

System Analysis and Modeling Using SysML

Muzaffar Iqbal, Muhammad Uzair Khan and Muhammad Sher

Abstract In software engineering, Unified Modeling Language (UML) is considered as the de-facto standard for modeling Object Oriented Systems. On the other hand when it comes to system engineering, then UML is believed to be not as good. More precisely, UML is not effective when modeling system's (Non-functional) requirements, linking these non-functional requirements with other artifacts of the system and defining constraints on the system, in an effective manner to define the system architecture. OMG (Object Management Group) released SysML (a UML-profile) to overcome such limitations of UML, when applied to system engineering. This paper is an effort to show how system's structure, its constraints and (non-functional) requirements can be effectively modeled and linked with each other in SysML with help of a case study.

Keywords Unified modeling language (UML) · System modeling language (SysML) · System engineering · Software engineering

M. Iqbal (✉) · M. Sher
Department of Computer Science and Software Engineering,
International Islamic University, Islamabad, Pakistan
e-mail: muzaffar.iqbal123@yahoo.com

M. Sher
e-mail: m.sher@iiu.edu.pk

M. U. Khan
Department of Computer Science, National University of Computer
and Emerging Sciences, NUCES-FAST, Islamabad, Pakistan
e-mail: uzair.khan@nu.edu.pk

1 Introduction

UML is considered the best choice while modeling software systems but is considered inadequate for modeling system engineering [1–3, 4]. Considering the inefficient constructs of UML for system engineering, OMG (Object Management Group) has released a new Domain Specific Language (DSL) called SysML [5, 6]. SysML (System Modeling Language) is designed to model complex systems specifically modeling system requirements and parametrics.

SysML is based on UML 2. Some diagrams of UML 2 (e.g. Object Diagram, Communication Diagram, Components Diagram, Deployment Diagram, Interaction Overview Diagram, Profile Diagram) are not included in SysML [7]. SysML has introduced some diagrams which were not present in UML 2 (e.g. Requirement Diagram and Parametric Diagram). Some diagrams of UML 2 are reused without any modifications and some are reused after some modifications.

Figure 1 shows the broad relationship between UML 2 and SysML.

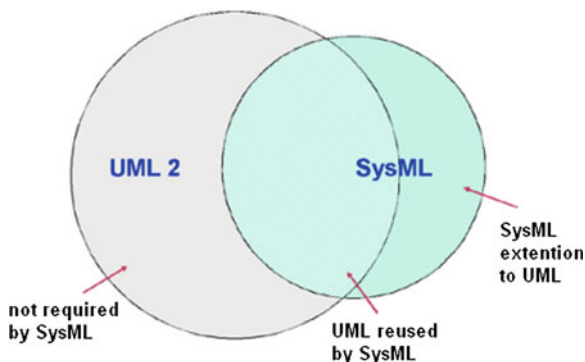
Figure 2 below depicts the various diagrams of SysML. It also shows the diagrams reused from UML 2 with or without modifications. The diagrams new to SysML are also shown in the following figure.

The primary structural unit in SysML is «block». SysML uses this structural unit to represent different artifacts of the system including hardware, software, personnel or any other facility. Block Definition Diagram (BDD) and Inter Block Diagram (IBD) are used for defining the basic system structure [5, 8].

SysML introduces a diagram known as Requirement Diagram. SysML Requirement Diagram can be used to model functional as well as non-functional requirements and their links with each other and other artifacts of the system to show that how they are satisfied.

In this paper, we present the case study of traffic controller to discuss the structure of the system using SysML. We first draw the system's requirement diagram to show the functional and non-functional requirements of the system and show the connection of different requirements with different parts of the system. We present the requirements of the system by using SysML requirement diagram.

