Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-2 (2018) 248–253

Using Unified Enhanced Time Petri Net Models for Cyber-Physical System Development

T. S. Letia * **A. O. Kilyen** **

* *Technical University of Cluj-Napoca, Romania (e-mail: Tiberiu.Letia@aut.utcluj.ro).*
** *Technical University of Cluj-Napoca, Romania (e-mail: ors.kilyen@accenture.com)*

Abstract: The Cyber-Physical Systems (CPSs) are designed as a network of interacting entities with physical inputs and outputs. They usually require reactions involving artificial intelligence and fulfilling some temporal constraints. The newly proposed models, named here Unified Enhanced Time Petri Nets (UETPNs), combine the well-known features of standard Petri Nets (PNs) with timed PNs, fuzzy logic and rule based systems. They are used to conceive a method for developing CPSs that uses a set of communicating components, each of them integrating a UETPN model. The CPSs verification concerns the logical and temporal verification, as well as the performance evaluations. The method is used to conceive a distributed control system for an urban vehicle traffic.

Keywords: Petri nets, cyber-physical system, fuzzy logic, genetic algorithm, urban vehicle traffic.

1. INTRODUCTION

Cyber-Physical Systems (CPSs) are given as examples of ubiquitous advanced complex technological systems linked to physical plants and endowed with computing environment intensively running software often involving time critical applications. Features often associated to CPSs are: adaptability, autonomy, efficiency, functionality, reliability, resilience, safety, security and usability. The main goal of the current research was to conceive a new kind of models suitable for solving problems raising up in CPS applications. The article shows how these models can be used for developing a complex CPS. The considered applications involve different components capable of cooperations and reactions relative to asynchronous events and asynchronously received messages, or continuous changing of some measured variables.

The main contributions of the current research consist of a kind of models that can be used for CPSs in the context of:

- the lack of information because some signals or information are not generated, are not transmitted or do not arrive in time,
- some information is not consistent with others,
- the regular behavior is interrupted by urgent tasks,
- the entrance and exit into system of different entities occurs asynchronously without notifying their collaborators,
- the applications work with predictions having assigned different degrees of confidence,
- the information is presented as logical or numerical values and
- the events occur asynchronously and have different importance levels.

CPSs include software and hardware components that have dynamic behavior and interact with each other. The development of such applications needs models that are capable to describe both kinds of components and their collaboration. To make all of the above mentioned goals possible, the models have to describe simultaneously both the program structures and their dynamic behavior. Some non-functional parameters and functions could be necessary to verify if the applications fulfill the temporal requirements and the expected performances. The distributed architectures are achieved using some components that implement the proposed kind of Petri net models.

2. RELATED WORKS

The control and verification of CPS behaviors involve reactive systems. Some methods of approaching the reactive systems are based on: specification languages, fix point calculi, finite automaton, temporal logic and predicate logic (Schneider, 2004). Often the CPSs can be seen as network of event systems (Ramesh et al., 2014). The use of reactive event based systems in CPSs is performed by integrating the dynamics of physical processes with that of software and communication. A survey that presents the main characteristics, the main domains where they are applied and the main topics is given in (Shi et al., 2011).

The Petri Nets (PNs) capability to simultaneously describe the structure and the dynamic behavior of reactive applications has been known for a very long time. They are used for specification, design, verification and performance analysis (Murata, 1989). Ghezzi et al. (1991) introduced

10.1016/j.ifacol.2018.03.043

the Environment/Relationship Nets defining the marking as an assignment of multisets of environments to places. The tokens are functions from a set of identifiers to a set of values. They further developed a strong formalism capable to deal with timing issues and functional requirements. The PNs were endowed with time features that open a new research direction (Juan et al., 2001). Another direction was introduced linking the PNs with Fuzzy Logic (FL) and later their involvement in rule based systems (Meseguer, 1990; Yang et al., 2013; Zhou et al., 2016; Munakata et al., 1994; Lee, 1990; Virtanen, 1995; Liu et al., 2017).

The current approach of CPSs is based on components endowed with UETPN models. These models have the advantages:

- They have a single kind of places and transitions even if they describe discrete event or discrete time behavior, continuous and logical variables.
- They are simple to be implemented and included in components and thus they can be used for complex and large applications.
- They do not need conditional expressions as thresholds for enabling the transitions.
- They include all the main features available into different kinds of PNs (like inhibitor arcs, reset arcs, variable delays etc.).
- They are easy to be synthesized due to their simple and unified structure. A simple genetic algorithm (GA) can be used for UETPN model training.

