IOP Conf. Series: Materials Science and Engineering **147** (2016) 012050 doi:10.1088/1757-899X/147/1/012050

Issues to be considered on obtaining plant models for formal verification purposes

R Pacheco[1], L Gonzalez[1], M Intriago[2] ,J Machado[1,2] , G Prisacaru[3] and D Olaru[3]

[1]Mechanical Engineering Department, University of Minho, Guimarães, Portugal
[2]MEtRICs Research Center, University of Minho, Guimarães, Portugal
[3]Mechanical Engineering, Mechatronics and Robotics Department, "Gheorghe Asachi" Technical University of Iasi, Iasi, Romania

E-mail: jmachado@dem.uminho.pt

Abstract. The development of dependable software for mechatronic systems can be a very complex and hard task. For facilitating the obtaining of dependable software for industrial controllers, some powerful software tools and analysis techniques can be used. Mainly, when using simulation and formal verification analysis techniques, it is necessary to develop plant models, in order to describe the plant behavior of those systems. However, developing a plant model implies that designer takes his (or her) decisions concerning granularity and level of abstraction of models; approach to consider for modeling (global or modular); and definition of strategies for simulation and formal verification tasks. This paper intends to highlight some aspects that can be considered for taking into account those decisions. For this purpose, it is presented a case study and there are illustrated and discussed very important aspects concerning above exposed issues.

1. Introduction

A mechatronic system is composed, mainly, by three parts: Controller, Plant and Human Machine Interface (HMI) (see figure 1). These parts interact and behave together, and the development of the software to be introduced in the controller, must take into account the behavior of those parts and the interrelation between them.

Several steps can be performed in order to obtain a dependable controller: first, the use of methodologies for obtaining the structure of the controller's specification [1]; second, the use of a formalism to describe, formally, the intended behavior for the controller [2]; third, the use of analysis techniques, in order to guarantee the dependability of the specification [3]; and, fourth, the translation of the specification into a controller program and respective implementation on a physical controller [4]. Concerning use of analysis techniques, plant modeling is one of the bigger issues when performing simulation and formal verification tasks for obtaining dependable software for mechatronic systems [5].

This paper intends to demonstrate how to obtain meaningful plant models for formal verification purposes, taking into account the aspects related with level of abstraction, granularity, modular approach and use of global or partial plant models on the process of formal verification. For achieving this purpose, the paper is organized as follows: section 2 presents the context of analysis techniques and focuses the approach on formal verification by model-checking; section 3 presents a case study, in which are presented the developed specification for the controller and some modules of the plant

model are presented, discussing the above mentioned aspects; section 4 presents some general discussions based on the results obtained at section 3; and, finally, section 5 presents some conclusions.

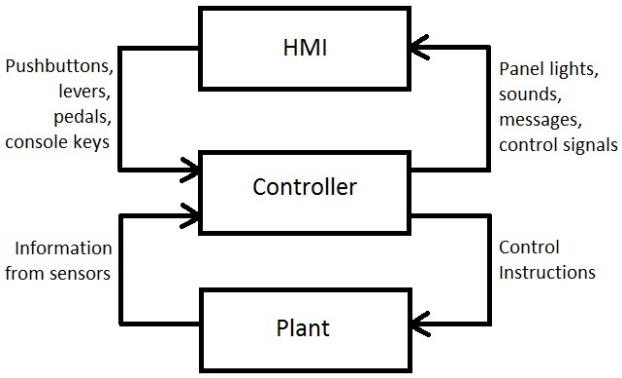

Figure 1. General configuration of a mechatronic system.

2. Context of the work

Among analysis techniques, commonly used for obtaining dependable software for mechatronic systems, simulation and formal verification are used, usually, in a complementary way: simulation is faster and allows fast results obtaining and formal verification is used for achieving more complex results that simple use of simulation doesn't allows [6].

3.1 Simulation

Simulation is used with different objectives: from the simulation of mechanical systems behavior, of process behavior [7], or of integrated manufacturing plants [8] to the simulation of more complex systems, more precisely the simulation of software for the control of automation systems, considering different software tools [9]. Considering these last systems, several approaches are possible. However the final goal is still the same, to avoid major damages, and to be sure, before the implementation of the controller, that the system will comply with the expected behavior. Relevant computerized tools, suitable for integration with traditional design methods, are essential to meet future needs of efficient engineering. A number of approaches can be used to improve this technique, in order to obtain accurate Simulation results.