Fig. 1 Relationship between UML and SysML



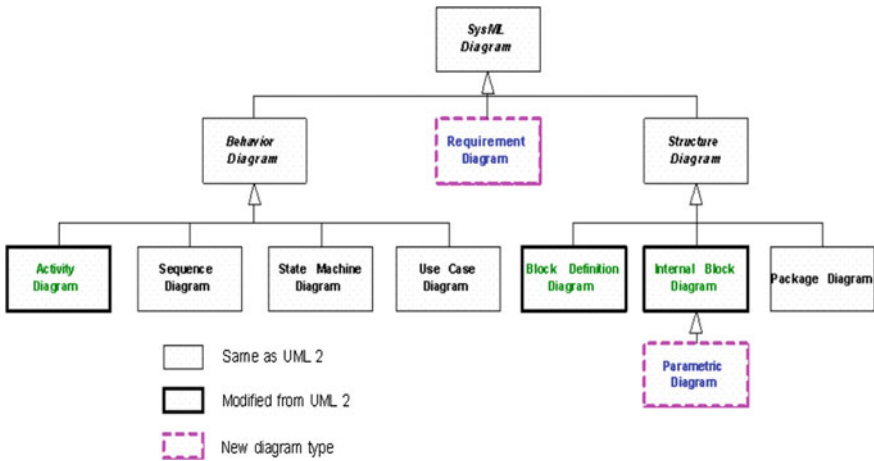


Fig. 2 SysML diagrams (courtesy [5])

After representing the external entities and their relationship with the system, the system’s structure will be constructed by using SysML Block Definition Diagram and Parametric Diagram.

Here is the paper’s structure. Section 2 discusses the selected case study. Section 3 discusses system’s requirements and SysML Requirement Diagram. Section 4 introduces SysML Block Definition Diagram (BDD) and its application to our case study. Section 5 describes SysML Parametric Diagram and its related application to our selected case study. Finally we provide conclusion in Sect. 6.

2 Case Study

The selected case study for this work is based on [9]. This case study is about a traffic controller. The traffic controller is responsible to control traffic signals at a traffic junction as shown in Fig. 3. Traffic lights 1 and 3 are required to always show the same signal as shown by lights 2 and 4 at a particular time.

One traffic cycle is Green–Orange–Red and another cycle is Green–Orange–Red. The safety requirement is that one pair of traffic lights must be red for 30 s and another pair of traffic lights must be green at any given time. The controller is responsible for ‘change direction’. The currently red signals must be set to green, and the currently green signals to red in response to ‘change direction’.

We will be using OMG SysML to model some of the major components of traffic controller discussed in this case study. The focus of this paper is using Requirement Diagram and Parametric Diagram of SysML to represent system requirements and its constraints respectively. Additionally we will show the structure of traffic controller by using Block Definition Diagram (BDD).

Fig. 3 Layout of traffic lights (reproduced from [8])

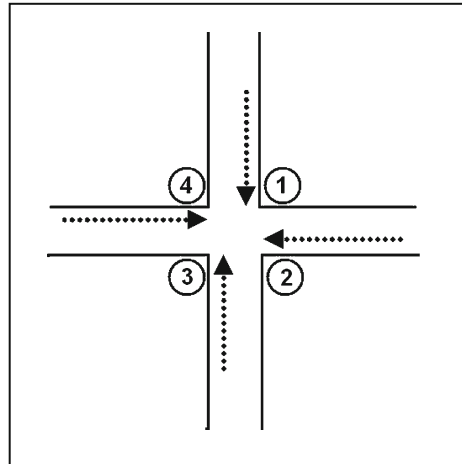
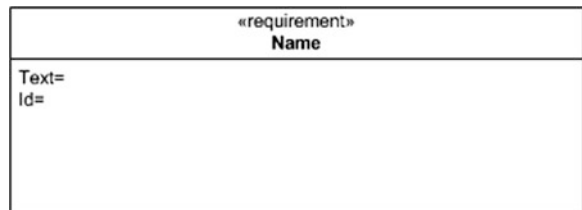


Fig. 4 Requirement diagram



3 SysML Requirement Diagram

UML is considered inefficient to model non-functional requirements [4]. To remove this inefficiency of UML, SysML introduces Requirement Diagram which can model and represent non-functional requirements as model element of the system structure.

