Coupling event domain and time domain models for manufacturing systems

DOI:
10.1109/CDC.2006.377701

Document status and date:
Published: 01/01/2006

Document Version:
Accepted manuscript including changes made at the peer-review stage

Coupling event domain and time domain models of manufacturing systems

J.A.W.M. van Eekelen, E. Lefeber and J.E. Rooda

***Abstract*— Manufacturing systems are often characterized as discrete event systems (DES) and consequently, these systems are modeled with discrete event models. For certain discrete event modeling paradigms, control theory/techniques have been developed in event domain. However, from a control or performance perspective, a lot of notions are time related, like stability, settling time, transient behavior, throughput, flow time, efficiency, etc. Moreover, if we also consider market/customer requirements, almost all requirements are within time perspective: due dates, deliverability, earliness, tardiness, etc. Therefore, it is also useful to have time driven models of manufacturing systems. To combine the insights in modeling and control obtained in both time and event domain, it is useful to create a coupling between those two domains.**

This paper describes modeling techniques in both time domain and event domain for a class of manufacturing systems and establishes a generic coupling between two model descriptions. The coupling exists of two maps between the models' states, enabling real-time control of manufacturing systems.

I. INTRODUCTION

Manufacturing systems are often characterized as discrete event systems and therefore modeled as discrete event systems. These models are event driven, i.e. events occur and time labels are assigned to events. Different modeling techniques exist for discrete event systems. A detailed overview of modeling techniques is presented by Cassandras and Lafortune in [3]. Control methods have been developed in event domain. Cottenceau et al. present a modeling and control framework for timed event graphs in dioids in [4]. For max-plus models in a model predictive control framework, a lot of work has been done by De Schutter and Van den Boom, for example in [5]. A great advantage of some discrete event modeling paradigms is that models are simple and elegant, and scale up linearly when enlarging the system to be modeled. However, a disadvantage of discrete event models is that a mathematical background for analysis in the time domain is hard to establish. Moreover, especially in the timed event graph and dioid paradigm, the models can not deal with initial conditions of a manufacturing system, which makes practical relevance doubtful. In event domain, no state exists as a function of time, which makes real-time control difficult, if not impossible. In [6] we developed a new characterization of the state as a function of time for manufacturing systems (which by nature are event driven), which works very intuitively. This characterization is well suited for incorporating initial conditions of a manufacturing system. We even presented a first working example of real-time control using state feedback with this new type of state.

In physical manufacturing systems, although often event driven, events occur while time elapses and a lot of control and performance notions are specified in time domain. One could think of stability issues, transient behavior, throughput measurements and flow time of jobs. In addition, for time domain systems (not necessarily manufacturing systems) a lot of control methods have been developed. This is the reason why we want to enable real-time control of discrete event systems using time domain notions.

In this paper we present a framework in which event and time domain models can be coupled. This facilitates use of the advantages of both domains and maybe lose the disadvantages of some modeling techniques. We present maps between states of models in different modeling paradigms. Furthermore, we add the requirement that the maps are generic in a way that when manufacturing systems grow, the models and maps between states grow proportionally.

The remainder of this paper is organized as follows. In Section II definitions and notations are presented for dynamical systems and state space dynamical systems. In Section III we develop models for the most basic manufacturing system: one workstation. Three different modeling techniques are used, for both event domain and time domain. In Section IV maps are presented to couple state vectors from different model descriptions. In this way, time domain and event domain models can be interconnected for analysis and real-time control, combining all advantages of the two domains. Finally, an example is given which illustrates the different state forms and maps.

II. (STATE SPACE) DYNAMICAL SYSTEMS

A. Class of manufacturing systems under consideration

In this research we consider manufacturing systems in which only synchronization occurs. All product recipes, orders and routes are predetermined and all system parameters are deterministic. Moreover, we only consider systems where processing a job takes a significant amount of time. Examples of manufacturing systems under consideration are: buffers, single-lot machines, batch machines and assembly stations. These examples are building blocks from which larger manufacturing systems can be constructed by means of interconnection. In this paper, we restrict ourselves to an elementary building block: a workstation, consisting of a first-in-first-out buffer with finite capacity and a single-lot machine. However, all concepts presented here are suitable for the manufacturing building blocks as described above.

All authors are with the Department of Mechanical Engineering, Systems Engineering Group, Technische Universiteit Eindhoven, The Netherlands [j.a.w.m.v.eekelen,a.a.j.lefeber,j.e.rooda]@tue.nl

B. Definitions and notational aspects

The manufacturing systems are considered to be *dynamical systems*, as described by Willems in [9]. For time domain systems we have:

Definition 2.1: A *time domain dynamical system* $\Sigma_{\mathscr{T}}$ is a triple $\Sigma_{\mathscr{T}} = (\mathbb{T},\ \mathbb{W}_{\mathscr{T}},\ \mathscr{B}_{\mathscr{T}})$ with $\mathbb{T}$ the *time axis*, $\mathbb{W}_{\mathscr{T}}$ the *signal space* in which certain time driven signals, containing event counters, take on their values and $\mathscr{B}_{\mathscr{T}}$ the *model*, the subset of $\mathbb{W}_{\mathscr{T}}$ to which all allowable time trajectories of the system belong.

The general model of a manufacturing system in time domain can be written as:

$$\mathscr{B}_{\mathscr{T}} = \left\{ \mathbf{w}_{\mathscr{T}} \;\middle|\; \begin{array}{l} \sigma^{\tau}\mathbf{w}_{\mathscr{T}} \leq \mathbf{w}_{\mathscr{T}},\ \forall \tau > 0 \\ \text{“Physical laws of system satisfied”} \end{array} \right\} \quad (1)$$

where σ is the time shift operator: $\sigma^{\tau} w_{\mathscr{T}}(t) \triangleq w_{\mathscr{T}}(t-\tau)$. We only allow non-decreasing signals in vector $\mathbf{w}_{\mathscr{T}}$ since we want to be able to transform these signals into the event domain, as explained in [8]. The phrase *“Physical laws of system satisfied”* contains constraints on product recipes, routes, capacities and production policies.

