# Mapping SysML to modelica to validate wireless sensor networks non-functional requirements

S. Berrani, Ahmed Hammad, Hassan Mountassir

# Mapping SysML to Modelica to Validate Wireless Sensor Networks Non-Functional Requirements

Samir Berrani
*Femto-ST Institute*
*University of Franche-Comté*
*Besançon, France*
*Email: samir.berrani@yahoo.fr*

Ahmed Hammad
*Femto-ST Institute*
*University of Franche-Comté*
*Besançon, France*
*Email: ahmed.hammad@femto-st.fr*

Hassan Mountassir
*Femto-ST Institute*
*University of Franche-Comté*
*Besançon, France*
*Email: hassan.mountassir@femto-st.fr*

***Abstract*—Wireless Sensor Networks (WSN) have registered a large success in the scientific and industrial communities for their broad application domains. Furthermore, the WSN specification is a complex task considering to their distributed and embedded nature and the strong interactions between their hardware and software parts. Moreover, most of approaches use semi-formal methods to design systems and generally simulation to validate their properties in order to produce models without errors and conform to the system specifications. In this context, we propose a Model Driven Architecture (MDA) approach to improve the verification of the WSN properties. This approach combines the advantages of the System Modeling Language (SysML) and the Modelica language which promote the reusability and improve the development process. In this work, we specify a model transformation from SysML static, dynamic and requirement diagrams to their corresponding elements in Modelica. Thanks to the SysML requirement diagram which is transformed into Modelica properties (constraints), we propose a technique using dynamic tests to verify WSN properties. We have used the Topcased platform to implement our approach [1] and chosen a crossroads monitoring system which is based on wireless sensors to illustrate it. Besides, we have verified and validated some wireless sensors properties of the studied system.**

***Keywords*-Specification, SysML, Modelica, Design, Virtual verification, Model transformation, MDA, WSN.**

## I. Introduction

Since the beginning of 2000's, the WSN have been adapted to several application domains; therefore, scientific and industrial actors are interested to them. However, this kind of networks is mainly used in the observation of physical phenomena in a restricted environment. They are characterized by simple deployment and low production costs. Furthermore, the WSN industry actors must develop the modeling methods of these networks in order to maintain the competitiveness of their products. In addition, modeling WSN is equivalent to design distributed and embedded system conjointly.

Moreover, formal methods are generally used in the modeling of critical systems that require rigorous verification. This reduced application area is due to the nature of these methods which require good skills in mathematics. As a result, industrial actors have widely adopted semi-formal methods to design WSN applications. These methods are based mainly on semi-formal languages (text or graphic) for which are defined a precise syntax and relatively weak semantics. Besides, the semi-formal languages are firstly simple to understand, on the other hand they provide a rich structuring mechanism allowing the reduction of the design time and cost.

However, in order to improve the checking of the WSN non-functional properties which is done largely by simulation in domain of industry, we propose an MDA approach to design and check the WSN energy consumption. Our approach is based on the SysML language which will be extended to an executable language namely Modelica. Considering the Modelica characteristics, we propose a virtual verification of the properties deduced from the SysML requirement diagram. This approach includes the benefits of the SysML modeling and the possibility to simulate and verify the modeled system properties by Modelica. In other words, this modeling approach allows us to reduce the design time, decrease the design cost and increase the efficiency of WSN models.

This paper is structured as follows: first we discuss existing works related to the WSN modeling. Then, we give brief introductions to WSN, SysML and Modelica. After that, we explain our approach to design and verify the WSN energy consumption property. Finally, we conclude this work.

## II. Related works

As mentioned in the previews section, the WSN modeling and checking methods are widely studied by scientific and industrial communities.

Concerning the modeling, N. Belloir and al. [1] carried out a comparative study between UML2 (Unified Modeling Language 2.0) and SysML related to the design of WSN applications. Other studies have focused on improving code reusability of WSN protocol. K. Klues and al. [2] proposed a modeling of the MAC layer using Component-Based Architecture (CBA). C. T. Ee and al. [3] introduced a modular network layer to allow the co-existing (stacking) protocols with the aim to reduce code and consumed resources at run time. In these both works, the authors focused on implantation techniques and not on the system design.

In addition, I. Khemapech and al. [4] considered that simulation is an interesting technique to study the WSN. For this, they reviewed several researches on the development of simulators for WSN. E. Cheong and al. [5] developed a Framework that provides a graphical environment for the modeling and simulating of WSN

[1]This research is supported by the Regional Council of Franche-Comté with the SyVad project (http://syvad.univ-fcomte.fr).

applications using mainly TinyOS. Another similar work was proposed by M. M. R. Mozumdar and al. [6] released with Simulink. This Framework provides the ability to make a performance analysis of the designed system through simulation, and it also allows code generation which is compatible with TinyOS. One more work was presented by D. Riley and al. [7] defined a method to integrate Simulink and ns-2 for the hybrid networked control systems. However, the behavior of any node sensor in the networks is defined closely with communication protocols, which limits the reusability. In addition, the verification of WSN properties is done through simulation.

Besides, M. Previstini and al. [8] proposed an approach to design Systems On Chip (SoC) which included the WSN. For this, they designed a SysML profile for modeling SoCs. Then, they carried out a model transformation from SysML model to SystemC model according to the MDA standard in order to make a simulation. The proposed model transformation is not completely automated. Moreover, S. Villa and al. [9] proposed an approach for modeling WSN using UML and SystemC. They defined two UML profiles, namely: UML-SystemC in order to include (in UML) the SystemC specific concepts and the UML-Marte profile which allows hardware-software modeling (co-design). Then, they designed a Framework that enables the model transformation between UML model and SystemC model. The verification of SoC properties is done by simulation.

Another work was presented by F. Losilla and al. [10] proposed an MDE (Model Driven Engineering) approach to develop WSN applications. They defined three levels of abstraction; these allow designers to specify: the domain-specific model, the component-based architecture descriptions and finally, the plateform-specific model. Next, the authors developed a Framework that enables the model transformation from an UML model (using the WSN domain-specific language) to a nesC model that enables the simulation of this designed system. In the same context, P. Boonma and al. [11] developed the Framework "Moppet" based on model-driven performance engineering. It allows the WSN architects to design applications using the proposed WSN libraries and to estimate their performances (energy consumption and sensor node lifetime).