Requirement Diagram of SysML has requirement name, requirement text and optional unique ID as shown in the Fig. 4.

Requirements and their relationships can be represented in a number of ways while using Requirements Diagram [6]. The relationships among different requirements of the system and their hierarchy can be easily shown. The relationships: hierarchy, satisfy, refine, verify, trace etc. can be used while requirements are being modeled. In SysML, requirements can also be represented in a tabular form [5] but is beyond the scope of the paper.

In the case study we have selected, different requirements (both functional and non-functional) exist. Some of the requirements are as below.

3.1 Software Requirements

The intended system to be developed in our selected case study is a traffic controller at a road junction (Fig. 3). Pairs of traffic lights (1, 3) and (2, 4) are required to show the same signal indication for a specified interval. There are two light cycles Green–Orange–Red and Red–Orange–Green.

There are two more (non-functional) requirements: safety and reliability. In order to ensure safety, it is required that one pair must be red at certain given time.

The reliability requirements ensure that:

- Lights of all directions must not be red at the same time
- Lights of all directions must not be green at the same time
- Continuous power supply is provided.

The above stated requirements can be modeled using SysML Requirement Diagram as shown in Fig. 5.

In Fig. 5 we can see that how flexible and powerful is the SysML Requirement Diagram. It enables us to show the requirements as a standardized model. The figure shows that how the traffic controller reliability (NFR) is contained by other

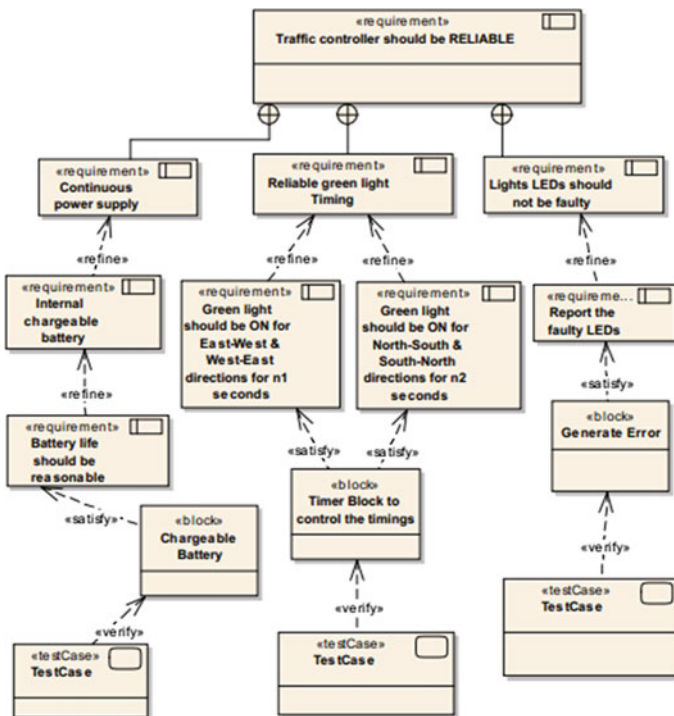


Fig. 5 Requirement diagram of the case study

requirements. It means that the system will be called reliable when all the following (contained) requirements are satisfied.

Figure 5 also illustrates some of the relationships like satisfy and refine. For example, ‘Reliable green light Timing’ requirement is refined by two requirements. On the other hand, ‘TrafficController’ block satisfies some of the requirements.

4 SysML Block Diagram

4.1 Block

In object orientation, classes are the central elements while in SysML classes are replaced by the blocks [10]. Blocks have the capability for modeling different type of systems and its features including both structural and behavioral [5]. Through the concept of block, SysML describes the static structure of systems. A block might be physical or logical unit of a system [10].

4.2 Block Definition Diagram (BDD)

The SysML Block Definition Diagram (BDD) in SysML is the simplest way to describe the system’s structure. It also provides a number of features like associations, dependencies and generalizations among blocks. Just like UML class, SysML BDD also uses properties and operations in order to define a static feature of the system [5].