Definition 2.2: A *time domain state space dynamical system* $\Sigma_{\mathscr{T}s}$ is a quadruple $\Sigma_{\mathscr{T}s} = (\mathbb{T},\ \mathbb{W}_{\mathscr{T}},\ \mathbb{X}_{\mathscr{T}},\ \mathscr{B}_{\mathscr{T}s})$ where $\mathbb{X}_{\mathscr{T}}$ is the space of the *state* variables and $\mathscr{B}_{\mathscr{T}s}$ is called the *full behavior* of the system.

The state variables specify the internal memory of a dynamical system. Formally, the axiom of state is given in [9]. Informally, one could say that the *state should contain sufficient information about the past so as to determine future behavior* [9].

Note that the signalspace $\mathbb{W}_{\mathscr{T}}$ contains values for event *counters*. As this name implies, the counters are all non-decreasing signals, which is relevant in the remainder of this paper (also explained in [8]).

Complementary to the time domain dynamical system and state space system, we define the event domain dynamical system as follows:

Definition 2.3: An *event domain dynamical system* $\Sigma_{\mathscr{K}}$ is a triple $\Sigma_{\mathscr{K}} = (\mathbb{K},\ \mathbb{W}_{\mathscr{K}},\ \mathscr{B}_{\mathscr{K}})$ with $\mathbb{K}$ the event counter, $\mathbb{W}_{\mathscr{K}}$ the *signal space* in which event driven signals, containing time instances, take on their values, and $\mathscr{B}_{\mathscr{K}}$ the *model*, a subset of $\mathbb{W}_{\mathscr{K}}$, to which all allowable event trajectories of the system belong.

Note that since we assume that events take place in the ‘right order’, i.e. the first event does not take place after the second event and so on, the signals in $\mathbb{W}_{\mathscr{K}}$ are also non-decreasing signals.

The general model of a manufacturing system in event domain can now be written as:

$$\mathscr{B}_{\mathscr{K}} = \left\{ \mathbf{w}_{\mathscr{K}} \;\middle|\; \begin{array}{l} \gamma^{k}\mathbf{w}_{\mathscr{K}} \leq \mathbf{w}_{\mathscr{K}},\ \forall k > 0 \\ \text{“Physical laws of system satisfied”} \end{array} \right\} \quad (2)$$

where γ is the event shift operator: $\gamma^{n} w_{\mathscr{K}}(k) \triangleq w_{\mathscr{K}}(k-n)$.

Definition 2.4: An *event domain state space dynamical system* $\Sigma_{\mathscr{K}s}$ is a quadruple $\Sigma_{\mathscr{K}s} = (\mathbb{K},\ \mathbb{W}_{\mathscr{K}},\ \mathbb{X}_{\mathscr{K}},\ \mathscr{B}_{\mathscr{K}s})$ where $\mathbb{X}_{\mathscr{K}}$ is the space of the *state* variables and $\mathscr{B}_{\mathscr{K}s}$ is called the *full behavior* of the system.

Now that we have defined the manufacturing systems under consideration and the type of signals we allow in the models, we can use modeling techniques to describe the evolution of the signals in both time and event domain.

III. MODELING A WORKSTATION

In this section, we present a basic building block of manufacturing systems: a workstation. First we specify the dynamics in an informal way and then we use three modeling techniques to explicitly model the dynamics of this workstation.

A. Informal description

Consider the workstation with finite buffer B and single-lot machine M as shown in Fig. 1. Buffer B has a capacity of N lots. Machine M has a constant process time of d time units. Lots are pushed through the workstation, i.e. the machine never stays idle when lots reside in the buffer. In the next sections, this informal description of the dynamics is elaborated in different time domain and event domain models.

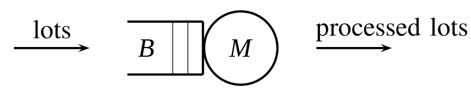

Fig. 1. Workstation with finite buffer B and single lot machine M.

B. Max-plus model: event domain

We use max-plus algebra to obtain a model description in event domain. For an introduction to max-plus algebra with references, we refer to [7]. Informally, the max-plus algebra consists of two operators: $\otimes$ or max-plus multiplication, and $\oplus$ or max-plus addition. The operators are defined as follows:

$$a \otimes b \triangleq a + b \quad \text{and} \quad a \oplus b \triangleq \max(a,\ b) \qquad (3)$$

with $a,\ b \in \mathbb{R} \cup \{-\infty\}$. For better readability, we use the conventional $+$ and max operators in the remainder of this paper. Let $w_{\mathscr{K}_u}$ be the signal denoting the time instances at which lots arrive at the workstation. $w_{\mathscr{K}_1}$ denotes the signal containing the time instances lots enter the buffer, while $w_{\mathscr{K}_2}$ is the signal containing the time instances lots leave the workstation after being processed. Furthermore, $w_{\mathscr{K}_u}(k)$ denotes the time instance a lot arrives at the workstation for the k^{th} time, and similarly for $w_{\mathscr{K}_1}(k)$ and $w_{\mathscr{K}_2}(k)$.