Moreover, L. Samper [12] proposed a formal approach for WSN modeling. It is based on virtual prototyping to reduce the system's complexity. Therefore, the author devised WSN applications in several abstraction levels, namely: the application layer, the network layer, the MAC (Media Access Control) layer, the hardware layer and the environment layer. Then, he modeled each level by the transition systems (automata). However, this approach requires skills in transition systems and its appropriate languages.

Our WSN modeling approach is based on the MDA standard. It allows a semi-formal modeling with the SysML language which will be extended to the Modelica in order to make verification of the WSN energy consumption. In other words, our approach offers the possibility to design clear models and allows also the simulation and the virtual verification of designed system properties. Furthermore, our approach is applicable to the WSN as shown in the case study and also to any physical system such as the SoC. This work was presented briefly in [13].

## III. Description of WSN modeling, SysML and Modelica

### A. WSN modeling

The WSN modeling is influenced by many factors, which include fault tolerance, scalability, production costs, operating environment, sensor network topology, hardware constraints, transmission media and power consumption. These factors are important because they serve to direct the design of the WSN protocols and algorithms. In addition, they can be used to compare different WSN architectures. Moreover, the prospects of WSN applications are promising but the challenges that they present are not less numerous and not less complex too. Among the crucial issues, which the WSN non-functional properties represent, we can mention the energy consumption, the automatic configuration, the communication security, etc.

In this project, we study the WSN energy consumption which is heavily dependent on the type of node. These nodes are designed with the aim to maximize their life expectancy. In [14], the authors defined an energy consumption model for WSN. The major feature of this model is its accuracy in estimating the energy consumption. Therefore, this model allows the estimation of the overall lifetime of the WSN accurately. For this reason, we adopted this energy consumption model in our study.

### B. SysML language

The System modeling language (SysML) is based on UML2. It replaces the class concept in modeling by block vocabulary which is more suited to the Systems Engineering (SE). A block includes any software, hardware, data and processes concepts. The SysML does not use all UML2 diagrams, it provides a reduced set of diagrams adapted to the SE. The software designers have several advantages to work with the SysML because there are many similarities with UML2. The SysML allows heterogeneous teams to work together for unique modeling system by creating hardware blocks and software blocks. Furthermore, knowledge is modeled in a single repository that improves communication between different teams. The SysML is based on nine diagrams, each one of them is dedicated to represent particular concepts of studied systems. As illustrated in figure 1, these diagrams are divided into three main groups as follows: shared diagrams with UML2, modified UML2 diagrams and new SysML diagrams.

### C. Modelica language

The Modelica is an object-oriented modeling language that allows the modeling of physical systems which can be complex and heterogeneous. It can be considered as a multidisciplinary modeling language [16]. The Modelica is an open language which is developed and promoted by the Modelica Association. The Modelica models are described mathematically in acausal way through differential equations, algebraic equations and discrete equations.

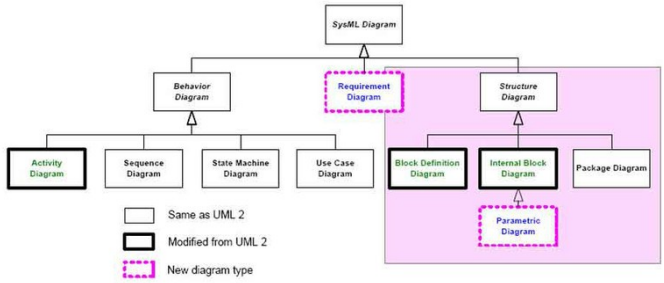


Figure 1. SysML diagrams

The Modelica solvers contain very effective algorithms for solving equation systems which allow the handling of complex models that are described by thousands of equations. The main benefit of Modelica is the reusing of model components that simplify the modeling task to designers.

*1) The motivations of the chosen language (Modelica):* In order to execute the SysML models, our choice was made on the Modelica language for the following reasons:

- The Modelica language allows an object-oriented modeling approach;
- The Modelica language allows a graphical representation of the model;
- The Modelica language considers at any time the differential equations, the algebraic equations and the discrete equations which describe the model, as single system equations;
- The Modelica language has only one simulator that allows automatic synchronization between discrete and continuous parts;
- There is a close similarity between the Modelica model and the B (B formal language) model which allows formal proofs (after transformation to B model);
- H. Lundvall and al. [15] proposed an algorithm to make transformation from the Modelica model to the Hytech model in order to do a formal verification (model checking).

## IV. MDA APPROACH FOR MODELING WSN

The OMG (Object Management Group) has revolutionized the software industry by providing the MDA standard. This standard describes a new approach for designing computer applications. It introduces a separation between business logic and implementation logic (technical platform). This approach is defined in the Model Driven Engineering (MDE). The designers have observed that the business logic does not change during the time contrary to the technical architecture. As a result, it becomes evident to make a separation between the business logic and the technical logic in order to reduce the system's complexity, maintenance cost and the technology migration cost. This transformational approach changes the active role of the developers in building computer systems that is defined in the classical approach to a simplified and less involved role thanks to automated construction. The MDA approach proposes the definition of business models independently of any technical platforms and it allows the code generation automatically to the chosen platform.