The general structure of SysML Block Definition Diagram (BDD) is given in Fig. 6. BDD is equivalent to Class diagram in UML but offers more powerful features than class diagram of UML. The Block Definition Diagram (BDD) of our case study is given in Fig. 7. This BDD depicts the composition of a block by relating blocks with one another using the composition relationship. The figure shows that the Traffic Controller is composed of a number of subsystems; Power, Processing and TrafficLight. Each subsystem is modeled by a separate block while each block is subsequently decomposed in sub-blocks.

5 SysML Parametric Diagram

SysML has introduced another diagram which was not present in UML called ‘Parametric Diagram’. Parametric diagram works with association with a diagram called ‘Constraint Block’ [11].

Fig. 6 SysML block definition diagram (BDD) (courtesy [5])

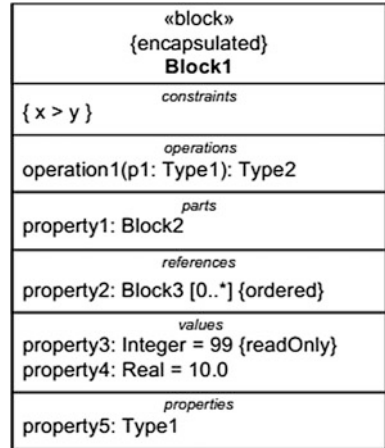
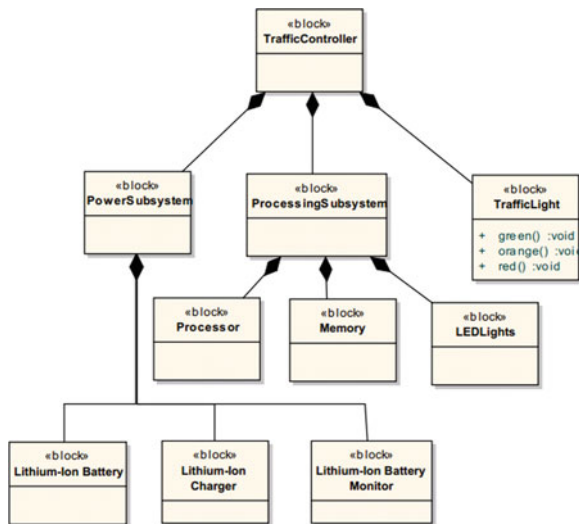


Fig. 7 Block definition diagram of traffic controller



5.1 Constraint Block

Constraint blocks can be used to present a way for integrating different system engineering analysis e.g. performance and reliability models with other SysML models [5]. By using Constraint block we can define a number of constraints and rules (based on these constraints) that must be conformed by the system.

Formulated constraints are regular UML constraints [9] and are usually written by using OCL which is recognized and evaluated (automatically) by almost all the tools that support SysML.

A constraint block, as shown in Fig. 8, defines the parameters used by the constraints like attributes. The notation used by constraint block is very similar to

that for blocks with the difference that stereotype «constraint» is used instead of «block» . A constraint block also contains another compartment used to write constraints using the parameters defined in the ‘parameters’ compartment. Figure 8 shows the constraint block of our case study.

Figure 8 shows that how constraint’s parameters and constraint’s formulae can be defined by using SysML constraint block. The list of constraints can be large but the figure shows some of the primary constraints related to our case study.

5.2 Parametric Diagram

Once constraints are defined then they can be applied to the system by using parametric diagram. A parametric diagram is a special kind of SysML Internal Block Diagram (IBD) that shows the usage of constraint blocks along with the properties they constrain within a given context [5, 9]. Parametric diagrams are based on one or more constraint blocks. Figure 9 shows the parametric diagram of our case study based on the constraint block shown in Fig. 8.

A parametric diagram could be very complex and as simple as demonstrated in Fig. 9. The use of a constraint block on a parametric diagram can be shown as round-cornered rectangles as shown in Fig. 9 above. Small rectangles at the inside edge of the parametric diagram represent constraint parameters. They also provide connection points while linking them to other constraints or parts.