A lot enters the buffer as soon as it has arrived and an empty space is available in the buffer. A lot leaves the workstation as soon as it has been processed. Only one lot can be processed at a time. The model description is then given by:

$$w_{\mathscr{K}_1}(k) = \max\left(w_{\mathscr{K}_u}(k),\ w_{\mathscr{K}_2}(k-N-1)\right) \qquad (4)$$

$$w_{\mathscr{K}_2}(k) = \max\left(w_{\mathscr{K}_1}(k) + d,\ w_{\mathscr{K}_2}(k-1) + d\right) \qquad (5)$$

The event driven max-plus model of the workstation then becomes: (the $\oplus$ superscript indicates the modeling technique)

$$\mathscr{B}^{\oplus}_{\mathscr{K}} = \left\{ \mathbf{w}_{\mathscr{K}} : \mathbb{Z} \to \mathbb{R}^3 \,\middle|\, \gamma^{k}\mathbf{w}_{\mathscr{K}} \leq \mathbf{w}_{\mathscr{K}},\ \forall k > 0;\ (4);\ (5) \right\} \quad (6)$$

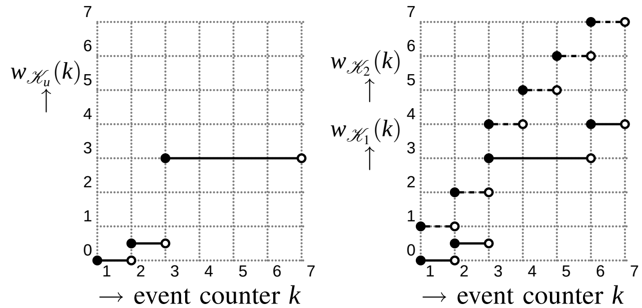

Fig. 2. Example of signals $w_{\mathscr{K}_u}(k)$ (left), $w_{\mathscr{K}_1}(k)$ (solid) and $w_{\mathscr{K}_2}(k)$ (dashed) for one workstation with $N = 2$ and $d = 1$.

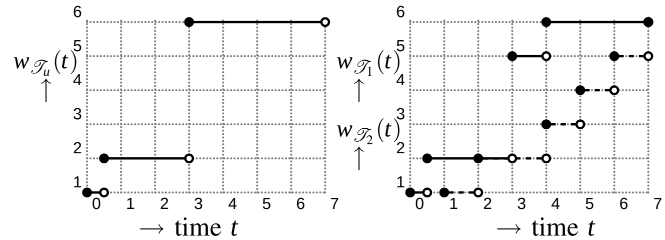

Fig. 3. Example of signals $w_{\mathscr{K}_u}(k)$ (left), $w_{\mathscr{K}_1}(k)$ (solid) and $w_{\mathscr{K}_2}(k)$ (dashed) for one workstation with $N = 2$ and $d = 1$.

with $\mathbf{w}_{\mathscr{K}} = \begin{bmatrix} w_{\mathscr{K}_u} & w_{\mathscr{K}_1} & w_{\mathscr{K}_2} \end{bmatrix}^T$. Note that modeling in max-plus algebra is very generic, since adding a second workstation in line would simply be a matter of adding 1 max-algebraic equation to (4) and (5). In Fig. 2 an example is given of the signals in this behavior with $N = 2$ and $d = 1$. The horizontal axis is the event counter, while the vertical axis is a time axis. The graphs thus show the *time instances* at which *events* occur. In this figure, we see that 4 lots arrive at the workstation at the same time (time = 3), but they can not enter the buffer all at once, because of the limited capacity: $w_{\mathscr{K}_1}(6) = 4$ while $w_{\mathscr{K}_u}(6) = 3$. The sixth lot must remain at its source until time = 4. This can be a preceding workstation, which is blocked because it can not send the lot away when possible. Signal $w_{\mathscr{K}_u}$ can be regarded as input signal. We restrict ourselves to right-continuous input signals, also resulting in right-continuous signals $w_{\mathscr{K}_1}$ and $w_{\mathscr{K}_2}$.

Note that the vector $\mathbf{w}_{\mathscr{K}}(k)$ in this model is *not* the state of the system. A state contains information of the signals over an event horizon. This is not elaborated here. Details are provided in [9] and [8].

C. Min-plus model: time domain

The workstation described in III can also be modeled in time domain, for example by means of min-plus algebra. Closely related to the max-plus algebra, formal aspects of this algebra can be found in [1]. The min-plus algebra consists of two operators, $\otimes$ or min-plus multiplication and $\ominus$ or min-plus substraction. The operators are defined as:

$$a \otimes b \triangleq a + b \quad \text{and} \quad a \ominus b \triangleq \min(a,\, b) \tag{7}$$

with $a,\, b \in \mathbb{R} \cup \{\infty\}$. Again, we only use the conventional $+$ and min operators in our expressions.

Let $w_{\mathscr{T}_u}$ be the signal denoting the number of lots that arrived at the workstation. $w_{\mathscr{T}_1}$ denotes the signal containing the number of lots that entered the buffer, while $w_{\mathscr{T}_2}$ is the signal containing the number of lots that have left the workstation after being processed. Furthermore, $w_{\mathscr{T}_u}(t)$ denotes the number of lots that have arrived at the workstation at time t, and similarly for $w_{\mathscr{T}_1}(t)$ and $w_{\mathscr{T}_2}(t)$.

The dynamics in time domain perspective can now be described as: The number of lots that have entered the buffer at time t equals the minimum of the number of available lots and the number of lots that have left the workstation minus the complete capacity of the workstation. In addition, the number of lots that have left the workstation equals the minimum of lots that has entered the buffer and of the number of lots that had left d time units ago. In other words:

$$w_{\mathscr{T}_1}(t) = \min\left(w_{\mathscr{T}_u}(t),\; w_{\mathscr{T}_2}(t) + N + 1\right) \tag{8}$$

$$w_{\mathscr{T}_2}(t) = \min\left(w_{\mathscr{T}_1}(t-d),\; w_{\mathscr{T}_2}(t-d) + 1\right) \tag{9}$$

The time driven min-plus model of the workstation then becomes: (the $\ominus$ superscript indicates the modeling technique)

$$\mathscr{B}^{\ominus}_{\mathscr{T}} = \left\{ \mathbf{w}_{\mathscr{T}} : \mathbb{R} \to \mathbb{Z}^3 \,\middle|\, \sigma^{\tau}\mathbf{w}_{\mathscr{T}} \leq \mathbf{w}_{\mathscr{T}},\, \forall \tau > 0;\, (8);\, (9) \right\} \tag{10}$$

with $\mathbf{w}_{\mathscr{T}} = \begin{bmatrix} w_{\mathscr{T}_u} & w_{\mathscr{T}_1} & w_{\mathscr{T}_2} \end{bmatrix}^T$. Note that similar to max-plus model (6), min-plus model (10) is also scalable: adding more workstations makes the model grow proportionally.

