
FEMTO-ST Institute, CNRS, University of Bourgogne Franche-Comté, Besançon, France

Keywords: CPS, Specifications, Structural Modeling, SysML, Timed Reo, Timed SysReo.

Abstract: Modeling Cyber-Physical Systems (CPS) with timing constraints remains a challenge due to the complex behaviors of their interconnected components that operate in a physical environment. In this paper, we introduce "Timed SysReo", a novel incremental design methodology that integrates SysML and Timed Reo for modeling CPS architectures and timed interaction protocols during the design phase. We first define the meta-models to formalize CPS model architecture and to detail timed connections between its components. Then, we propose to precise the meta-model with Object Constraint Language (OCL), that imposes rules to be respected in order to ensure consistency between timed models in our incremental design approach. Finally, we demonstrate our approach through an example of the Smart Medical Bed (SMB) system.

1 INTRODUCTION

Cyber-Physical Systems (CPSs) consist of both software and physical elements that interact continuously with each other (Kim and Kumar, 2012). They are used in diverse domains such as healthcare, smart cities, and autonomous vehicles (Tartarisco et al., 2024; Barroso et al., 2023).

Efficient CPS modeling is crucial for verifying functionality before implementation, especially in critical domains. However, incorporating timing constraints adds complexity and risks. Errors in these models can be particularly risky in sectors like emergency response and medical applications.

Various languages are used in CPS modeling (Mallet, 2015; Bouskela et al., 2022; Genius and Apvrille, 2023). We chose SysML (Hause et al., 2006) for its ability to model heterogeneous systems, which is widely employed in industrial applications. However, CPS components interact with timing constraints, leading to complex behaviors. While SysML enhances stakeholder understanding through its graphical representation, it may not fully capture and verify these intricate timed interactions.

Timed Reo (Arbab et al., 2007), an extension of Reo (Arbab, 2004), is crucial for managing complex system protocols, especially in CPS where timing is critical. It enables precise determination of data flow timing between component ports, ensuring compliance with essential timing constraints. This simplifies system design, aids validation, and ensures interoperability within specified time frames, addressing SysML's limitations. Timed Reo also offers a formal representation of timed coordination among components, facilitating system property analysis, and its graphical representation enhances usability. However, its complexity and limited adoption compared to SysML may pose challenges for stakeholders in understanding its implementation.

To date, there is no comprehensive study integrating SysML and Timed Reo in CPS modeling. Previous research has mainly focused on SysML (Huang et al., 2018; Xie et al., 2021) or Timed Reo (Arbab et al., 2007; Kokash et al., 2013) separately. Recently in (Tannoury, 2022; Tannoury et al., 2022; Tannoury et al., 2023), "SysReo", a domain-specific language (DSL) combining SysML with Reo, was introduced to enhance CPS validation and verification. While "SysReo" effectively models CPS requirements, structure, and behavior, it struggles with handling timing constraints.

In this paper, we introduce "Timed SysReo", a novel incremental design methodology that integrates SysML and Timed Reo for modeling CPS architectures and timed interaction protocols during the design phase. Our methodology involves decomposing complex CPSs into manageable levels, initially establishing an abstract model that is iteratively refined while ensuring consistency. Initially, an abstract model is defined and progressively refined while en-

Tannoury, P. and Hammad, A.
Integrating SysML and Timed Reo for Modeling Interactions in Cyber-Physical Systems Components.
DOI: 10.5220/0012854100003753

In *Proceedings of the 19th International Conference on Software Technologies (ICSOFT 2024)*, pages 477-484
ISBN: 978-989-758-706-1; ISSN: 2184-2833

suring consistency throughout. The Timed SysReo framework adopts the hierarchical structure diagram from SysReo's Extended SysML Block Definition Diagram (ExtBDD) to represent the CPS structure. Unlike SysReo, Timed SysReo focuses on modeling the internal structure and interaction protocols of CPS components, accommodating timing constraints which SysReo cannot handle. Our contribution introduces the Timed Reo Internal Block Diagram (Timed Reo IBD) to represent the internal structure and timed interaction protocols of CPS components, enhancing clarity, particularly in critical CPS scenarios requiring precise timing. Timed SysReo comprises three primary diagrams for CPS modeling: (1) the requirement diagram for capturing CPS needs, including functional and non-functional requirements, (2) the ExtBDD diagram for representing CPS hierarchical structure, and (3) the Timed Reo IBD diagram for modeling CPS inner architecture and timed interaction protocols. In this work, we define the Timed SysReo language and present its meta-models. Additionally, we establish OCL constraints on the Timed Reo IBD model to enhance precision. Our methodology aims to address challenges in designing medical CPSs by systematically expressing timed interaction protocols at each design level. Finally, we demonstrate our approach through a Smart Medical Bed (SMB) example.