Simulation cane be performed on four different approaches (see figure 2): Model-in-the-loop (MiL), when only models of the controller and of the pant are simulated; Software in the Loop (SiL), where a model of the controller is simulated interacting with the real plant; Hardware in the Loop (HiL), where a model of the plant is interacting with the real controller; and Laboratory Testing (LT), where both, the real controller and the real plant are interacting.

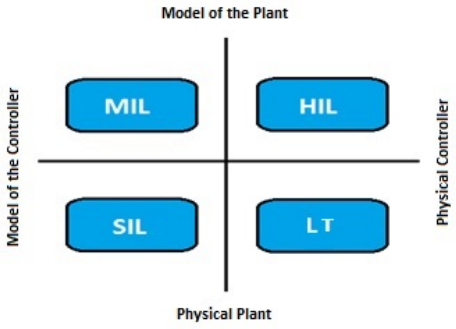

Figure 2. Simulation approaches on the domain of developing reliable software for mechatronic systems.

2.2 Formal Verification

The process of reasoning about formal models of systems has since long been an object of study [10]. For a long time, however, formal proofs tended to stay in the area of envisaged benefits that were never actually completely fulfilled. It would be said that formal proofs could be done, but little was shown about how to actually prove interesting properties of a system. The fact is that formal proof tends to be a delicate, detailed, and time consuming process. As the complexity of models grows, tackling such proofs by hand becomes increasingly harder. This has led to the study of mechanical reasoning techniques as a way to (at least partially) automate the analysis.

Because of the above mentioned reasons, formal verification by model-checking is one of the most used techniques, due the facilitating of designer tasks, mainly because is an automated technique.
When performing formal verification by Model-Checking [11] some approaches can be adopted for modeling the plant, considering different levels of abstraction and, also, some levels of granularity for developing the models [12]. Despite that, designers can consider a global model or, as alternative, a modular approach for obtaining the model of the plant [13].

All the indicated above decisions must be performed taking into account some aspects concerning the goals intended to reach by the analysis of the plant behavior, namely concerning with the behavior properties that are intended to prove, using a formal language [14].

3. Case study

3.1 Description of the system

The object of study of this paper is a workbench with a mechanism that recreates a lift. This lift has been performed at Research Laboratory on Mechatronic Systems of Mechanical Engineering Department of University of Minho, Portugal.

From technological point of view, this systems is characterized by representing a lift for a building of four different floors starting on zero and ending on the third floor, which respective photo is presented in figure 3. Authors want to highlight that this physical system is a physical model representing a real lift, but the used controller (a Siemens programmable logic controller SIMATIC S7-1200) is the same that can be used in a real application with a real lift.

This case study is representative of some bad consequences that can result if controller's software is not developed in a correct way. For this, for obtaining the software for the controller of this system, a systematic approach is used: to create a formal specification for the system, using Sequential Function Chart (SFC) formalism [15]; to model the plant behavior, using Timed Automata formalism [16]; to use simulation (MiL) and to use formal verification by model-checking.

Timed automata are used for modeling the plant, due to two main reasons: this is a non-deterministic formalism, that is suitable for modeling the plant; and allows considering the time, on the modeling tasks. Despite those reasons, it is the input formalism for UPPAAL model-checker [17], the software tool that has been chosen for performing both analysis techniques: simulation and formal verification.

The physical part is composed by a structure of aluminum (see figure 3) and a cabin which is the one that is going to travel by the floors. It's also composed by a base where is placed the Programmable Logic Controller (PLC) and the Human Machine Interface (HMI) used to interact with the workbench. The movement is possible thanks to a motor and a spindle. When the cabin is detected by the sensors available on the structure a light will turn on and indicate in which floor the cabin is. Thanks to four call push buttons is possible to call the lift from any floor at any time.

The logical part or the control system is composed by the microcontroller SIMATIC S7-1200 which can be programmed over the software TIA Portal V11 and by an interface in which case is the SIMATIC Panel KTP600 Basic Color PN.

Figure 3. General configuration of a mechatronic system.

3.2 Model of the controller

The model of the controller was specified using SFC,from the IEC 60848 standard, also known as "*Grafcet*".