It is important to note that although we can define constraints (in form of OCL) using constraint block but some SysML supported tools still encourage to write the OCL constraint in form of scripts or comments. This allow us to write more complex constraints which would be difficult to write or show using constraint block.

Once constraints are defined then most SysML supported tools allow us to draw parametric diagram automatically based on the available scenario (constraint block).

Fig. 8 Constraint block of traffic controller

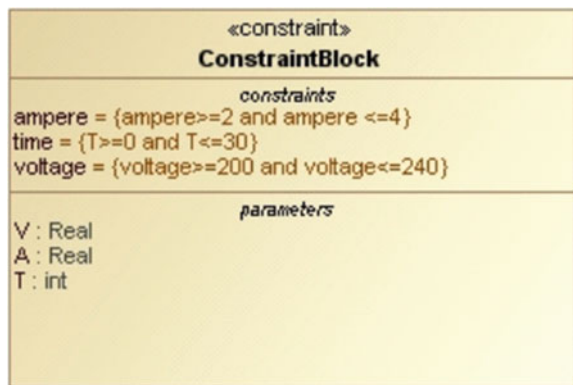
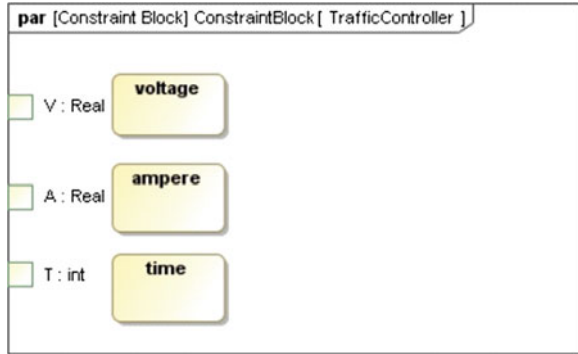


Fig. 9 Parametric diagram of traffic controller



The generated parametric diagram can be further customized to meet the requirements.

It is important to note that after construction of parametric diagram some code is required to write to accomplish the task. This code can be written in an appropriate language like VHDL, Verilog etc. depending upon the system requirements and the available facilities of the language.

6 Conclusion

In this paper, we have shown how SysML Requirement Diagram, BDD (Block Definition Diagram) and Parametric Diagram can be exercised to model the system's structure. SysML Requirement Diagram is one of the powerful SysML diagram valued by both software and system engineers. As we have seen that Requirement Diagram has made the life of engineers very simple while modeling non-functional requirements as system construct and linking them with other parts of the system. It also helps us to analyze whether a particular requirement is satisfied or not. At the other hand, BDD is used to model the system's structure just like class diagram is used in UML. The paper also illustrated that how Parametric Diagram can be used in association with constraint block to define constraints of the system. Since there are only a few case studies available on SysML, this effort will provide a basis of understanding of SysML for modeling complex systems.

References

1. Cantor M (2003) Panel: Extending UML from Software to Systems Engineering. In: Proceedings of the 10th IEEE international conference and workshop on the engineering of computer-based systems, IEEE, Huntsville, Alabama
2. www.omg.org/syseng/

3. Pettit RG (2004) Lessons learned applying UML in embedded software systems designs. In: IEEE
4. Vanderperren Y (2005) UML 2 and SysML: an approach to deal with complexity in SoC/NoC design. In: Proceedings of design, automation and test in Europe
5. OMG Systems Modeling Language, The Official OMG SysML Specification Site. www.sysml.org/
6. www.omgsysml.org/
7. <http://www.sysmlforum.com/faq/>
8. Fonoage M (2008) SysML, UML, and SDL: a comparison. Center for systems integration. Florida Atlantic University, Boca Raton
9. Lano1 K, Evans A (1999) Rigorous development in UML. In: Proceedings of the second international conference on fundamental approaches to software engineering, Springer, London
10. Weilkie T (2007) Systems engineering with SysML/UML: modeling, analysis, design. The MK/OMG Press, USA
11. Holt J, Perry S (2008) SysML for systems engineering. Professional applications of computing series 7. The Institution of Engineering and Technology, London, UK