The signals in $\mathbf{w}_{\mathscr{T}}$ for the same example as Fig. 2 are shown in Fig. 3. Note that vector $\mathbf{w}_{\mathscr{T}}(t)$ in this model is *not* the state of the system. In order to have sufficient information from the past at time t as to determine future behavior, we need the information from the signals over at least memory span Δ. From [9], [8] we know that this memory span Δ must be greater than or equal to d. If we regard the signal $w_{\mathscr{T}_u}$ as the input of a state space model, we define state variables $x_{\mathscr{T}_i}(t)$ as:

$$x_{\mathscr{T}_i}(t) : [-\Delta,\, 0] \to \mathbb{Z} \text{ with } x_{\mathscr{T}_i}(t)(\tau) = w_{\mathscr{T}_i}(t+\tau) \tag{11}$$

with $i \in \{1, 2\}$ and the full behavior of the workstation is:

$$\mathscr{B}^{\ominus}_{\mathscr{T}s} = \left\{ \begin{array}{l} \mathbf{w}_{\mathscr{T}} : \mathbb{R} \to \mathbb{Z}^3 \\ \mathbf{x}_{\mathscr{T}} : \mathbb{R} \to ([-\Delta,\, 0] \to \mathbb{Z})^2 \end{array} \,\middle|\, \begin{array}{l} \sigma^{\tau}\mathbf{w}_{\mathscr{T}} \leq \mathbf{w}_{\mathscr{T}},\, \forall \tau > 0 \\ (8),\, (9),\, (11) \\ \Delta \geq d \end{array} \right\} \tag{12}$$

with $\mathbf{x}_{\mathscr{T}}(t) = \begin{bmatrix} x_{\mathscr{T}_1}(t) & x_{\mathscr{T}_2}(t) \end{bmatrix}^T$. Note that this state can be infinitely dimensional (especially in case of infinite buffer capacity), since it consists of piece-wise constant signals over time interval Δ (as defined in (11)).

D. Hybrid model in χ formalism

Another way of modeling the dynamics of the workstation is by means of formalism χ. This formalism is suited for modeling, simulation and analysis of hybrid systems, i.e. with discrete event dynamics and continuous dynamics, in a formal and mathematically unambiguous way. For a detailed description of the χ formalism, the reader is referred to [2]. In this subsection, we present the χ-model and explain it in an informal way. We also present an example of state variables evolution in this formalism.

The idea for modeling the workstation in χ is that the concept of using a memory span, which was needed to determine a state in min-plus modeling, can be omitted by introducing a continuous state variable in time, representing the remaining process time of the machine, $\overline{x}_3(t) \in [0,d]$. As discrete state variables we introduce the number of lots in the buffer, $\overline{x}_1(t) \in \{0,1,2\}$, the number of lots on the machine, $\overline{x}_2(t) \in \{0,1\}$ and the number of finished lots, $\overline{x}_4(t) \in \mathbb{Z}$. This way of defining the state of a manufacturing system (with these variable types) was first introduced in [6]. We define the state of the workstation as:

$$\overline{\mathbf{x}}(t) = \begin{bmatrix} \overline{x}_1(t) \in \{0,1,2\} \\ \overline{x}_2(t) \in \{0,1\} \\ \overline{x}_3(t) \in [0,d] \\ \overline{x}_4(t) \in \mathbb{Z} \end{bmatrix} = \begin{array}{l} \text{\# of lots in buffer } B \\ \text{\# of lots on machine} \\ \text{remaining process time} \\ \text{\# of finished lots} \end{array} \tag{13}$$

(Remark: for the number of finished lots we use the integer type $\mathbb{Z}$, since the event counters in max-plus models have been defined in $\mathbb{Z}$.)

We now present a χ-model of the workstation, which is explained afterwards.