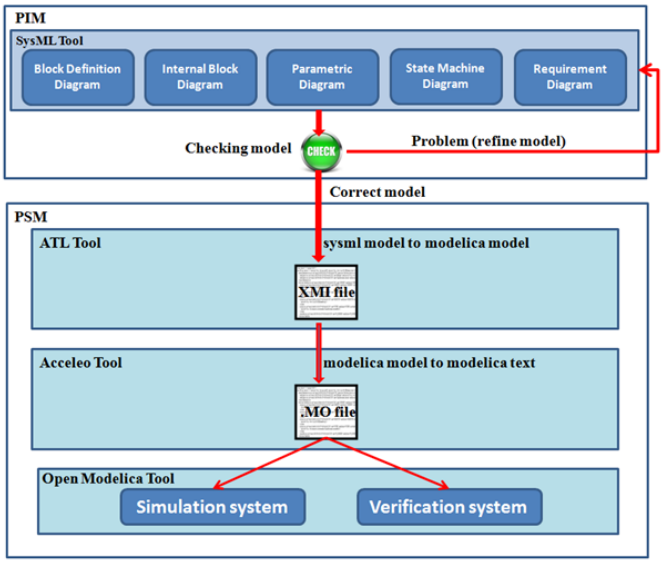


Figure 2. Adopted approach

### A. The adopted approach

The SysML is a modeling language for system engineering which doesn't allow the model execution. Therefore, we can't make the verification of system properties. For this, we propose to transform the SysML model to an executable model, such as the Modelica model. The methodology adopted in our project is constituted of four steps. In the first stage, the modeler designs the WSN application with the SysML language which is recognized as a Platform Independant Model (PIM). In the next phase, the designer is invited to run a check of its model in order to refine it when there are problems reported by the analyzer. Among the problems identified by the analyzer, we list, the undefined elements (initialization, typing, etc.) which are generally due to forgetfulness and the elements which are not considered by the model transformation (SysML model to Modelica model) rules. The objective of this analysis is to ensure that the generated code at the end of the transformation process is executable by the OpenModelica compiler. After this check, we can undertake the third step that defines the transformation from the SysML model to the Modelica model (PIM to Platform Specific Model - PSM). The file resulting from this transformation is in XMI (XML Metadata Interchange) format which is conformed to the proposed Modelica meta-model 3. Finally, for the last step, we transform the Modelica model to the Modelica code (PSM to PSM). The input file of this transformation (model to text) is the XMI file resulting from the last transformation (SysML model to Modelica model) and the output file is the Modelica code. Figure 2 illustrates the process followed in our project.

*1) The Modelica verification and validation techniques:* The Modelica language considers all differential equations, algebraic equations and discrete equations, as a single equation system. Synchronization between these different types of equations is handled automatically by the associated simulator. In addition, Modelica allows to express in each model or sub-model (class: model, block) the invariants and also one or more safety constraints, in

general, with the instructions ”Assert” and ”Terminate”. The ”Assert” instruction evaluates the boolean expression that represents the safety constraint at each elementary cycle of simulation, if it is true the constraint is preserved, otherwise it raises an error or a warning according to the configuration of ”Assert” instruction. Regarding the statement ”Terminate”, it evaluates its boolean expression after the simulation is completed, if it is violated, an error is generated. The reference work about the verification in Modelica is illustrated in the approach proposed by W. Shamai [17].

*2) The SysML metamodel:* The SysML modeling tool that we have adopted is Topcased. This tool provides a SysML meta-model which is based on a UML metamodel. We preferred to use the SysML metamodel existing in the Topcased because it is more complete and adequate with all SysML model edited by this tool. However, any SysML model represented by Topcased is consistent with its integrated metamodel.

*3) The Modelica metamodel:* The Modelica language is based on object-oriented paradigm. Any element of this language is an instance of a class. In other words, the basic structure in Modelica is the entity class. However, this language is recognized by its flexibility. It defines special keywords that replace the word class. These keywords are used in different contexts according to special restrictions, for example:

- The package: It is mainly dedicated to manage the namespace. The declaration of variables and parameters isn't allowed.
- The connector: It allows the specification of communication nature between objects (blocks). The equation definition isn't authorized in the connectors.
- The block: It is used to define components. All declared variables must be prefixed by input or output keywords (causality). A block should not be used in the connection definition.
- Type: It allows the definition of new types (simple or complex) and also enumerations.
- Record: It allows the data structure specification. The equations are not permitted in the record structure. In addition, it doesn't define a connection.
- Partial class: It allows the structures or the common behavior generalization among several model entities.

The proposed metamodel recognizes all these structures, variables and parameter constructions. Furthermore, this metamodel includes the behavioral model description such as equation and algorithm statements. Moreover, it allows the depiction of the model invariants or safety constraints. Figure 3 illustrates the proposed Modelica metamodel.

*4) The model transformation tools:* The used tool to carry out model transformations is the ATL (ATLAS Transformation Language). This language is based on the standard QVT (Query View Transformation) defined by OMG. It is available as a plugin in the Eclipse project. The official website of Eclipse contains more informations about ATL language, its environment and its model transformation libraries.

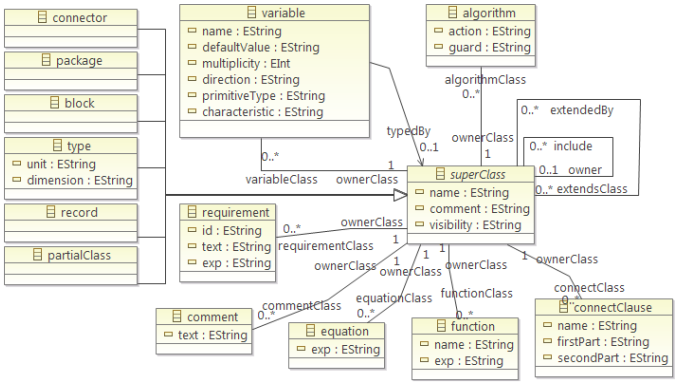


Figure 3. The Modelica metamodel

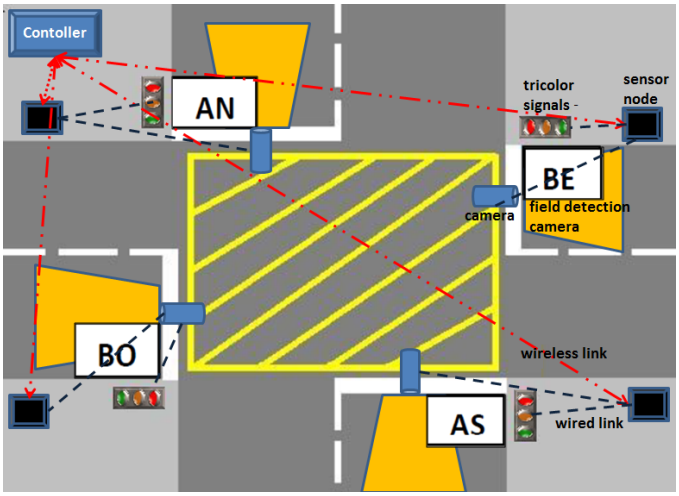


Figure 4. Crossroads system environment

## B. The case study

Motorized traffic density in urban areas requires the establishment of signaling traffic laws to improve safety and fluidity. The toughest traffic problems are at road intersections. In fact, the passage priority associated with eventual changing direction could create bottlenecks. The solution adopted by traffic operators to regulate circulation is signalized systems(tricolor and bicolor lights).