The paper is organized as follows. Section 2 provides a brief overview of SysML, Timed Reo, and Object Constraint Language (OCL). Section 3 describes the proposed modeling approach using Timed SysReo models. Section 4 presents our example of the SMB (Smart Medical Bed) system using Timed SysReo models. Finally, Section 5 concludes the paper while briefly discussing future work.

2 BACKGROUND

This section offers a succinct overview of SysML, Timed Reo, and Object Constraint Language (OCL).

2.1 SysML in a Nutshell

The System Modeling Language (SysML) (Hause et al., 2006) facilitates modeling heterogeneous complex systems across diverse industries. It aligns the input from different stakeholders to maintain consistency and uphold superior design standards. Maintained by the Object Management Group (OMG) (Delligatti, 2013), SysML comprises nine diagram types used to model the requirement, structure, and behavior of CPS. In the context of Smart Medical Bed (SMB) system modeling, our focus lies on requirement, block definition, and internal-block diagrams. Below, we outline the SysML concepts relevant to these diagrams, essential for modeling the SMB system effectively.

The Requirement Diagram (RD): delineates the system requirements anticipated by users. It illustrates the relationship between requirements and other model elements that either "satisfy" or "verify" them. This diagram provides a modeling framework for text-based requirements.

The Block Definition Diagram (BDD): is a structural diagram that visually represents system components using blocks. It depicts relationships between these blocks and their hierarchical structure. Two types of blocks are distinguished: Atomic and composite blocks. Each block comprises a name, values, properties, referenced blocks, components, operations, and constraints. Ports located on the sides facilitate communication with the system. Blocks can represent tangible or conceptual entities like hardware, software, physical objects, and abstract entities.

The Internal Block Diagram (IBD): is a structural diagram that represents the static state of the system. It comprises sub-blocks detailing the internal arrangement of the system. These sub-blocks interact via properties, parts, connectors, ports, and interfaces, facilitating various interactions such as state transitions, software operations, input/output flows, and continuous interactions.

2.2 Timed Reo

Before Timed Reo, Reo (Arbab, 2004) served as a channel-based coordination language for concurrent and distributed systems. Reo constructs complex connectors using basic channels to regulate communication in CPS, with these connectors being exogenous and combinable. Constraint Automata (CA) (Baier et al., 2006) formally represent and analyze Reo, capturing behavior and data flow.

Timed Reo expands upon Reo by introducing channels with timing constraints, which include timed channels for timeouts and delays (Arbab et al., 2004; Arbab et al., 2007). Its primary aim is to specify exogenous protocols governing timed interactions among components in concurrent applications. Additionally, Timed Reo's formal semantics are captured by Timed Constraint Automata (TCA), extending Constraint Automata (CA) to describe behavior incorporating timing constraints. TCA feature two types of transitions: internal changes of locations driven by time constraints and transitions representing synchronized input/output operations at ports.

Definition of TCA: A TCA A = $(L, L_0, N, \rightarrow, C, ic)$ is composed of:

- L: set of locations (or states).
- L_0: initial location where $L_0 \in L$.
- N: set of port names.
- $\rightarrow$: transition relation $\rightarrow \subseteq L \times 2^N \times DC \times CC \times 2^C \times L$, where DC is the set of Data Constraints (DC) over a finite data domain. DC is a condition that must be met for data to be exchanged between two components. For example, DC can be used to implement a filter pattern that only allows certain types of data to be exchanged, or they can be used to implement a synchronization pattern that ensures that two components exchange data in a specific order. CC is a clock constraint.
- C: set of clocks.
- $ic : L \rightarrow CC$ is a function that assigns to any location L an invariance condition ic(L).

2.3 OCL

The Object Constraint Language (OCL) (Cabot and Gogolla, 2012) serves as a general-purpose formal language enhancing UML and SysML models. Its main purpose is to precisely specify constraints, filling a gap where graphical notations may lack clarity and conciseness.

3 MODELING APPROACH USING TIMED SYSREO

In this section, we outline our model-based design methodology. First, we detail the incremental CPS design process using Timed SysReo. Then, we define and refine the meta-models employed in this process using OCL.

