12th IFAC/IEEE Workshop on Discrete Event Systems
Cachan, France. May 14-16, 2014

Unified Model for Synthesis and Optimization of Discrete Event and Hybrid Systems [⋆]

Bengt Lennartson [*] **Oskar Wigström** [*] **Martin Fabian** [*]
Francesco Basile [**]

[*] *Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Göteborg, Sweden (e-mail: bengt.lennartson@chalmers.se).*
[**] *Dip. Ingegneria dell'Informazione, Ingegneria elettrica e Matematica applicata (DIEM), Università di Salerno, Italy, (e-mail: fbasile@unisa.it).*

Abstract: A recently proposed generic discrete event model is further developed and exemplified in this paper. Since every transition is expressed as a predicate on the current and next values of a set of variables, the model is called Predicate Transition Model (PTM). It is briefly illustrated how a number of well known discrete-event models, including automata and Petri nets extended with shared variables, can be formulated and synthesized in the PTM framework. More specifically modular Petri nets with shared variables (PNSVs) are shown to be significantly more readable compared to ordinary Petri nets. PTMs are also naturally extended to hybrid systems, and finally it is shown how easy and efficiently PNSVs can be optimized concerning performance based on Constraint Programming. To summarize, the proposed modeling framework unifies and simplifies both synthesis, optimization and implementation of discrete event systems.

Keywords: discrete-event systems, automata, Petri net, supervisory control, synthesis, hybrid systems, optimization

1. INTRODUCTION

The lack of a unified model representation for discrete signal and discrete event systems (DESs) is well known, see Cassandras and Lafortune (2008). For continuous systems differential equations and transfer functions serve this purpose. In a recent paper a unified model for DESs is proposed, called State-Vector Transition model, Lennartson et al. (2014). Inspired by classical continuous-time state space models, the state of a system is represented by a number of state variables x_j, where the domain of definition of the individual variables can be symbolic states as in automata, or integer values as in Petri nets. All variables together constitute a state-vector x, and a transition from one value of x to an updated next value $\acute{x}$ is enabled when a related predicate $\mathcal{C}(x, \acute{x})$ is satisfied. To emphasize the predicate formulation of the transition model, it is in this paper called Predicate Transition Model (PTM).

This type of model was introduced by Manna and Pnueli (1991) as a generic model for transition systems, and the predicate $\mathcal{C}(x, \acute{x})$ is a natural formulation in model checking , see Clarke et al. (2000). In the supervisory control community a slightly different formulation has been used based on predicate transformers, cf. Kumar et al. (1993). The transition predicate $\mathcal{C}(x, \acute{x})$ is shown to be very useful both from a modeling and a computational point of view.

Communication between different discrete event models is often obtained by shared events and full synchronous composition Hoare (1978), as in automata and Petri nets. For automata extended with variables, one version called Extended Finite Automata (EFAs), Sköldstam et al. (2007), communication and synchronization can also be determined by *shared variables* that are updated in more than one EFA. The same feature is used for PTMs, where it is assumed that any variable in the state-vector x can be assigned to new values in more than one model.

In Lennartson et al. (2014) it is shown that the suggested PTM includes automata, EFAs, Petri nets (PNs) and colored Petri nets (CPNs) and their synchronous composition as special cases. Since EFAs are automata extended with variables, PNs are also extended with additional shared variables, including guards and actions. This model, called *Petri net with shared variables* (PNSV), is an interesting complement to the formal PTM, that does not include any specific graphical model structure. In many situations a set of local PNSVs can give a clear and understandable graphical model. For the suggested PTM a synthesis procedure is also developed where supervisor guards are generated. Based on a set of local plant and specification models, synchronized in the PTM framework, it is shown how a controllable, nonblocking and maximally permissive supervisor can be computed.