The traffic lights installed at intersections are used to adjust the vehicle movements. They are managed by a system that synchronizes the color changes of the different junction lights. The traffic-light colors are managed by a controller which depends on the number of cars waiting to cross the junction. The duration of a cycle lights (yellow - red - green) and the time of each phase is defined by the traffic center of the city. This center supervizes all street intersections. Figure 4 illustrates the crossroads environment.

*1) hypothesis:* In this work, we focused on the WSN energy consumption. For that, we must simplify our case study in order to concentrate on the study of this parameter. The retained hypotheses which allow us to identify and simplify the studied system are as follows:

- The sensors that control traffic lights on each road lane are powered by a battery;
- The sensors used in the lanes serve to control the traffic lights and communicate with the controller;

- The traffic light colors are yellow, red and green;
- The system failures are not treated in order to simplify the case studied here;
- The video traffic detection camera and the tricolor signal lights are powered by the electricity network;
- The video traffic detection (camera) detects the vehicles at a distance which is fixed during the equipment installations;
- The camera sensor can estimate the vehicle numbers waiting to cross the intersection;
- The pedestrian crossing and the communication between the controller and the traffic center aren't studied in order to simplify the case studied here.

*2) The technical system studied in this case:* We assume that the intersection of our case has two roads (A and B) in both directions. The four traffic-light units are appointed such as: AN, AS, BO and BE, according to figure 4. The system specifications are as follows:

- The system must be safe :
  - The traffic lights AN and AS are in the same color and it is the same for BO and BE;
  - When AN and AS are green, BO and BE are red;
  - When AN and AS are red, BO and BE are green or yellow;
  - When the controller sends a command to change color, all junction lights should change their color simultaneously.
- The system must be efficient :
  - When the light flashes on green, it changes to the yellow color only if there are oncoming vehicles on the other road and its light time is completed;
  - The transmission between a sensor node deployed in a lane and the controller is via the communication protocol ZIGBEE (IEEE 802.15.04);
  - All sensor messages pass through a controller;
  - The duration of the traffic signal depends on the number of vehicles waiting on each road. This duration should be long enough for cleaning the queue.
- The system must be economic : The energy consumption should be minimized.
- The system must be compatible with traffic laws : The control design should be according to the current traffic laws.

### C. SysML modelisation

*1) The requirement diagram:* This diagram is used to represent the requirements of the designed system. From the case study, we have identified two main requirements which relate to the safety and the longevity of the system. Figure 5 shows the requirement diagram of monitoring junction system.

*2) The block definition diagram:* The block definition diagram allows to give a structural description of the studied system. The main block that represents the crossroad monitoring system consists of a controller unit, four extended sensor nodes that manage the traffic at the lane and finally a phenomenon that represents the arrival of vehicles. Figure 6 shows the global system structure.

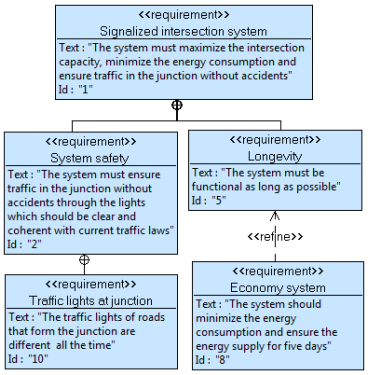


Figure 5. Requirement diagram

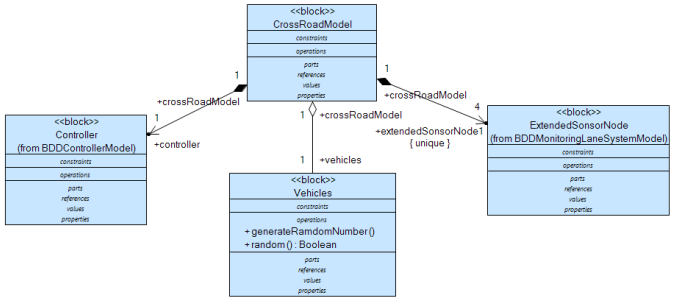


Figure 6. Global system structure

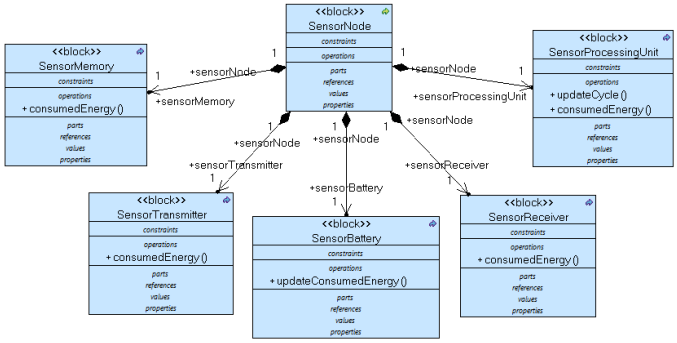


Figure 7. Sensor node structure

This extended sensor node consists of a simple sensor node, an additional sensing unit "camera" and an actuator "traffic lights". The simple sensor node block consists of a memory unit, a processing unit, a transmitter unit, a receiver unit and a battery. Figure 7 shows the simple sensor node structure.

*3) The internal block diagram:* The internal block diagram (IBD) is a white box view of a block. It describes the system internal structure in terms of parts, ports and connectors. These parts are assembled by connectors that connect their ports. In our case study, the simple sensor node is a basic internal block diagram. This diagram describes the sub-blocks which constitute the sensor node block and the communication type between them and the external blocks.

*4) The parametric diagram:* The parametric diagram is a SysML specific modeling technique that allows the integration of constraints or equations into the model in the analysis aims. These constraints are defined by parameters and rules which describe the evolution of these parameters