3.1 Approach Steps

We outline our modeling approach in Figure 1. Initially, the modeler specifies a Requirement Diagram (RD) to analyze and organize CPS requirements. The subsequent phase focuses on defining CPS architecture, wherein system components are delineated as blocks using ExtBDD. In the third phase, an incremental approach is employed, starting with an abstract specification of the global internal system using Timed Reo IBD. This specification evolves gradually by selecting components that adhere to constraints defined in the abstract specification. Notably, Timed Reo IBD can incorporate timing constraints critical for modeling CPS. Failure to model timing accurately in CPS can result in life-threatening situations such as medication errors, vehicle accidents, aviation incidents, and industrial disasters. The fourth phase involves linking the requirement diagram to ExtBDD and Timed Reo IBD. Finally, our OCL rules are applied to ExtBDD and Timed Reo IBD to refine and specify the abstract diagrams, ensuring consistency in Timed SysReo models.

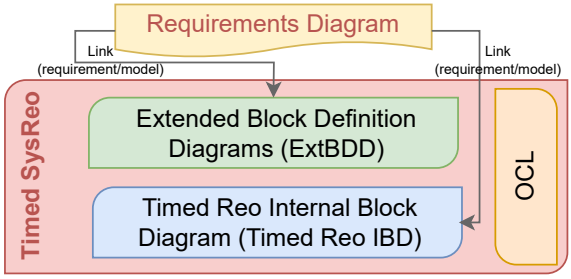

Figure 1: Modeling approach based on Timed SysReo.

3.2 Timed SysReo Meta-Models

In Figure 2, we represent the process of transforming the SysML Block Definition Diagram (BDD) and Internal Block Diagram (IBD) into Timed SysReo Extended Block Definition Diagram (ExtBDD) and Timed Reo Internal Block Diagram (Timed Reo IBD), respectively. The transformation consists of two parts:

1. The transformation of SysML Block Definition Diagram (BDD) into Timed SysReo Extended Block Definition Diagram (ExtBDD) involves using SysML BDD meta-models and splitting the CPS hierarchy into two levels. The first level represents the abstract model of CPS, with primary components as main blocks, and the second level depicts concrete components as sub-blocks. ExtBDD provides a more detailed view of the system structure compared to traditional SysML BDD, aiding in managing system complexity and ensuring design alignment with requirements. Additionally, ExtBDD meta-model comprises blocks containing proxy ports and internal operations, facilitating the translation of operations into functions representing system behavior.
2. The conversion of SysML Internal Block Diagram (IBD) into Timed Reo Internal Block Diagram (Timed Reo IBD) entails employing SysML IBD meta-models and replacing IBD connectors with "Timed connectors". These Timed connectors enable explicit representation of component inter-