$$\begin{array}{l}
\mathsf{proc}\ G(\mathsf{chan}\ a\,!:\mathsf{void},\ \mathsf{alg}\ w_{\mathcal{T}_u}:\mathsf{int}) = \\
[\![\ \mathsf{var}\ n:\mathsf{nat}=0,\ pu:\mathsf{int}=w_{\mathcal{T}_u} \\
::\ *(\ w_{\mathcal{T}_u} \neq pu \to n := n + w_{\mathcal{T}_u} - pu;\ pu := w_{\mathcal{T}_u} \\
\quad [\!]\ n > 0 \to a\,!;\ n := n-1 \\
\quad) \\
]\!] \\
\\
\mathsf{proc}\ W(\mathsf{chan}\ a\,?, b\,!:\mathsf{void},\ \mathsf{val}\ d:\mathsf{real},\ N:\mathsf{nat} \\
\qquad ,\ \overline{w}_{\mathcal{T}_1}:\mathsf{int},\ \overline{x}_1,\ \overline{x}_2:\mathsf{nat},\ \tilde{x}_3:\mathsf{real}) = \\
[\![\ \mathsf{cont}\ \overline{x}_3:\mathsf{real}=\tilde{x}_3 \\
::\ *(\ \overline{x}_1 < N \to a\,?;\ \overline{w}_{\mathcal{T}_1} := \overline{w}_{\mathcal{T}_1}+1;\ \overline{x}_1 := \overline{x}_1+1 \\
\quad [\!]\ \overline{x}_1 > 0\ \wedge\ \overline{x}_2 = 0 \to \overline{x}_1 := \overline{x}_1 - 1;\ \overline{x}_2 := 1;\ \overline{x}_3 := d \\
\quad [\!]\ \overline{x}_2 > 0\ \wedge\ \overline{x}_3 = 0 \to b\,!;\ \overline{x}_2 := 0 \\
\quad) \\
\|\ \overline{x}_3 > 0 \Rightarrow \dot{\overline{x}}_3 = -1 \\
\|\ \overline{x}_3 = 0 \Rightarrow \dot{\overline{x}}_3 = 0 \\
]\!] \\
\\
\mathsf{proc}\ E(\mathsf{chan}\ b\,?:\mathsf{void},\ \mathsf{val}\ \overline{w}_{\mathcal{T}_2},\ \overline{x}_4:\mathsf{int}) = \\
[\![\ *(\ b?;\ \overline{x}_4 := \overline{x}_4+1;\ \overline{w}_{\mathcal{T}_2} := \overline{x}_4\ )\ ]\!] \\
\\
\mathsf{model}\ S(\mathsf{alg}\ w_{\mathcal{T}_u}:\mathsf{int},\ \mathsf{val}\ d:\mathsf{real},\ N:\mathsf{nat}) = \\
[\![\ \mathsf{chan}\ a,\ b:\mathsf{void} \\
::\ G(a,\ w_{\mathcal{T}_u}) \\
\|\ W(a,\ b,\ d,\ N,\ \overline{w}_{\mathcal{T}_{10}},\ \overline{x}_{10},\ \overline{x}_{20},\ \overline{x}_{30}) \\
\|\ E(b,\ \overline{w}_{\mathcal{T}_{20}},\ \overline{x}_{40}) \\
]\!]
\end{array} \tag{14}$$

A schematic representation of this χ-model is given in Fig. 4. The χ-model consists of 4 parts: 3 processes and 1 model. Model S consists of 3 *concurrent* processes: lot generator G, workstation W and exit process E. The processes are interconnected by means of channels a and b. Some parameters are passed to workstation W: buffer capacity N and process time d. Moreover, initial conditions for the state elements are constants (indicated with subscript 0) and passed to W and E. Process G sends lots to the workstation and has an input signal $w_{\mathcal{T}_u}(t)$, which is similar to the input in the min-plus model. The number of lots that can not be sent immediately to the workstation (due to a fully loaded buffer) is stored in variable n. Process E receives lots from the workstation after they have been finished and the process updates state element $\overline{x}_4(t)$ and variable $\overline{w}_{\mathcal{T}_2}(t)$. Workstation W is a bit more complex. The three lines within the $*($ and $)$ parentheses are 3 *alternatives* (separated by $[\!]$) which can be executed if the guard before the $\to$ arrow evaluates to true. Informally, the lines say: if empty space exists in the buffer, it can accept lots, which are stored in the buffer. If the machine is idle and there is a lot in the buffer, put it on the machine, and if the machine has finished a lot, try to send it to the next process. The final 2 lines of process W represent the continuous dynamics, stating that if the remaining process time of a lot is positive, it should decrease linearly over time with slope -1 and if the remaining process time is 0, it should remain 0 (until the next lot is put on the machine).

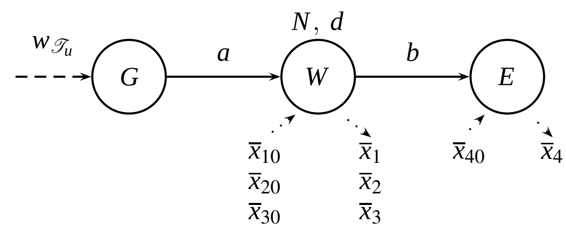

Fig. 4. Schematic representation of χ-model in (14).

Since elementary building blocks proc are specified separately from the model, hybrid χ models are very generic, since adding an extra workstation (with different parameters!) is a matter of adding one line to model S. In this way, very large systems can be modeled using a relatively small specification. The state space dynamical model $\overline{\mathscr{B}}^{\chi}_{\mathscr{T}s}$ of the workstation is now given by:

$$\overline{\mathscr{B}}^{\chi}_{\mathscr{T}s} = \left\{ \begin{array}{l} \overline{\mathbf{w}}_{\mathscr{T}} : \mathbb{R} \to \mathbb{Z}^3 \\ \overline{\mathbf{x}} : \mathbb{R} \to \mathbb{N}^2 \times \mathbb{R} \times \mathbb{Z} \end{array} \middle| \ \chi\text{-model (14)} \right\} \tag{15}$$

with $\overline{\mathbf{w}}_{\mathscr{T}} = \begin{bmatrix} w_{\mathscr{T}_u}(t) & \overline{w}_{\mathscr{T}_1}(t) & \overline{w}_{\mathscr{T}_2}(t) \end{bmatrix}^T$.

A major difference with the max-plus or min-plus models is that due to the discrete event nature, signals are piecewise linear functions over closed intervals. In other words: at time instances where events take place, they can have multiple values. Since we do not want this property in coupling different model types, we want to map the signals from our χ-model onto right continuous signals. The physical meaning of right-continuous signals in this case is that the state on a certain time instance is measured only if all events that can happen at that time instance have taken place. The resulting state space dynamical model is then:

$$\mathscr{B}^{\chi}_{\mathscr{T}s} = \left\{ \begin{array}{l} \mathbf{w}_{\mathscr{T}} : \mathbb{R} \to \mathbb{Z}^3 \\ \mathbf{x} : \mathbb{R} \to \mathbb{N}^2 \times \mathbb{R} \times \mathbb{Z} \end{array} \right| \\ \left. \begin{array}{l} \exists \overline{\mathbf{w}}_{\mathscr{T}}(t) \in \overline{\mathscr{B}}^{\chi}_{\mathscr{T}s} \text{ s.t. } \forall\, t \in \mathbb{R} : \mathbf{w}_{\mathscr{T}}(t) = \lim_{\bar{t}\downarrow t} \overline{\mathbf{w}}_{\mathscr{T}}(\bar{t}) \\ \exists \overline{\mathbf{x}}(t) \in \overline{\mathscr{B}}^{\chi}_{\mathscr{T}s} \text{ s.t. } \forall\, t \in \mathbb{R} : \mathbf{x}(t) = \lim_{\bar{t}\downarrow t} \overline{\mathbf{x}}(\bar{t}) \end{array} \right\} \tag{16}$$

with $\mathbf{w}_{\mathscr{T}}(t) = \begin{bmatrix} w_{\mathscr{T}_u}(t) & w_{\mathscr{T}_1}(t) & w_{\mathscr{T}_2}(t) \end{bmatrix}^T$ and $\mathbf{x}(t)$ the right-continuous variant of $\overline{\mathbf{x}}(t)$. The state evolution from the right-continuous χ-model for the same situation as in Figures 2

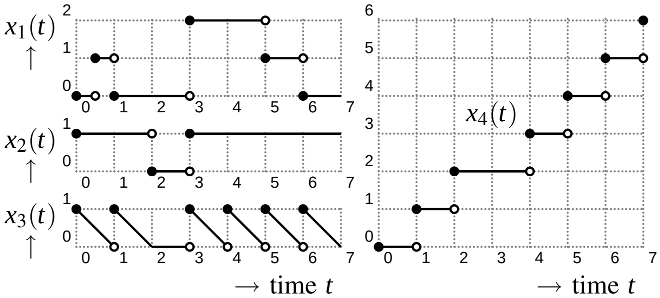

Fig. 5. Example of state elements $x_1(t)$, $x_2(t)$, $x_3(t)$ and $x_4(t)$ for one workstation with $N = 2$ and $d = 1$, modeled in χ.

and 3 is shown in Fig. 5. Properties (not exhaustive) of the right-continuous state:

$$\text{if } x_2(t) = 0 \text{ then } x_1(t) = 0 \text{ and } x_3(t) = 0 \tag{17}$$

$$\text{if } x_1(t) \neq 0 \text{ then } x_2(t) \neq 0 \tag{18}$$

or in words: if the machine is idle, then the buffer must be empty and the remaining process time equals 0 (Eq. (17)) and if the buffer is not empty, a lot must reside on the machine (Eq. (18)). These properties do also show in Fig. 5.

Now that we have specified a workstation in a pure event domain model (max-plus), pure time domain (min-plus) and a hybrid form, in which events take place and the state can be measured continuously over time, we investigate the coupling between these models.

IV. MAPS BETWEEN STATE TRAJECTORIES

So far, three different models have been made of the workstation, in event domain (6), in time domain (10), and a hybrid form (15). A coupling between the models facilitates the use of analysis and control methods from both domains. In order to establish a coupling between the model descriptions, we want to find a map between the state trajectories of the different models, since the state of the system contains sufficient information about the past to determine future behavior. The three models all represent the same physical manufacturing system, so having information about the past in one model form should be sufficient to construct the state in a different model form. In this section, we first treat the map between min-plus and max-plus states shortly in Section IV-A. Then, in Section IV-B, we focus on the coupling between the min-plus model and the hybrid χ-model.

A. Coupling the max-plus and min-plus model

Max-plus models (in event domain) and min-plus models (in time domain) of the same physical system are closely related. The coupling between the models has been investigated in [8], where π is a bijection (both surjective and injective map) and n is the number of signals in $\mathbf{w}_{\mathscr{K}}$.

$$\mathbf{w}_{\mathscr{K}_\mathbf{i}}(k) = \pi(\mathbf{w}_{\mathscr{T}_\mathbf{i}}(t)) = \inf_{\mathbf{w}_{\mathscr{T}_\mathbf{i}}(t) \geq k,\ t \in \mathbb{R}} t,\ i \in \{1, \ldots, n\},\ \forall k \in \mathbb{Z} \tag{19}$$

The inverse map can be presented in a similar way. Loosely speaking, this map swaps the axes of a counter-time graph in min-plus modeling to obtain the time-event graph in max-plus modeling (cf. Figures 3 and 2). More details, conditions and properties of this coupling between max-plus and min-plus models is given in [8].

B. Coupling the min-plus model and hybrid χ model

The state in the min-plus model (as defined in (11)) is (in general) infinitely dimensional (signals over a time interval $\Delta \geq d$, while the state of the hybrid χ model consists of 4 scalars. We want to map these states onto each other, to establish the coupling between the two model descriptions. So the map from min-plus state to hybrid χ state constructs scalars from a signal over an interval, the map from hybrid χ state to min-plus state constructs a signal over time interval Δ.

Proposition 4.1: For any state $\mathbf{x}_{\mathscr{T}}(t)$ in the min-plus model, we can find a corresponding hybrid χ state $\mathbf{x}(t) \in \mathscr{B}^{\chi}_{\mathscr{T}_S}$ using map $\mathscr{M}_{\ominus\to\chi} : ([-\Delta, 0] \to \mathbb{Z})^2 \to \mathbb{N}^2 \times \mathbb{R} \times \mathbb{Z}$:

$$\begin{aligned}
x_1(t) &= \max(0,\ x_{\mathscr{T}_1}(t)(0) - x_{\mathscr{T}_2}(t)(0) - 1) \\
x_2(t) &= \min(x_{\mathscr{T}_1}(t)(0) - x_{\mathscr{T}_2}(t)(0),\ 1) \\
x_3(t) &= \begin{cases} 0 \quad \text{if } x_{\mathscr{T}_1}(t)(0) = x_{\mathscr{T}_2}(t)(0) \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \max\left(\inf\{\tau \in [-\Delta,\ 0] | x_{\mathscr{T}_2}(t)(\tau) = x_{\mathscr{T}_2}(t)(0)\} + d,\ 0\right) \\ \qquad \text{if } x_{\mathscr{T}_1}(t)(-\Delta) \geq x_{\mathscr{T}_2}(t)(0) + 1 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \max\begin{pmatrix} \inf\{\tau \in [-\Delta,\ 0] | x_{\mathscr{T}_2}(t)(\tau) = x_{\mathscr{T}_2}(t)(0)\} + d \\ \inf\{\tau \in [-\Delta,\ 0] | x_{\mathscr{T}_1}(t)(\tau) = x_{\mathscr{T}_2}(t)(0) + 1\} + d \\ 0 \end{pmatrix} \\ \qquad \text{if } x_{\mathscr{T}_1}(t)(-\Delta) < x_{\mathscr{T}_2}(t)(0) + 1 \\ \qquad \text{and } x_{\mathscr{T}_1}(t)(0) > x_{\mathscr{T}_2}(t)(0) \end{cases} \\
x_4(t) &= x_{\mathscr{T}_2}(t)(0)
\end{aligned} \tag{20}$$

In (20) we recognize the infima which were present in (19). Map $\mathscr{M}_{\ominus\to\chi}$ has a simple and elegant structure (the cases in $x_3(t)$ only make sure that all infima exist), which makes it very suitable also for larger manufacturing systems.

Proposition 4.2: For any state $\mathbf{x}(t)$ in the hybrid χ model, we can find a corresponding non-empty set of min-plus states $\mathbf{x}_{\mathscr{T}}(t) \in \mathscr{B}^{\ominus}_{\mathscr{T}_S}$ using map $\mathscr{M}_{\chi\to\ominus}$:$\mathbb{N}^2 \times \mathbb{R} \times \mathbb{Z} \to ([-\Delta, 0] \to \mathbb{Z})^2$:

$$\begin{aligned}
\{\mathbf{x}_{\mathscr{T}}(t) \mid\ & \exists \tilde{\mathbf{x}}_{\mathscr{T}}(t) : [-\Delta - d,\ 0] \to \mathbb{Z}^2 \text{ with } \Delta \geq d \text{ subject to:} \\
& \cdots\cdots\cdots\cdots\cdots\cdots \\
& \tilde{x}_{\mathscr{T}_1}(t)(\tau) \leq \tilde{x}_{\mathscr{T}_2}(t)(\tau) + N_1 + 1 \\
& \tilde{x}_{\mathscr{T}_2}(t)(\tau) = \min\left(\tilde{x}_{\mathscr{T}_1}(t)(\tau - d),\ \tilde{x}_{\mathscr{T}_2}(t)(\tau - d) + 1\right) \\
& \cdots\cdots\cdots\cdots\cdots\cdots \\
& \tilde{x}_{\mathscr{T}_1}(t)(0) = x_4(t) + x_2(t) + x_1(t) \\
& \tilde{x}_{\mathscr{T}_2}(t)(\tau) = \begin{cases} x_4(t) \text{ for } \tau \in [x_3(t) - d,\ 0] \text{ if } x_2(t) > 0 \\ x_4(t) \text{ for } \tau = 0 \text{ if } x_2(t) = 0 \end{cases} \\
& \tilde{x}_{\mathscr{T}_1}(t)(x_3(t) - d) \geq x_4(t) + 1 \text{ if } x_2(t) > 0 \\
& \cdots\cdots\cdots\cdots\cdots\cdots \\
& \tilde{x}_{\mathscr{T}_i}(t)(\tau - \varepsilon) \leq \tilde{x}_{\mathscr{T}_i}(t)(\tau) \text{ with } i \in \{1,\ 2\},\ \varepsilon > 0 \\
& \cdots\cdots\cdots\cdots\cdots\cdots \\
& \forall \tau \in [-\Delta,\ 0] \text{ and } i \in \{1,\ 2\} : x_{\mathscr{T}_i}(t)(\tau) = \tilde{x}_{\mathscr{T}_i}(t)(\tau)\}
\end{aligned} \tag{21}$$

Map $\mathscr{M}_{\chi\to\ominus}$ constructs the state elements in $\mathbf{x}_{\mathscr{T}}(t)$, which are a function over time interval $\Delta \geq d$, as defined in (11). Since in general many states $\mathbf{x}_{\mathscr{T}}(t)$ in the min-plus model map to the same state $\mathbf{x}(t)$ in the hybrid χ model, a set of states $\mathbf{x}_{\mathscr{T}}(t)$ corresponds to a single state $\mathbf{x}(t)$. Therefore, map $\mathscr{M}_{\chi\to\ominus}$ returns a non-empty *set of states*. One can interpret this set as all possible states of the min-plus models that could have led to the current physical situation in the manufacturing system. Map $\mathscr{M}_{\chi\to\ominus}$ yields in the complete set of feasible trajectories. In (21) we recognize the dynamics equations (8) and (9), conditions on the upper bound of interval Δ, and the requirement for non-decreasing signals. This map therefore can easily be extended for larger manufacturing systems, which was one of our goals. Furthermore, we used as auxiliary state $\tilde{\mathbf{x}}_{\mathscr{T}}(t)$ to be able to construct feasible signals at the left boundary of interval Δ, but this is of no importance for understanding the map.