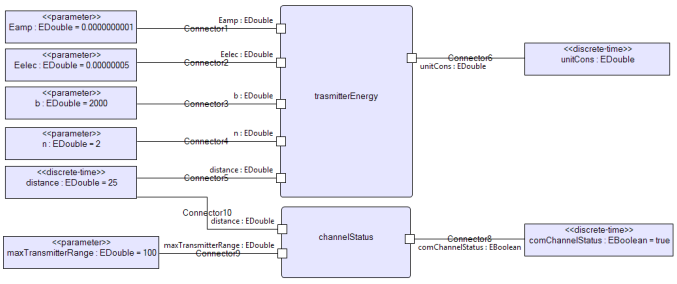


Figure 8. Transmitter parametric diagram


Figure 9. Receiver parametric diagram

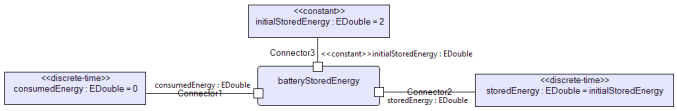


Figure 10. Battery parametric diagram

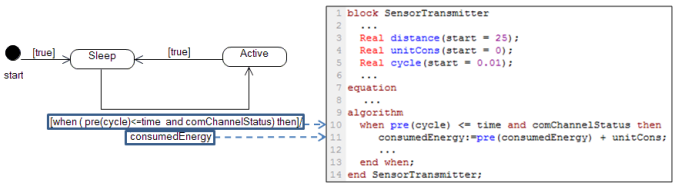


Figure 11. Tansmitter state machine diagram

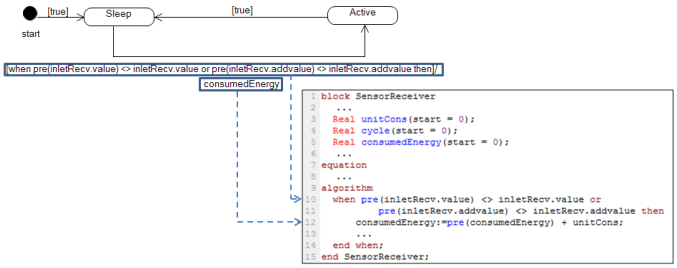


Figure 12. Receiver state machine diagram

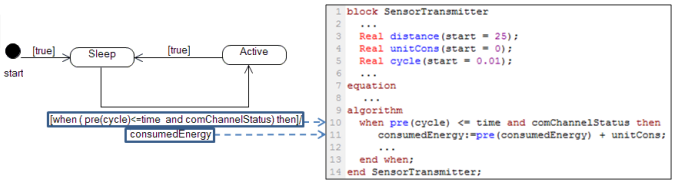


Figure 13. Processor state machine diagram

related to each other. The main objective of our project is to study the WSN energy consumption. For this, we adopted the energy consumption model proposed in [14]. In this model we can distinguish several sources of energy consumption, such as the processor, the memory , the transmitter, the receiver, the sensor and the actuator. We assume that the actuator (traffic light unit) and the sensor unit (additional sensing unit - camera) are supplied by the electricity networks. However, we ignore the energy consumed by the processor because we haven't identified operations such as aggregations, compressions or treatments done by processor. In addition, we have also ignored the energy consumed by the memory because the sensor node doesn't save data at its level.

However, we retained the energy consumed by the radio, due to the data transmission, the data reception and the changing of the operating mode (transient energy). In figure 8, we present the transmitter parametric diagram which describes the energy consumed by this component. This diagram includes also the transmitter transient energy. Furthermore, this diagram includes the communication model that describes the state of the transmission channel between the sensor node and the controller. Moreover, the parametric diagrams of the receiver, the processor, the memory and the battery are defined as the transmitter parametric diagram. Moreover, in figure 9, we present the receiver parametric diagram which describes the receiver energy consumption due to receiving data from the controller. Finally, the update of the battery stored energy is illustrated in the parametric diagram shown in figure 10.

*5) The state machine diagram:* The state machine diagram describes the behaviour of the SysML block using a finite state automaton. It shows the possible sequences of states and actions that a block can handle during its lifetime in reaction to discrete events (signals). The UML2 state machine properties are also available with SysML: conditions on events, effects, sustainable activity, transitions, composite states, modularity, concurrent regions, etc.

In our case study, we have described the operations done by the sensor node sub-components in order to calculate the detailed and the global consumed energy. Globally, the controller and the sensor node sub-components have same operations, such as: receiving, transmitting and processing data. For example, the sensor node sends to the controller, continuously, the lane status(number of vehicles on the road). The sensor processing unit receives this data from the additional sensing unit (camera). Then, the processor forwards this data to the transmitter (radio) in order to send its to the controller. Figure 11 illustrates the transmitter state machine diagram. However, once the receiver (radio) receives a message from the controller, it sends it directly to the processor. Figure 12 illustrates the receiver state machine diagram. After that, the processor unit extracts the command signals from the receiving message and sends them to the actuator (traffic lights). In our study, we assumed that for each sending or receiving message, the processor changes its state at least once. This hypothesis allows us to calculate the processing unit transient energy. which is illustrated in figure 13. Finally, in figure 14, we illustrate the global energy consumption by the sensor node which is updated for each sending or receiving message.

### *D. Transformation rules from SysML to Modelica*

In the literature, we found several attempts to define correspondences between the SysML (or UML2) and the Modelica languages. These works are based on the MDA

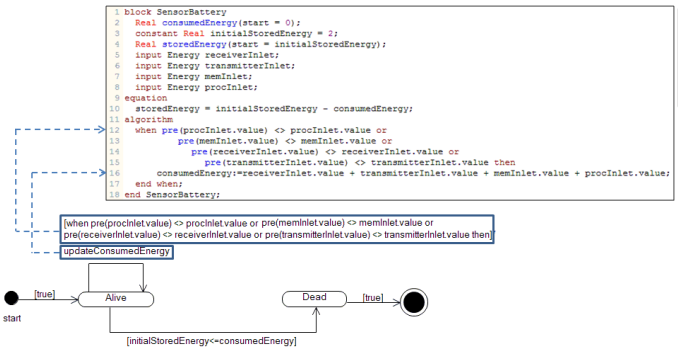


Figure 14. Battery state machine diagram

Table I
TRANSFORMATION RULES FROM SYSML TO MODELICA

| Elements of SysML model | Elements of Modelica model |
|---|---|
| package, block, abstract-block | package, block, partial-class |
| flow-specification, value-type | connector, type |
| flow-property, flow-port | property, connector |
| connector flux(x,y) | equation connect(x,y) |
| constraint property | equation |
| state machine guard | 'when' statement guard |
| operation without a value to return | instruction block of 'when' statement |
| operation with a value to return | function |
| requirement | boolean expression (invariant or safety constraint) |

standard that describes an approach to make model transformations. For example, in [19], [18], [21] the authors proposed model transformation from the SysML model to the Modelica model based on the SysML static diagrams. Otherwise, W. Shamai and al. [20] proposed graphical language (ModelicaML) that can generate Modelica code. In this work, they took into consideration the UML structural and dynamic diagrams (class diagram, composite structure diagram, activity diagram and state machine diagram). Concerning this work, we think that it is better to keep the original UML2 semantics then to make UML2 profiles for Modelica which limite the UML2 language expressiveness.