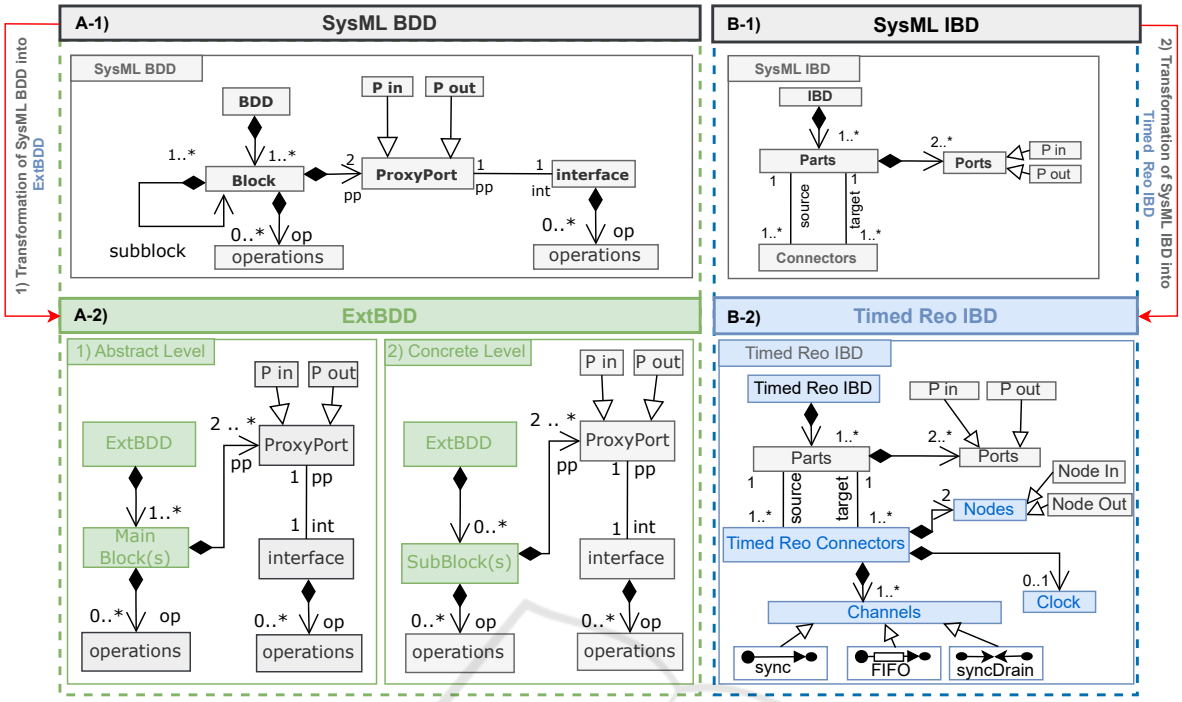

Figure 2: From SysML diagrams to Timed SysReo diagrams.

nal composition and timed interaction protocols, thereby enhancing system reliability, safety, and performance. Moreover, Timed Reo connectors possess formal semantics, ensuring precision and verifiability through formal methods, which ultimately improves reliability and reduces development time and costs. The Timed Reo IBD meta-model, extending from ExtBDD, aims to represent the internal structure of CPS components and their timed connections across various parts of the system. It encompasses components such as Parts, Ports, and Timed Reo connectors. These connectors offer a range of functionalities to control data flow effectively. For instance, the Sync channel promptly transfers data from input to output, while FIFO temporarily stores data before transmitting it. SyncDrain, on the other hand, receives data from two sources simultaneously, discarding it thereafter. The Filter channel selectively forwards data only when specific conditions are met. Additionally, Timed Reo connectors feature nodes facilitating data exchange between system components. They also incorporate clocks to monitor time intervals, enabling the system to manage timeouts or delays efficiently.

In summary, "Timed SysReo" offers a more comprehensive and adaptable approach to system design, which results in more effective and reliable CPS. The use of ExtBDD provides a more precise and detailed hierarchical view of the system, while Timed Reo IBD allows for explicit modeling of the internal composition of the system and the timed interaction protocols among its components.

3.3 OCL on Timed SysReo Meta-Models

To formalize constraints on Timed SysReo meta-models, we used predefined OCL rules to precise the latter. These rules should be respected by CPS designers in order to insure consistency in Timed SysReo models.

The OCL constraints for ExtBDD meta-model (see Figure 2A-2) in Table 1 ensure the integrity of its structure. The first constraint mandates that every operation within a Block must also exist in its subblocks, maintaining consistency across levels of detail. The second constraint requires each Block to have at least two Proxy Ports, ensuring input and output functionality. These constraints guarantee the reliability and completeness of the model.

The OCL constraints for Timed Reo IBD meta-model (see Figure 2B-2) in Table 2 ensure the completeness and consistency of the model. The first constraint is used to differentiate parts by names. The second one mandates that each Part must have at least one input and one output Port, essential for data exchange. Lastly, the third constraint specifies that a

Table 1: OCL Well-formedness constraints of ExtBDD.

Description	OCL Constraints
1) Operation within a Block must be present in the set of operations of its subblocks.	`context Block inv: self.pp.int.op-> forAll (p:op implies p in self.subblock.pp. int.op)`
2) Block should contain a minimum of two Proxy Ports, encompassing both input and output functionalities.	`context Block inv : Block.proxyport.pin-> size()≥ 1 and Block.proxyport.pout-> size()≥ 1`

Timed Reo connector can have either one clock or none, ensuring consistent timing behavior. These constraints collectively maintain the accuracy and reliability of the Timed Reo IBD model.

Table 2: OCL Well-formedness constraints of Timed Reo IBD.

Description	OCL Constraints
1) Part needs to be distinguishable from other blocks by name	`context Parts inv : Parts.AllInstances() ->forAll (p1, p2 | p1 <> p2 implies p1.name <> p2.name)`
2) Parts must have at least two external Ports (input, output)	`context Parts inv: Parts.Ports.pin->size() ≥ 1 and Parts.Ports.pout-> size() ≥ 1`
3) Timed Reo connector can have only one clock or none	`context TimedReoConnector inv: TimedReoConnector .clock->size() = 0 or TimedReoConnector .clock->size() = 1`

4 SMART MEDICAL BED (SMB) EXAMPLE

In this section, we present our example of the Smart Medical Bed (SMB) system. First, we begin by briefly introducing the SMB system. Subsequently, we collect information about the SMB system and analyze it using our Timed SysReo models. This process involves specifying the system's requirements, designing its structure, internal composition and interaction protocol while considering timing constraints.