For specifying the controller behavior, some approaches can be adopted: a simpler one where the lift moves to a defined floor without taking into account calls from other floors when it is moving; or the opposite: to consider all calls, on different floors, even when it is moving. Between both extreme specifications, there are several configurations that can be adopted. The specification, considering this last approach (more complex) can be hard to obtain, because two main reasons:it is increased the number of variables to consider and SFC is not, exactly, the perfect formalism for modeling the specification of the controller of a lift.

Because development of controller's specification is not the main goal of this paper, only one SFC (concerning specification of the movement of the lift) is presented in figure 4. There are essentially two variables that must be controlled in order to make the lift system work: the first one is the one that identifies where the lift is and what is it doing (Pa); the second one is the one that specifies where the lift should be (Pch). For the considered complexity for the controller, three SFCs have been developed: one describes behavior of "Pa" variable; another one describes behavior of "Pch" variable; and another one describes the specification of the lift's movement.

The inputs and outputs of the controller's model are presented in table 1 and table 2.

Table 1. Inputs of the controller model.

Variable	Meaning
b0	Button to call the lift to the floor 0
b1	Button to call the lift to the floor 1
b2	Button to call the lift to the floor 2
b3	Button to call the lift to the floor 3
fc_0	Floor 0 - sensor
fc_1	Floor 1 - sensor
fc_2	Floor 2 - sensor
fc_3	Floor 3 - sensor

Table 2. Outputs of the controller model.

Variable	Meaning
mt_sb	Lift goes up
mt_dc	Lift goes down
led0	Floor 0 - indicator light
led1	Floor 1 - indicator light
led2	Floor 2 - indicator light
led3	Floor 3 - indicator light
buzzer	Buzzer makes sound
door open	Open door

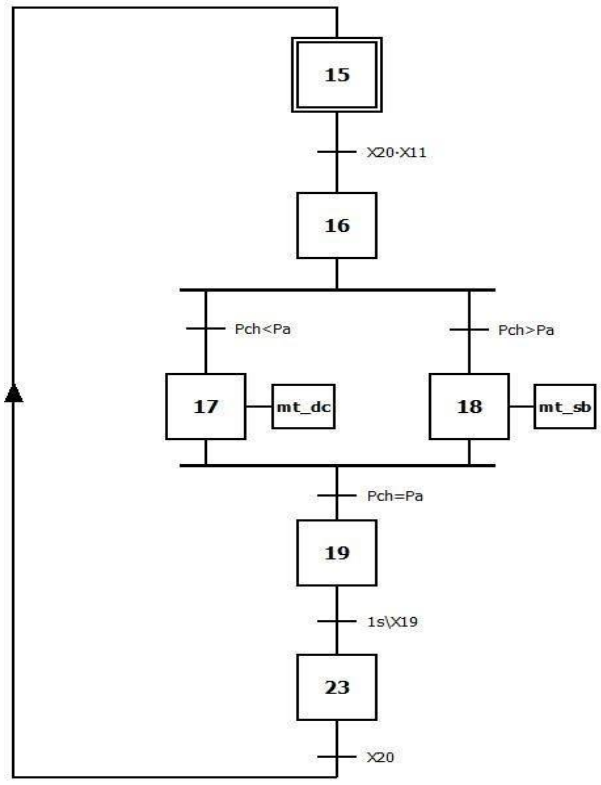

Figure 4. SFC specification of the lift's movement.

3.3 Model of the plant

The plant was modeled taking into account some decisions. Those decisions will be discussed, in detail, on next topics.

In this section only three modules, of the entire model of the plant, are presented. Those models represent some aspects that are proposed to be discussed in this paper. For the entire plant model all the physical components were modeled: sensors, buttons, leds, motor, door and lift.

3.3.1 Level of abstraction and granularity

The level of abstraction and granularity of the plant models is the first issue that designer must handle. Usually, the adopted solution must take into account the kind of behavior properties that it is intended to prove using analysis techniques. For instance, on the model presented in figure 5 (model of the door), designer could decide for a model with two locations ("door closed" and "door open"); a model with three locations ("door closed", "door in the intermediate position", between open and closed, and "door open"); and a model with four locations ("door closed", "door opening", "door open" and "door closing"), among other possible solutions. In fact, decision must take into account, for instance, if there is a behavior property – important for dependability of the system – in which it is needed to prove something if door is opening or closing. If there is, the model must have the configuration of the model of figure 5; if not, the model must me as simples as possible, taking into account the behaviors that are intended to prove by simulation and/or formal verification.