Our approach is based on the SysML language. It takes into account the structural and behavioural diagrams. The structural diagrams allow us to describe the structure of the target model and the parametric diagrams allow us to consider the mathematical models which represent the behaviour of the real system. Finally, the state machine diagrams are used to describe the behaviour of each component of the system under study. Table I provides a list of correspondences between the elements of the SysML metamodel and the elements of the Modelica metamodel.

### *E. Virtual verification and validation techniques*

The Modelica language permits simulation and verification through tests. In this context, the seminal work is presented by W. Schamai and al. [17] which shows an approach to verify the properties of the Modelica model. This approach relies on the MBSE (Model Based Systems Engineering). This model will be executed and checked

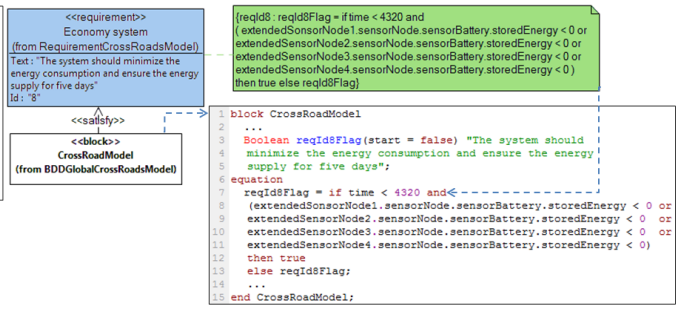


Figure 15. Annotation of the WSN lifetime requirement

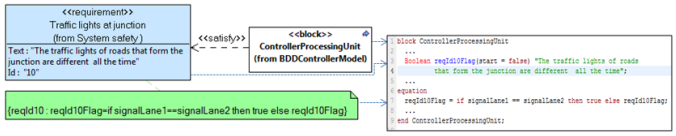


Figure 16. The annotation of safety requirements : the traffic signals in the junction roads are different all the time

against the system's requirements in the early stages of the development cycle. Furthermore, in the MBSE approach, requirements are connected with model elements which allows traceability. The ModelicaML [20] adopts this approach in order to ensure the verification.

In our approach, we rely on the SysML requirement diagrams. The test designer selects the checkable requirements by linking them with the boolean expression (constraint) that represents the invariant or safety constraint of the system. To ensure traceability, the test designer must connect each checkable requirement with one or more blocks that will satisfy this requirement. In this way, we ensure the reusability and the traceability of requirements.

For example, in figure 15 the requirement (Id=08) expresses the longevity that specifies the WSN minimum lifetime(05 days). In other words, the energy in each sensor node will not be exhausted before the desired minimum lifetime. This constraint is the invariant of the "intersection monitoring system" block.

Furthermore, in Figure 16 we detail the safety requirements of the crossroads. For example, the requirement (Id=10) expresses that the traffic lights in both roads that form the junction are different all the time. The constraint that represents the requirement is considered like an invariant of the block "ControllerProcessingUnit".

### *F. Test scenarios (phenomena description)*

The component which represents the phenomena (arrival of cars on the lane) is the bloc 'car'. This bloc contains an operation which allows the generation of integer numbers between 0 and 7. These numbers represent the number of cars which are on the observed lane.

*1) Random tests:* In figure 17, we describe the observed phenomena with state machine diagram. For each five (05) seconds, we generate randomly a new situation for all junction lanes. This information will be transmitted to the camera in order to simulate the sensing of the cars which are on the lane and the estimation of their numbers. This operation mode represents in our case study a random test scenario.

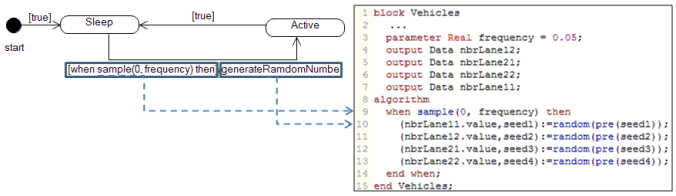


Figure 17. Random test (Arrival of vehicles for each lane in intersection)

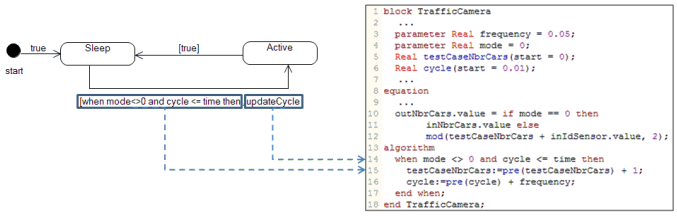


Figure 18. Worst test (Arrival of vehicles for each lane in intersection)

*2) Worst tests:* In addition, we have previewed the examination of the studied system in the worst case situations. For that, we have imagined an operating mode which defines every five seconds one road, from the both roads of junction, empty (from cars) and the other contain at least one car. This operating mode must unroll in altered and cyclic manner. In this way, our system will sent every five seconds two messages in order to change the junction signalisation. In figure 18, we describe the worst case operating mode.

### G. Obtained results

In our adopted modeling approach, we have designed two test cases (random test and worst test). These test scenarios allow us to study the designed model according to the requirement system. Besides, the comparison of the obtained results related to each test scenario allows us to understand the relation between each parameter of the designed system. Moreover, these test scenarios aren't realist because circulation intensity in any junction changes according to time. Consequently, the obtained results don't reflect the real system lifetime.

On the other hand, these test cases provide pertinent elements to study the WSN application. Furthermore, if these test cases don't allow a realistic estimation of a system's lifetime, we can deduce this property from the number of exchanged messages between the controller and sensor nodes placed on each junction lane. However, we must mention that this problem is related to the chosen case study.

*1) Random test results:* To execute the modeled system, we must choose, firstly, the nature of the phenomena. For that, we have decide to execute our system with random data (random tests). In this case, the number of vehicles on each lane will be randomly generated every five seconds.