4.1 SMB Overview

The Smart Medical Bed (SMB) integrates various sensors to monitor patient vital signs, with a focus on temperature sensing. Data from the temperature sensor is transmitted to a Remote Terminal Unit (RTU) within a latency period of 1 Time Unit (TU). The RTU facilitates seamless data communication between the Smart Bed (SB) and the Nursing Station (NS), analyzing data and promptly alerting healthcare staff if abnormalities are detected, all within a latency period not exceeding 3 TUs. The SMB system consists of three main components: SB, RTU, and NS, each playing a vital role in patient monitoring and communication.

Our study focuses on modeling and validating the requirements, structure, and timed interaction protocol of the SB and RTU within the SMB system. Leveraging the Timed SysReo model-driven approach, we analyze system requirements and architect the system while considering timing constraints.

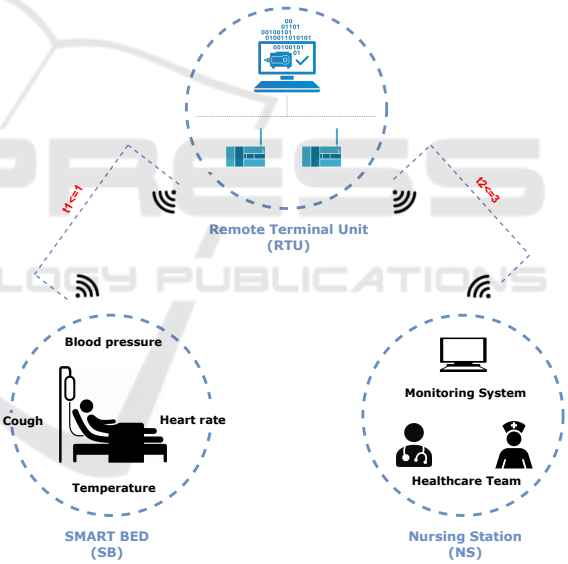

Figure 3: Smart medical bed architecture.

4.2 Modeling SMB with Timed SysReo

4.2.1 Requirements

During the modeling process, ensuring system functionality and usability is paramount. The initial step involves identifying specific system needs outlined in the requirement diagram in Figure 4, which models the SMB system's functional requirements to ensure smooth operation and user satisfaction.

For instance, requirement R1 underscores the need for the SMB system to collect temperature data

from sensors within the Smart Bed and monitor patient vital signs to promptly respond to abnormalities. This requirement is refined by R2 and R3. R2 mandates that the Smart Bed (SB) must transmit temperature data to the Remote Terminal Unit (RTU) within a maximum latency of 1 Time Unit (TU), satisfied by the SB block. R3 highlights the RTU's role in analyzing temperature data and taking prompt action, requiring data processing within 3 TUs, either updating patient info or alerting the healthcare team, satisfied by the RTU block.

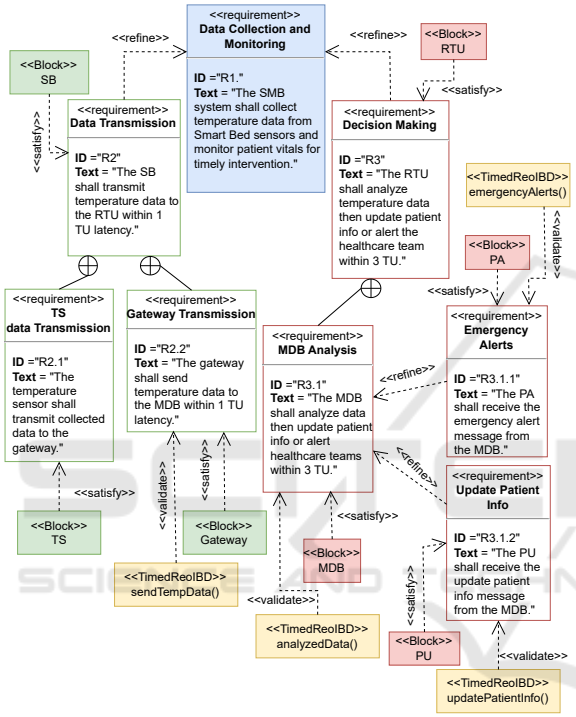

Figure 4: The requirement diagram of the SMB system.