3.3.2 Instantiable models

These kinds of models are modular and correspond to physical parts that can appear on this system or on other systems. In all systems that they appear, they are the same component, so we can reuse those models instantiating them with different variables. This is a very important issue: the models are created once, it is possible to build a library of modules, and then they are reused when necessary. Model of the door are representative of this group of models.

- Model of the door

The model of the door (figure 5) considers four locations for describing the behavior of the door: P1 – door closed; P2 – door opening; P3 – door closed; and P4 – door closing. The input of the model is the actuation of monostable actuator for opening the door. If the actuator doesn't receives electrical signal, the door, closes.

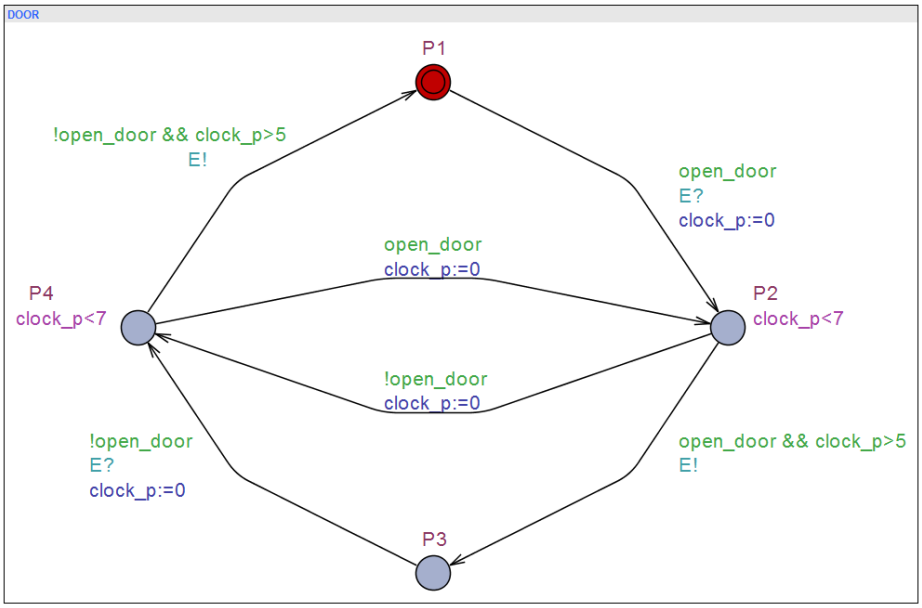

Figure 5. Model of the door.

- Model of the motor

For the model of the motor (figure 6), three locations have been considered: stopped (P5), Moving up (P7) and Moving down (P6). In this model two Boolean variables have been created (MS and MD). Those variables are responsible for connecting this model with the model of the lift. Variables "mt_dc" and "mt_sb" are described on table 2.

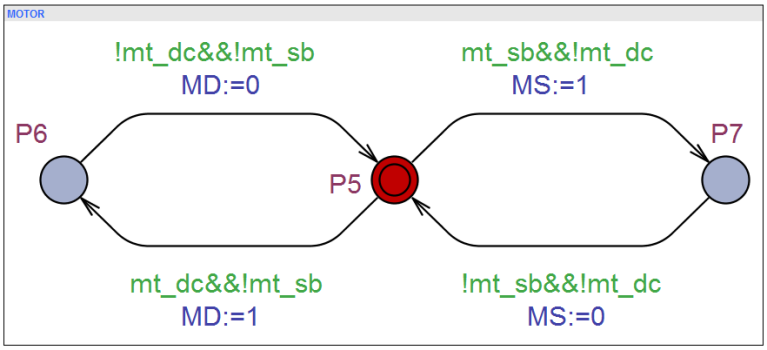

Figure 6. Model of the motor.

3.3.3 Specific models

Those models are models that are specific for each application or system. Usually they are not instantiable, for this reason. The model of the lift is representative of this group. It is a specific model of the mechatronic system.

- Model of the lift

The model of the lift (figure 4) depends of the movement of the motor (behavior described in figure 6) and the respective inputas are MS and MD, variables defined on the model of the motor. Lift moves up when MS=1 and moves down when MD=1. If MS=0 and MD=0, the motor stops and, also, the lift stops.