After these steps, we specified simulation time which must be greater than the WSN desired lifetime in order to analyze all the critical durations of time (minimal WSN lifetime). However, we observed in graph 19 that the requirement "reqId10Flag" which informs that the signal lights in the intersection roads are different is maintained. Otherwise, after "19.22" hours of the simulation, the

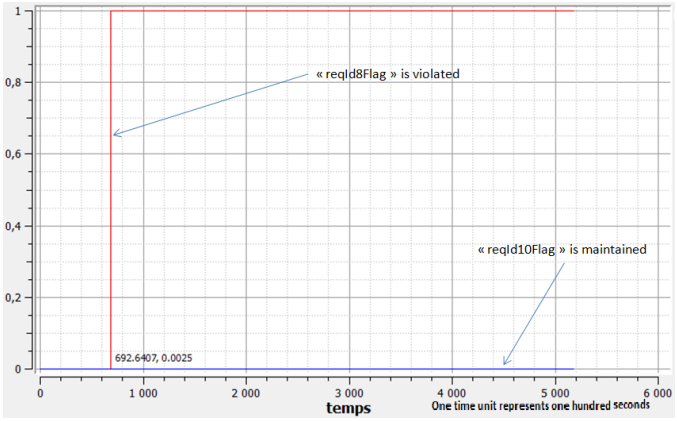


Figure 19. Requirements tracing

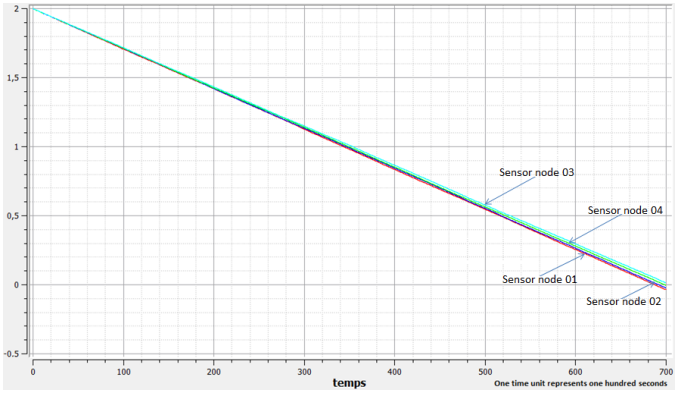


Figure 20. Energy consmuption of sensor nodes

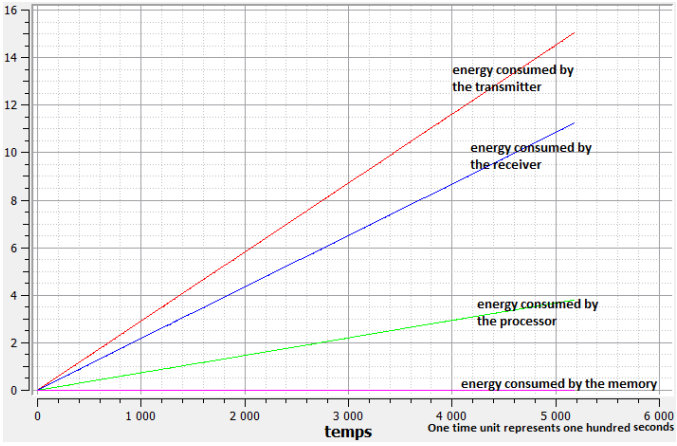


Figure 21. Sensor node energy consumption

requirement that expresses the WSN desired lifetime is violated.

To investigate these results, we can visualize the overall WSN energy consumption in figure 20 and the detailed energy consumption for each sensor node. Figure 21 illustrates the energy consumption of sensor node number one (01). We remark that the transmitter consumes more energy relative to the receiver and this difference is due to amplification. The processor's energy consumption is very small compared to the energy consumed by the transmitter and the receiver.

In addition, we can also see the number of messages exchanged between the controller and sensor nodes. Figure

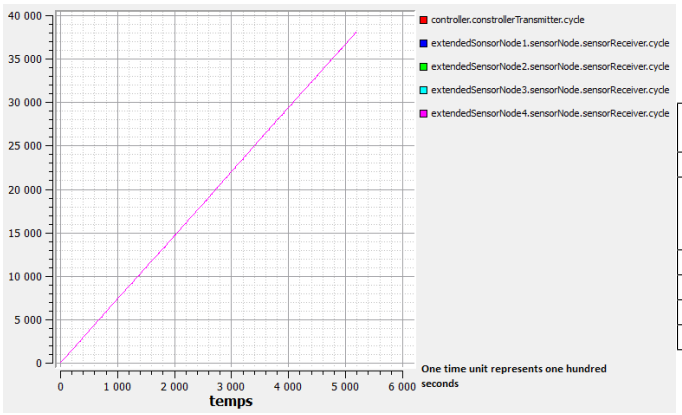


Figure 22. The number of messages sent by the controller and received by the sensor nodes

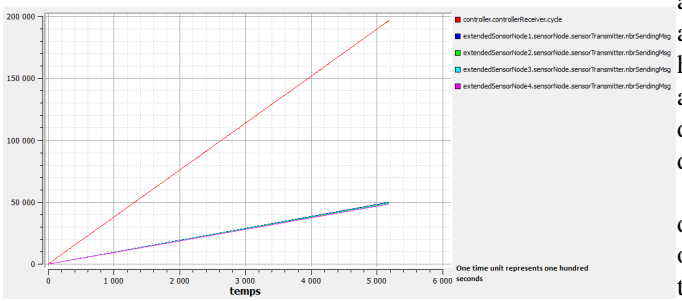


Figure 23. The number of messages sent by the sensor nodes and received by the controller

22 illustrates the messages which are sent by the controller and received by the sensor nodes. Figure 23 shows the messages received by the controller which are sent from the sensor nodes.

*2) Comparison between random and worst test results:* In this section, we establish a comparative table of the obtained results after execution of the studied case with the both test scenarios (random and worst tests). The main objective of this analysis is to deduce the relation between system parameters. According to table II, we remark that the safety property (reqId10Flag) is preserved. The lifetime property (reqId8Flag), which defines the minimum lifetime required, is violated. We observe that the battery energy is consumed more rapidly when applying the worst tests to execute the designed system. We explain this observation by the important number of exchanged messages between controller and sensor nodes, placed on each junction lane, related to the execution of the system with the worst test scenario. Moreover, we deduce that the energy consumption increases according to the exchanged number of messages between controller and sensor nodes. Consequently, the global lifetime of the system decreases regarding to the rise of the exchanged number of messages between the system's parts.