Both maps are complementary, i.e. if we start with a right-continuous hybrid χ state $\mathbf{x}(t)$, map it to a set of min-plus states $\mathbf{x}_{\mathscr{T}}(t)$ and then back again, we return to our original state. And the other way around: if we start with a min-plus state $\mathbf{x}_{\mathscr{T}}(t)$, map it to a hybrid χ state $\mathbf{x}(t)$ and then back to a set of min-plus states, the original state $\mathbf{x}_{\mathscr{T}}(t)$ lies within the resulting set. In other words:

Proposition 4.3:

$$\mathbf{x}(t) = \mathscr{M}_{\ominus\to\chi}(\mathscr{M}_{\chi\to\ominus}(\mathbf{x}(t))) \tag{22}$$

and

$$\mathbf{x}_{\mathscr{T}}(t) \in \mathscr{M}_{\chi\to\ominus}(\mathscr{M}_{\ominus\to\chi}(\mathbf{x}_{\mathscr{T}}(t))). \tag{23}$$

The proof follows straightforward by substituting (21) into (20) to obtain (22) and substituting (20) into (21) to obtain (23). Furthermore, the properties of the right-continuous hybrid χ state (17) and (18) and the definition of the hybrid χ state (13) are used.

C. Example

Consider the workstation as described in Section III, with buffer capacity $N = 2$ and process time $d = 1$. Six lots are pushed through the system. Lots were available at time 0, 0.5, 3, 3, 3 and 3. The flow of lots through the system is shown in Fig. 6. The horizontal axis is the time axis, the vertical axis shows the lot number. Blocks in the diagram indicate the presence of the lot in either the buffer or the machine. Note that although available at time $= 3$, the sixth lot can only enter the buffer at time $= 4$, due to the buffer capacity. Until time $= 4$, this lot must remain at its source, e.g. a preceding workstation. Figures 2, 3 and 5 were taken from this example. At time $= 4.5$, one lot is on the machine with remaining process time 0.5 and two lots are in the buffer. The number of already finished lots equals 3, so $\mathbf{x}(4.5) = \begin{bmatrix}2 & 1 & 0.5 & 3\end{bmatrix}^T$. Assume that memory span $\Delta = 2 \geq d$. Applying map $\mathscr{M}_{\chi\to\ominus}$ from (21) gives a set of solutions for $x_{\mathscr{T}_1}(4.5)$ and $x_{\mathscr{T}_2}(4.5)$. One possible realization is given in Fig. 7. This realization differs from the $w_{\mathscr{T}_1}$ and $w_{\mathscr{T}_2}$ graphs between time $= 2.5$ and 4.5 in Fig. 3, but it is a feasible realization of the past which leads to the current situation in the manufacturing system. Applying map $\mathscr{M}_{\ominus\to\chi}$ (20) on the realization of Fig. 7 yields the original state $\mathbf{x}(4.5) = \begin{bmatrix}2 & 1 & 0.5 & 3\end{bmatrix}^T$.

V. CONCLUSIONS

In this paper, we presented a generic way of coupling different model types for manufacturing systems by means of maps between the states of the models. Goal of this coupling method is to be able to use analysis techniques, real-time control methods, and performance measurement techniques which are either formulated in time domain or event domain. By coupling time domain and event domain models by state maps, we bridged the gap between those two domains. The models and maps presented in this paper are generic and scalable in a sense that enlarging the manufacturing system under consideration results in proportional growth of the models and maps. This is a great benefit compared to other modeling techniques, where the state grows exponentially when enlarging the system. Moreover, specifying manufacturing systems in the presented model techniques (max-plus, min-plus and hybrid χ) is straightforward i.e. the level of complexity does not increase when enlarging the system.

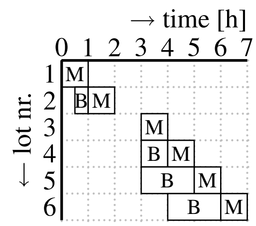

Fig. 6. Lot-time diagram.

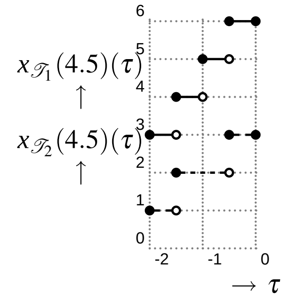

Fig. 7. Possible realization for $x_{\mathscr{T}_1}(4.5)$ (solid) and $x_{\mathscr{T}_2}(4.5)$ (dashed).

The methods and techniques in this paper were built up step-by-step and illustrated with an example of a workstation consisting of a buffer and a single-lot machine.

ACKNOWLEDGMENT

The authors gratefully thank Bas Roset and Henk Nijmeijer from the Dynamics and Control group for the discussions on the subject.

REFERENCES

[1] F. Baccelli, G. Cohen, G.J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity, An Algebra for Discrete Event Systems*. John Wiley and Sons, 1992.

[2] D.A. van Beek, K.L. Man, M.A. Reniers, J.E. Rooda, and R.R.H. Schiffelers. Syntax and consistent equation semantics of hybrid Chi. *Journal of Logic and Algebraic Programming*, 68(1–2):129–210, 2006.

[3] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

[4] B. Cottenceau, L. Hardouin, and J.L. Biomond. Model reference control for timed event graphs in dioids. *Automatica*, 37(8):1451–1458, August 2001.

[5] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049–1056, July 2001.

[6] J.A.W.M. van Eekelen, E. Lefeber, B.J.P. Roset, H. Nijmeijer, and J.E. Rooda. Control of manufacturing systems using state feedback and linear programming. In *Proceedings of the 44th IEEE Conference on Decision and Control and 2005 European Control Conference*, pages 4652–4657, 2005.

[7] B. Heidergott, G.J. Olsder, and J.W. van der Woude. *Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*. Princeton University Press, 2006.

[8] B.J.P. Roset, H. Nijmeijer, J.A.W.M. van Eekelen, E. Lefeber, and J.E. Rooda. Event driven manufacturing systems as time domain control systems. In *Proceedings of the 44th IEEE Conference on Decision and Control and 2005 European Control Conference*, pages 446–451, 2005.

[9] J.C. Willems. Paradigms and puzzles in the theory of dynamical systems. *IEEE Transactions on automatic control*, 36(3):259–294, March 1991.