3. UETPN MODELS

The UETPN models endow the previous Fuzzy Logic Enhanced Time Petri Net (FLETPN) models with the capability to describe arithmetic operations with real numbers besides the logical operations (Letia et al., 2016; Letia et al., 2017).

The FLETPN models use Fuzzy Sets (FSs) and mapping tables (MTs) storing synthetically the inference rule set involved by each transition. An example of FS is:

$$FS = \{X_{-2}, X_{-1}, X_0, X_1, X_2\} \quad (1)$$

with X_k (k=-2, -1, 0, 1 or 2) identifying a subinterval of a real variable domain or a fuzzy value of a fuzzy logic variable. An MT is exemplified in Table 1. The first role of MT expresses the condition to execute the transition under the current input values (or tokens). The second role provides the tokens that are created and moved through the net.

An example of UETPN is given in Fig. 1. To achieve the UETPN models, the FLETPN models have been modified with:

- Each place p_k has assigned a real number variable x_k.
- Each variable x_k has a specified scale factor s_k that determines the variable bounds in the continuous interval $[-s_k, s_k]$.
- A variable x_k can be involved in an arithmetic operation and/or a fuzzy logic operation.
- The previous FS is extended to $EFS = FS \cup \phi$, where ϕ is the empty set.
- Each transition t_i has assigned a mapping map_i defining the operations between the input variables of the input place set ${}^o t_i$ and the output variables of the output place set t_i^o:

$$map_i : ([-s_1, s_1] \cup \phi) \times ([-s_2, s_2] \cup \phi) \rightarrow ([-s_3, s_3] \cup \phi) \times ([-s_4, s_4] \cup \phi). \quad (2)$$

where ϕ is the empty set. The mapping is presented for a transition with two inputs and two outputs, but generally, the number of inputs and outputs can be different.

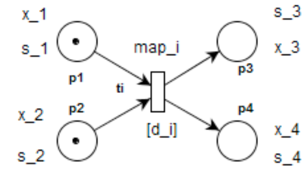

Fig. 1. Example of a UETPN.

Table 1. **Example of inference rules with empty set**

x_1/x_2	X_{-2}	X_{-1}	X_0	X_1	X_2
X_{-2}	X_2, X_2	X_2, X_2	X_2, X_2	X_2, X_1	X_2, X_0
X_{-1}	X_2, X_2	X_2, X_2	X_2, X_1	X_2, X_0	X_2, X_{-1}
X_0	X_2, X_2	X_2, X_1	X_2, X_0	X_2, X_{-1}	X_2, X_{-2}
X_1	X_2, X_1	X_2, X_0	X_2, X_{-1}	X_2, X_{-2}	X_2, X_{-2}
X_2	X_2, X_0	X_2, X_{-1}	X_2, X_{-2}	X_2, X_{-2}	X_2, X_{-2}
ϕ	ϕ, ϕ	ϕ, ϕ	X_2, ϕ	ϕ, X_0	ϕ, ϕ

For a logical mapping the notation $x_k = \phi$ means "x_k is not known" at the current moment of time and it is used according to the extended fuzzy logic rule set. The variable x_k remains neutral in the decision that has to be made executing the transition.

When the same value $x_k = \phi$ is used by an arithmetic mapping, it has to remain neutral too, so it is used as $x_k = 0$ to add and subtract operations and $x_k = 1$ for multiplication or division operations. Each mapping is related to a logical or arithmetical operation from the set $op = \{\wedge, +, -, \star, /\}$ with the notations: '$\wedge$' for logical AND, '+' to add, '-' to subtract, '$\star$' for multiplication and '/' for division. The mapping functions map_i for all arithmetic operations (denoted here by '$\circ$') are built of the following form:

$$x_k = map_i(x_1, x_2) = (x_1 \circ x_2) \star FL_{MT}(x_1, x_2); k = 3, 4 \quad (3)$$

If the result of $map_i(x_1, x_2)$ exceeds the specified domain $[-s_k, s_k]$, the variable x_k is bounded and an exception is signaled to simulator.

The fuzzy logic transformation is used even if the mapping involves an arithmetic operation for the following reasons: it is used to decide if a transition is enabled, and the designer is not very confident about the values of the input variables or about the specified operation in some of the sub-domains of the variables, the output is "bended" accordingly. For the case when the confidence is maximum, MT has to contain only the values X_2 and in this case the bending has no effect on the classical result. For complete definition of UETPN see Letia et al. (2017).

The UETPN models are capable to describe PN with *inhibitor arcs* adding a new line or column in MT for a