4. Simulation and Formal Verification strategies

Concerning simulation, a very complete plant model is the best approach for obtaining satisfactory results. This way, it is possible to detect mistakes and faults on the behavior of the system.

Concerning formal verification, one problem with model-checking is related to the state explosion problem. The model may become too big for verification to be feasible with reasonable resources. In this paper we report on results of work on model-based verification non resorting to partial models of the plant. This enables the use of smaller models, thus making it possible to verify larger systems.
Considering the approach proposed in [5], formal verification tasks can be performed with the assumption of a closed loop behavior of the controller model and the plant model. Also, in the same work, it is proposed that a possible solution for obtaining the plant model for this system is considering a set of plant modules, in order to obtain a modular solution for the entire system plant model.

In [5] a set of behavior properties for the exposed system is considered, to be proven using verification by model-checking. This set of properties is composed by safety properties and liveness properties. The same work proposes a systematic approach to prove the set of properties using, or not, the plant model of the system, depending on the specific type of property under consideration. It was observed that some safety properties were not proved without a plant model, but were proved when the entire plant model was used.

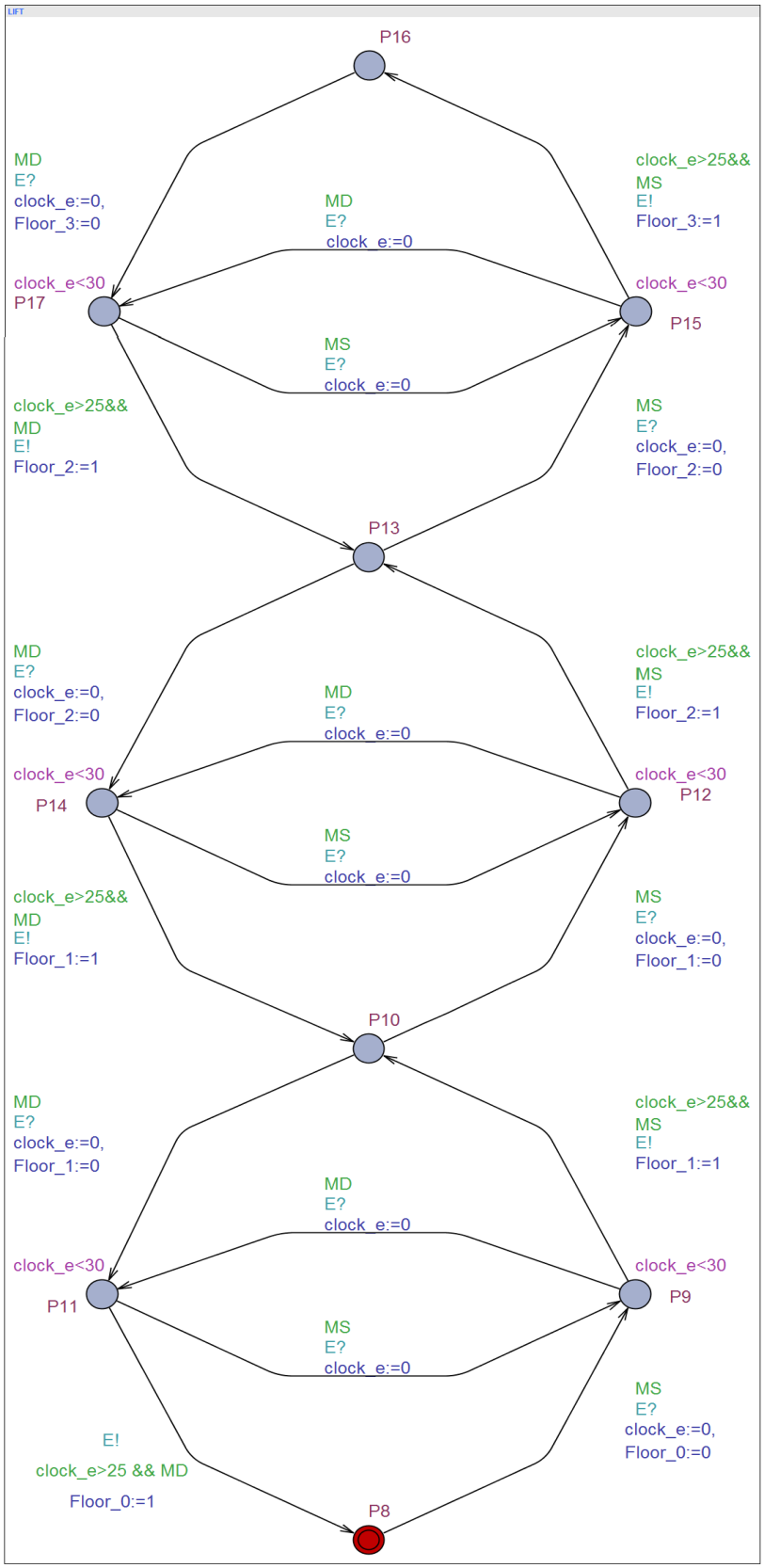

Figure 7. Model of the lift.

5. Conclusions and future work

The modeling of plant behavior of mechatronic systems for obtaining dependable software for their controllers is a very hard task. In order to do it, designers must be highly skilled because quality of simulation results and of formal verification results is highly depending of the "quality" of the developed models. In this paper some issues related with this subject have been discussed and some solutions have been presented.

Concerning future work, authors of this paper intend to develop methodologies and tools for obtaining those plant models in a systematic way, to be used by engineers that, usually, are not expert designers using those complex formalisms and tools for obtaining meaningful plant models for simulation and formal verification purposes.

6. References

[1] Machado J M and Seabra E 2009 A systematized approach to obtain dependable controllers specifications *Proc. of 20th International Congress of Mechanical*