4.2.2 ExtBDD

The ExtBDD diagram illustrates the hierarchical structure of the SMB system, with each component from the requirement diagram (Figure 4) represented as a block. These blocks detail a component's internal operations as private operations and its required and offered services. Each block includes two proxy ports: an input port for available services and an output port for required services.

In Figure 5 [A], the abstract overview of the system is depicted, emphasizing main components such as "SMB", subdivided into Smart Bed (SB) and Remote Terminal Unit (RTU), connected via composition.

Figure 5 [B] illustrates the concrete level of the SMB system, showcasing sub-components within primary components. For example, the smart bed component includes "Temperature Sensor" and "Gateway" blocks, with the former responsible for continuous data measurement, recording, and transmission to the Gateway.

4.2.3 Timed Reo IBD

In Figure 6, we compare the traditional internal block diagram of SysML with the Timed Reo IBD of Timed SysReo. The conversion from SysML IBD (Figure 6 [B-1]) to Timed Reo IBD (Figure 6 [B-2]) involves replacing SysML IBD connectors with "Timed Reo connectors", allowing for a more explicit representation of component internal composition and interaction protocols, including data and timing constraints.

SysML IBD connectors establish connections between compatible ends without managing connected entities, leading to "endogenous" coordination that complicates design, reduces reusability, and increases complexity, constraints, and project costs in CPS modeling. In contrast, Timed Reo IBD integrates coordination logic into connectors, enhancing reusability and simplifying component design, commonly used for model coordination in complex systems.

Timed Reo IBD provides a stakeholder-friendly diagram for visualizing CPS components and their timed interactions. It enhances reusability by combining simpler circuits at boundary nodes, easing the creation of complex circuits. Additionally, Timed Reo IBD offers an explicit means of expressing systems via Timed Constraint Automata (TCA) (Arbab et al., 2007), aiding accurate delineation and examination.

Timed Reo IBD enable CPS designers to precisely capture exogenous communication and synchronization patterns among components, enhancing system reliability, safety, and performance while potentially detecting errors early during development, thus saving time and costs.

In Figure 6[B](b,c), we illustrate the internal structure of the SB and RTU components, along with their timed interaction protocols. For instance, the gateway (gtw) component transmits the "sendTempData()" message to the medical database (MDB) component at $@t1 <= 1\ TU$. Subsequently, the MDB component analyzes the data at $@t2 <= 3\ TU$ and forwards the analyzed data to the xrouter component. Depending on the temperature status, the xrouter component decides whether to send an "emergencyAlerts()" message to the Patient Alert (PA) component or an "updatePatientInfo()" message to the Patient Update (PU) component. Thus, the Timed Reo IBD serves as a robust tool for modeling and analyzing systems, showcasing how Timed Reo connectors elucidate communication and coordination