The contribution of this paper is to briefly summarize the PTM in Lennartson et al. (2014), and clarify some important aspects related to controllability. The benefit of modular PNSV is also further exemplified and motivated in this paper. Already for a system with three straight sequences of token flows, it is illustrated how an ordinary PN becomes hard to read, while the PNSV model gives a clear view of the modeled behavior. It is also shown how PTMs are naturally extended to include continuous behavior, which results in modular Hybrid Predicate Transition Models (HPTMs). Finally, it is illustrated

[⋆] This work was carried out within the Wingquist Laboratory VINN Excellence Centre within the Area of Advance - Production at Chalmers, and supported by VINNOVA and the Swedish Science Foundation. The support is gratefully acknowledged.

how especially the PNSV model has a structure that makes it extremely simple to convert to a Constraint Programming (CP) model, which is used for minimization of the make span for the PNSV model discussed above. Note that CP, see Baptiste et al. (2001) is most often significantly more efficient for this type of performance optimization compared to more classical Mixed Integer Linear Programming (MILP) solvers, Manne (1960). To summarize: the proposed PTM unifies the classical DES model structures, also including discrete signals in a natural way, it gives a modular framework both for discrete event and hybrid systems, and the model is shown to be useful both for supervisor synthesis and performance optimization.

2. PREDICATE TRANSITION MODEL

A formal definition of PTMs is given in this section, see further details in Lennartson et al. (2014) where the model is called state-vector transition model. Since PTMs include a number of predicates, note that a subset W of a set X also can be defined by the *predicate* mapping $\mathcal{W} : X \to \mathbb{B}$ as $\mathcal{W}(x) = 1$ iff $x \in W$ and $\mathcal{W}(x) = 0$ iff $x \notin W$.

2.1 Definition of the Predicate Transition Model

Consider a universal ordered set (tuple) of discrete variables $(x_1, \ldots, x_n)$, where the domain of definition for each variable x_j is X_j. A subset of these variables are included in a tuple x, for which a transition model is now defined.

Definition 1. (Predicate Transition Model). A predicate transition model G is a 6-tuple

$$G = \langle \Omega_x, X, \Sigma, T, \mathcal{X}_i, \mathcal{X}_m \rangle \tag{1}$$

where:

(i) $\Omega_x = \{j_1, \ldots, j_n\}$ is the index set for the tuple $x = (x_{j_1}, \ldots, x_{j_n})$.
(ii) $X = X_{j_1} \times \cdots \times X_{j_n}$ is the domain of definition for x.
(iii) Σ is a finite set of events.
(iv) T is a finite set of transitions. Each transition is a tuple $(\sigma, \mathcal{C})$, where $\sigma \in \Sigma$ and $\mathcal{C} : X \times X \to \mathbb{B}$ is a predicate on the current value x and the next value $\acute{x}$.
(v) $\mathcal{X}_i : X \to \mathbb{B}$ is a predicate, defining possible initial values of x.
(vi) $\mathcal{X}_m : X \to \mathbb{B}$ is a predicate, defining desired marked values of x.

□

The reason to introduce the index set Ω_x is that variables will later be arbitrarily shared between different local models. Generally, the domain of the variables may be infinite, as for unbounded PNs, but for computational reasons a finite domain is normally assumed. The predicates are generated by boolean expressions, including conjunction $\wedge$, disjunction $\vee$, and negation $\neg$, while relations between variable values involve the operators $=, \neq, <, >, \leq$, and $\geq$.

A transition $(\sigma, \mathcal{C})$ is *enabled* when the predicate $\mathcal{C}(x, \acute{x})$ is true. When the enabled transition is executed the event σ occurs. For a model with tuple $x = (x_1, x_2)$, a transition predicate $\mathcal{C} \equiv x_1 < 2 \wedge x_2 = 0 \wedge \acute{x}_1 = 2$ means that the transition is enabled when $x_1 < 2 \wedge x_2 = 0$, and the next value of x_1 is $x_1 = 2$. Since there is no condition on the next value for x_2, a natural assumption is that it keeps its current value, i.e. $\acute{x}_2 = x_2 = 0$, cf. Sköldstam et al. (2007); Lennartson et al. (2014).

Also note the condition on the next value $\acute{x} \in X$. When for instance the domain $X = \{0, 1, 2\}$, the conditions $\acute{x} = x + 1$ and $\acute{x} = x - 1$ implicitly include the additional guards $x < 2$ and $x > 0$, respectively. These conditions do not need to be explicitly introduced, since they are achieved by the domain of definition for $\acute{x}$.