[2] David R 1995 Grafcet: a powerful tool for specification of logic controllers *IEEE Transactions on Ctrl. Sys. Tech.***3** pp 253-68

[3] Johnson T L 2007 Improving automation software dependability: A role for formal methods? *Ctrl. Eng. Pra.* **15(11)** pp 1403-15

[4] Machado J, Denis B, Lesage J J, Faure J M and Silva J F D 2006 Logic controllers dependability verification using a plant model *Proc. of 3rd IFAC Workshop on Discrete-Event System Design* pp 37-42

[5] Machado J, Seabra E, Campos J C, Soares F and Leão C P 2011 Safe controllers design for industrial automation systems *Comput. Ind. Eng.* **60(4)** pp 635–53

[6] Kunz G, Machado J and Perondi E 2015 Using Timed Automata for Modeling, Simulating and Verifying Networked Systems Controller's Specifications *Neural Comput. and App.* pp 1-11

[7] Huda A and Chung C 2002 Simulation modelling and analysis issues for high-speed combined continuous and discrete food industry manufacturing processes *Computers & Industrial Engineering* **43(3)** pp 473–83

[8] Eben-Chaimea M, Pliskina N and Sosna D 2004 An integrated architecture for simulation *Computers & Industrial Engineering* **46(1)** pp 159-70

[9] Hlupic V 1999 Simulation software: user'srequirements *Computers & Industrial Engineering* **37(1-2)** pp 185-8

[10] Jones CB 2003 The early search for tractable ways of reasoning about programs *Annals of the History of Computing IEEE* **25(2)** pp 26-49

[11] Moon I 1994 Modeling programmable logic controllers for logic verification *IEEE Control Systems* **14(2)** pp 53-9

[12] Machado J 2006 *Influence de la prise en compte d'un modèle de processus en verification formelle des Systèmes à Evénements Discrets (PhD Thesis)* (France: École Normale Supérieure de Cachan)

[13] Gouyon D 2001 *Application de techniques de sysnthèse de la commande en ingéniéried'automatisation (MSc Thesis)* (Nancy: University Henri Poincaré)

[14] Clarke E M, Emerson E A and Sistla A P 1986 Automatic verification of finite state concurrent systems using temporal logic specifications *ACM Transactions on Programming Languages and Systems* **8(2)** pp 244-63

[15] EN 2002 (2002) European Standard 60848: *SFC specification language for sequential function charts*

[16] Alur R and Dill D L 1990 Automata for modeling real-time systems *Proc. of 17th Int. Coll. Automata, Languages, and Programming*

[17] Behrmann G, David A and Larsen K G 2004 A tutorial on UPPAAL *Formal methods for the design of real-time systems* pp 200-236 (Springer Berlin Heidelberg)