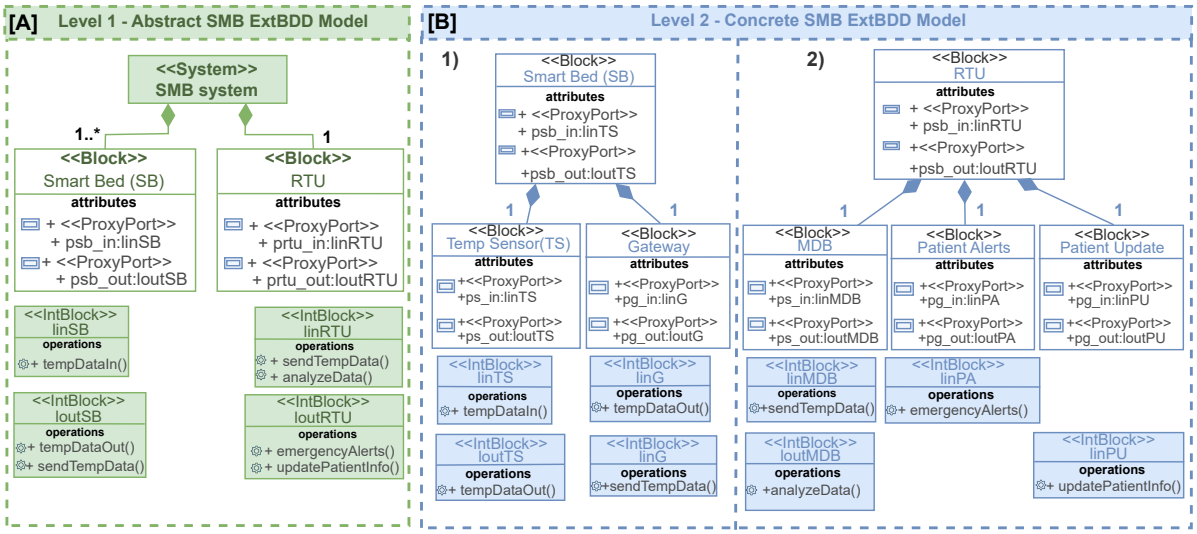

Figure 5: The ExtBDD model of the SMB system.

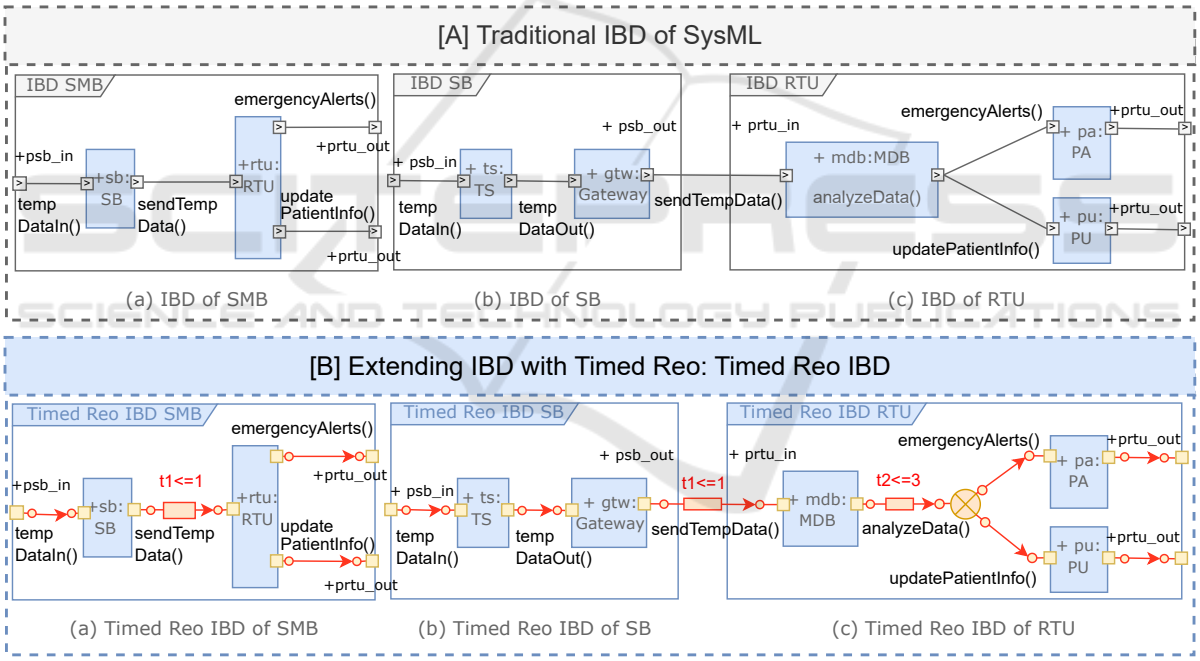

Figure 6: Comparison between a traditional SysML IBD and a Timed Reo IBD for the SMB system.

among components, resulting in a comprehensive representation of the system's structure and behavior.

Another notable aspect of Timed Reo IBD is its capability to validate predefined requirements, as depicted in Figure 4. For instance, the "R2.2" requirement is validated using a timed FIFO channel, coordinating the "sendTempData()" message from the gateway (GTW) component to the MDB component within a time constraint of $t_1 \leq 1$. Similarly, the "R3.1" requirement is validated through a timed FIFO channel, using the "analyzedData()" message that is transmitted from the MDB component to the Exclusive Router (EXR) within a time limit of $t_2 \leq 3$. Subsequently, upon entering the EXR router, the system dispatches either an "emergencyAlert()" message to the PA component if the analyzed data deviates from normal or an "updatePatientInfo()" message to the PU component if the analyzed data is within normal parameters.