## V. Conclusion and Future Works

This paper proposes an approach to specify and verify the WSN properties using the MDA standard. It combines the benefits of the SysML and the Modelica languages. In the first step, we propose rules to transform the SysML model to the Modelica model taking into account the

Table II
Comparison of obtained results for each tests scenarios (random and worst)

| Properties | Random tests | Worst tests |
|---|---|---|
| Safety property (reqId10Flag) | Preserved | Preserved |
| Lifetime property (reqId8Flag) | Violated (after 19 h. and 22 m.) | Violated (after 11 h. and 11 m.) |
| Controller sending messages | 32000 msg | 175000 msg |
| Sensor node receiving messages | 32000 msg | 175000 msg |
| Controller receiving messages | 160000 msg | 87600 msg |
| Sensor node sending messages | 40000 msg | 21900 msg |

static diagrams, dynamic diagrams and the requirement diagrams of the SysML. After that, we have done virtual verification and requirement tracing. These operations are allowed by the mapping between the SysML requirements and the Modelica properties (constraints). On the other hand, we promoted the reusability of modeled components and we also facilitated the modeling task by reducing the design time, the coding time, the maintenance time and decreasing their relative costs.

We plan in our future work to introduce sequence diagrams and activity diagrams in the action descriptions of the state machine diagrams. The purpose of this step is to provide opportunities for designers who do not master the Modelica programming to work with our Framework. The correctness of the transformation should be validated by unit testing and/or formal proofs.

## Acknowledgment

We would like to acknowledge the support of the Regional Council of Franche-Comté with the SyVad[2] project.

## References

[1] N. Belloir, J.-M. Bruel, N. Hoang, and C. Pham, *Utilisation de SysML pour la modélisation des réseaux de capteurs sans fil*, Conférence sur les Langages et Modèles à Objets (LMO), Montréal, Canada, 03/03/2008-07/03/2008, pages 171–186, http://www.cepadues.com/, mars 2008.

[2] K. Klues, G. Hackmann, O. Chipara, and C. Lu, *A component-based architecture for power-efficient media access control in wireless sensor networks*, Proceedings of the 5th international conference on Embedded networked sensor systems, SenSys '07, pages 59–72, New York, NY, USA, 2007.

[3] C. T. Ee, R. Fonseca, S. Kim, D. Moon, A. Tavakoli, D. Culler, S. Shenker, and I. Stoica, *A modular network layer for sensornets*, Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7, OSDI '06, pages 18–18, Berkeley, CA, USA, 2006.

[4] A. M. I. Khemapech and I. Duncan, *Simulating wireless sensor networks*, Technical report, School of Computer Science, University of St Andrews, 2005.

[5] E. L. E. Cheong and Y. Zhao *Joint modeling and design of wireless networks and sensor node software*, Technical report, Univ. of California, Berkeley, 2006.

[6] M. M. R. Mozumdar, F. Gregoretti, L. Lavagno, L. Vanzago, and S. Olivieri, *A framework for modeling, simulation and automatic code generation of sensor network application*, SECON, pages 515–522. IEEE, 2008.

[2](http://syvad.univ-fcomte.fr)

[7] D. Riley, E. Eyisi, J. Bai, X. Koutsoukos, Y. Xue, and J. Sztipanovits, *Networked control system wind tunnel (ncswt): an evaluation tool for networked multi-agent systems*, Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, pages 9–18, 2011.

[8] F. o. I. Mauro Prevostini, Elena Zamsa ALaRI, *Sysml profile for soc design and systemc transformation*, Technical report, University of Lugano, 2007.

[9] J. A. S. Villa, D. Serna, *Systemc code generation from uml for wireless sensor networks design*, International conference on modeling, simulation and visualization methods, MSV '11, pages 53–60, 2011.

[10] F. Losilla, C. Vicente-Chicote, B. Álvarez, A. Iborra, and P. Sánchez, *Wireless Sensor Network Application Development: An Architecture-Centric MDE Approach*, F. Oquendo, editor, ECSA, volume 4758 of *Lecture Notes in Computer Science*, pages 179–194. Springer, 2007.

[11] P. Boonma and J. Suzuki, *Model-driven performance engineering for wireless sensor networks with feature modeling and event calculus*, Proceedings of the 3rd workshop on Biologically inspired algorithms for distributed systems, BADS '11, pages 17–24, New York, NY, USA, 2011.

[12] L. SAMPER, Modélisations et Analyses de Réseaux de capteurs, PhD thesis, Institut national polytechnique de Grenoble - France Télécom Rech. et Dev. / VERIMAG, 2008.

[13] Ahmed Hammad, Hassan Mountassir, and Samir Chouali, *Combining SysML and Modelica to Verify the Wireless Sensor Networks Energy Consumption*, MODELSWARD 2013, 1st Int. Conf. on Model-Driven Engineering and Software Development, Barcelona, Spain, pages 198–201, February 2013.

[14] M. N. Halgamuge, M. Zukerman, K. Ramamohanarao, and H. L. Vu, *An estimation of sensor energy consumption*, Progress In Electromagnetics Research B (PIERB), (12):259–295, 2009.

[15] H. Lundvall, P. Bunus, and P. Fritzson, *Towards automatic generation of model checkable code from modelica*, SIMS 2004, the 45th Conference on Simulation and Modelling, September 2004.

[16] P. Fritzson and P. Bunus, *Modelica general object-oriented language for continuous and discrete-event system modeling and simulation*, Simulation Symposium, 2002.

[17] W. Schamai, P. Helle, P. Fritzson, and C. J. J. Paredis, *Virtual verification of system designs against system requirements*, Proceedings of the 2010 international conference on Models in software engineering, MODELS'10, pages 75–89, Berlin, Heidelberg, 2011.

[18] C. J. J. Paredis and T. Johnson, *Using omg's sysml to support simulation*, Proceedings of the 40th Conference on Winter Simulation, WSC '08, pages 2350–2352. Winter Simulation Conference, 2008.

[19] R. C. Roland Renier, *De sysml modelica : aide la formalisation de modèles de simulation en conception préliminaire*, 12me Colloque National AIP PRIMECA, 2011.

[20] W. Schamai, P. Fritzson, C. J. Paredis, and A. Pop, *Towards unified system modeling and simulation with modelicaml: Modeling of executable behavior using graphical notations*, Proc. 7th Modelica Conf., Sep. 2009.

[21] P. Vasaiely, *Interactive simulation of sysml models using modelica*, Master's thesis, Hamburg University of Applied Sciences, 2009.