2.2 State Transition Model

For analysis and synthesis purposes, the predicate transition model in (1) is now formulated as an explicit state transition model. This model defines the semantics of the PTM. To formally define the fact that no condition on $\acute{x}_j$ in $\mathcal{C}(x, \acute{x})$ implies that x_j keeps its current value, consider the index set

$$\Omega_{\mathcal{C}} = \{j \mid \text{condition on } \acute{x}_j \text{ in } \mathcal{C}(x, \acute{x})\}$$

This means that any expression involving $\acute{x}_j$, such as $\acute{x}_j = x_j + 2$ or $\acute{x}_j < 3$, implies that $j \in \Omega_{\mathcal{C}}$. No condition on $\acute{x}_j$ in $\mathcal{C}(x, \acute{x})$ yields $j \in \Omega_x \setminus \Omega_{\mathcal{C}}$, and the variable x_j will be assumed to keep its current value, i.e. $\acute{x}_j = x_j$. Introducing the keep-current-value predicate

$$\mathcal{C}_{cv}(x, \acute{x}) \equiv \bigwedge_{j \in \Omega_x \setminus \Omega_{\mathcal{C}}} \acute{x}_j = x_j, \tag{2}$$

the *complete transition predicate* for transition $(\sigma, \mathcal{C})$ becomes

$$\Phi(x, \acute{x}) \equiv \mathcal{C}(x, \acute{x}) \wedge \mathcal{C}_{cv}(x, \acute{x}) \tag{3}$$

Consider e.g. the predicate $\mathcal{C} \equiv x_1 = 1 \wedge \acute{x}_2 = 3$ and the index set $\Omega_x = \{1, 2\}$. Then $\Omega_{\mathcal{C}} = \{2\}$, and the complete transition predicate $\Phi \equiv x_1 = 1 \wedge \acute{x}_2 = 3 \wedge \acute{x}_1 = x_1$.

The defined complete transition predicate is the basis for the definition of the explicit state transition model. Indeed, it can be considered as an *evaluated* PTM, where $\Phi(x, \acute{x})$ is determined for all possible combinations of values of the variables.

Definition 2. (State Transition Model). A state transition model (STM) of a PTM $G = \langle \Omega_x, X, \Sigma, T, \mathcal{X}_i, \mathcal{X}_m \rangle$ is a 5-tuple

$$\widehat{G} = \langle X, \Sigma, \to, X_i, X_m \rangle \tag{4}$$

where X and Σ are defined in Definition 1, and using the complete transition predicate $\Phi(x, \acute{x})$ in (3)

- $\to = \{(x, \sigma, \acute{x}) \in X \times \Sigma \times X \mid \exists (\sigma, \mathcal{C}) \in T : \Phi(x, \acute{x})\}$;
- $X_i = \{x \in X \mid \mathcal{X}_i(x)\}$;
- $X_m = \{x \in X \mid \mathcal{X}_m(x)\}$.

□

The STM $\widehat{G}$ is an automaton, where the state transition relation is written $x \xrightarrow{\sigma} \acute{x}$, which is recursively extended to strings in Σ^* by letting $x \xrightarrow{\varepsilon} x$ for all $x \in X$, and $x \xrightarrow{s\sigma} z$ if $x \xrightarrow{s} y$ and $y \xrightarrow{\sigma} z$ for some $y \in X$. A path from an initial state in $\widehat{G}$ to a state x is written $\widehat{G} \xrightarrow{s} x$, while a path from a state x to a marked state in X_m is denoted $x \xrightarrow{s} X_m$. Furthermore, $x \xrightarrow{s}$ means that $x \xrightarrow{s} y$ for some $y \in X$, and $x \overset{s}{\nrightarrow} y$ implies that no string $s \in \Sigma^*$ exists such that $x \xrightarrow{s} y$, while $x \overset{s}{\nrightarrow}$ means that $x \overset{s}{\nrightarrow} y$ for all $y \in X$.

The PTM G in (1), and its corresponding STM $\widehat{G}$ in (4), naturally include nondeterministic behavior, but a *deterministic* PTM has a single initial state, and the transitions $x \xrightarrow{\sigma} \acute{x}$ and $x \xrightarrow{\sigma} \grave{x}$ always imply $\acute{x} = \grave{x}$. Many properties of a PTM are naturally expressed by its STM representation (4). One