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel incremental design approach exploiting Timed SysReo models that combines SysML and Timed Reo notations in order to enable a faithful modeling of CPS with timing constraints. Hence, we specified meta-models to define Timed SysReo language to formalize CPS architecture, and to detail timed connections between its components. We proposed to precise the meta-model with OCL, that imposes rules to be respected in order to ensure consistency between Timed SysReo models in our incremental design approach. Thanks to our Timed SysReo models, CPS designers can model all CPS facets and capture their timed complex coordination. Timed SysReo was illustrated by a Smart Medical Bed (SMB) example that has demonstrated the principal expressiveness and modeling conveniences. As future work we plan to: (1) Verify and validate temporal properties on Timed SysReo models. (2) Check our Timed SysReo models against time requirements and properties defined in the modeling steps using the UPPAAL (Bengtsson et al., 1996) tool.

REFERENCES

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

Arbab, F., Baier, C., De Boer, F., and Rutten, J. (2004). Models and temporal logics for timed component connectors. In *Proceedings of the Second International Conference on Software Engineering and Formal Methods, 2004. SEFM 2004.*, pages 198–207. IEEE.

Arbab, F., Baier, C., de Boer, F., and Rutten, J. (2007). Models and temporal logical specifications for timed component connectors. *Software & Systems Modeling*, 6:59–82.

Baier, C., Sirjani, M., Arbab, F., and Rutten, J. (2006). Modeling component connectors in reo by constraint automata. *Science of computer programming*, 61(2):75–113.

Barroso, S., Bustos, P., and Nunez, P. (2023). Towards a cyber-physical system for sustainable and smart building: a use case for optimising water consumption on a smartcampus. *Journal of Ambient Intelligence and Humanized Computing*, 14(5):6379–6399.

Bengtsson, J., Larsen, K., Larsson, F., Pettersson, P., and Yi, W. (1996). *UPPAAL—a tool suite for automatic verification of real-time systems*. Springer.

Bouskela, D., Falcone, A., Garro, A., Jardin, A., Otter, M., Thuy, N., and Tundis, A. (2022). Formal requirements modeling for cyber-physical systems engineering: An integrated solution based on form-l and modelica. *Requirements Engineering*, 27(1):1–30.

Cabot, J. and Gogolla, M. (2012). Object constraint language (ocl): a definitive guide. In *International school on formal methods for the design of computer, communication and software systems*, pages 58–90. Springer.

Delligatti, L. (2013). *SysML distilled: A brief guide to the systems modeling language*. Addison-Wesley.

Genius, D. and Apvrille, L. (2023). Hierarchical design of cyber-physical systems. In *Modelsward*.

Hause, M. et al. (2006). The sysml modelling language. In *Fifteenth European Systems Engineering Conference*, volume 9, pages 1–12.

Huang, P., Jiang, K., Guan, C., and Du, D. (2018). 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.

Kim, K.-D. and Kumar, P. R. (2012). Cyber–physical systems: A perspective at the centennial. *Proceedings of the IEEE*, 100(Special Centennial Issue):1287–1308.

Kokash, N., Jaghoori, M. M., and Arbab, F. (2013). From timed reo networks to networks of timed automata. *Electronic Notes in Theoretical Computer Science*, 295:11–29.

Mallet, F. (2015). Marte/ccsl for modeling cyber-physical systems. *Formal Modeling and Verification of Cyber-Physical Systems: 1st International Summer School on Methods and Tools for the Design of Digital Systems, Bremen, Germany, September 2015*, pages 26–49.

Tannoury, P. (2022). An Incremental Model-Based Design Methodology to Develop CPS with SysML/OCL/Reo. In *Journées du GDR GPL*, Vannes, France.

Tannoury, P., Chouali, S., and Hammad, A. (2022). Model driven approach to design an automotive cps with sysreo language. In *Proceedings of the 20th ACM International Symposium on Mobility Management and Wireless Access*, pages 97–104.

Tannoury, P., Chouali, S., and Hammad, A. (2023). Joint use of sysml and reo to specify and verify the compatibility of cps components. In *International Conference on Formal Aspects of Component Software*, pages 84–102. Springer.

Tartarisco, G., Cicceri, G., Bruschetta, R., Tonacci, A., Campisi, S., Vitabile, S., Cerasa, A., Distefano, S., Pellegrino, A., Modesti, P. A., et al. (2024). An intelligent medical cyber–physical system to support heart valve disease screening and diagnosis. *Expert Systems with Applications*, 238:121772.

Xie, J., Tan, W., Yang, Z., Li, S., Xing, L., and Huang, Z. (2021). Sysml-based compositional verification and safety analysis for safety-critical cyber-physical systems. *Connection Science*, pages 1–31.