



HAL
open science

CONFLICT MANAGEMENT AND RESOLUTION USING (MAX, +) ALGEBRA: APPLICATION TO SERVICES INTERACTION

Wafaa Ait-Cheik-Bihi, Ahmed Nait Sidi Moh, Maxime Wack

► **To cite this version:**

Wafaa Ait-Cheik-Bihi, Ahmed Nait Sidi Moh, Maxime Wack. CONFLICT MANAGEMENT AND RESOLUTION USING (MAX, +) ALGEBRA: APPLICATION TO SERVICES INTERACTION. 8th International Conference of Modeling and Simulation - MOSIM'10, May 2010, Hammamet, Tunisia. pp.9 Pages. hal-00470116

HAL Id: hal-00470116

<https://hal.science/hal-00470116>

Submitted on 3 Apr 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONFLICT MANAGEMENT AND RESOLUTION USING (MAX, +) ALGEBRA: APPLICATION TO SERVICES INTERACTION

W. Ait-Cheik-Bihi, A. Nait-Sidi-Moh, M. Wack

Systems and Transport Laboratory
UTBM

90010 Belfort cedex France

{wafaa.ait-cheik-bihi, ahmed.nait, maxime.wack}@utbm.fr

ABSTRACT: *Dioid algebra associated with Petri Nets are appropriate tools for modeling, analyzing, and evaluating the performance of Discrete Event Systems (DESs). These tools are restricted to model systems characterized by choice phenomena and conflicts. In this paper a routing policy to arbitrate and resolve the conflicts encountered while modeling DESs by dioid algebra, is proposed. This routing policy is illustrated by considering services interaction as a case study. Results are reported to show the effectiveness of the proposed policy.*

KEYWORDS: *Discrete Event Systems, Modeling, Performance evaluation, Petri Nets, Dioid algebra, Routing policy.*

1 INTRODUCTION

The study of discrete event systems (DESs) is the subject of several researches. The behavior of such systems is characterized by synchronization, parallelism, and concurrency phenomena. Several works have been realized in this field and several classes of DESs are studied. For example, manufacturing systems, transportation systems, web services composition, and telecommunication systems.

A DES can be modeled by many formalisms. Among them, Petri Nets (PNs) that are well suited for modeling and describing the behavior of these systems. The graphical representation of Petri nets, and the associated mathematical techniques, allow this tool to be one of the powerful formalisms used for modeling DESs. PNs allow to naturally describe a system and study its structural behavior, in order to evaluate its performances and deduct its qualitative properties. By analyzing a PN model, it is easy to verify certain properties, like boundedness, vivacity, absence of blocking, and reachability. A review of this graphical tool is given in (Murata, 1989) and (Diaz, 2001). However, despite the power and the capability of PN to model and evaluate DESs, certain performance and properties (like time management, optimization, prediction issues, etc.) can not be processed using this tool. In order to study DESs including these issues, a complementary tool is required for completing PN models, to make a perfect study. Hence, we will use dioid algebra, which is a useful mathematical tool, to translate the behavior of a PN model, particularly

Timed Event Graph (TEG), to a state representation. Dioid algebra has emerged as the suitable mathematical structure to model the phenomena of synchronization, assembly, concurrency, and parallelism. This tool has led to many accomplishments for performance evaluation and control of DESs (Heidergott et al., 2006). It is dedicated to the analysis of systems properties whose behavior can be represented by linear equations. Generally, this is not the case when using usual algebra due to the nonlinearity of certain operators. More details about this tropical algebra can be found in (Baccelli et al., 1992), (Cohen et al. 1989), and (Akian et al., 1990).

In this paper, PN and dioid algebra as complementary tool are used to specify, model, analyze, and evaluate the performance of interaction and interoperability in the services composition. The proposed scenario is considered as a DES wherein the events evolve in discrete state space. Furthermore, this scenario is characterized by choice phenomena, synchronization, and concurrency. To manage these phenomena, especially choice phenomena where conflicts appear, a routing policy is proposed in order to arbitrate and solve these conflicts. So, a routing function and its corresponding algorithm are defined. Finally, a case study of services interoperability is considered, and results are reported to show the effectiveness of the proposed routing policy.

The remainder of this paper is organized as follows. In section 2, we start by a related work of the used formalisms and their application domains. In sec-

tion 3, a brief overview of PNs and dioid algebra is given. Conflict management and routing policy are presented in section 4. Section 5 describes the case study considered to illustrate the proposal. Section 6 concludes the paper and presents some perspectives.

2 RELATED WORK

Several researchers are interested by modeling, analyzing, and evaluating the performance of DESs. Among the modeling tools used there are: Petri Nets (Alsaba et al., 2007), $(\max,+)$ algebra (Gaubert, 1992), Markov process (Wang et al., 1999), Network calculus (Bakhouya et al., 2009), fuzzy system theory (Hayat and Barret, 1997). Dynamic of DESs is managed by synchronization, parallelism, concurrency, and conflict phenomena. Variety of these phenomena make the study of such systems difficult, and therefore, require the use of several complementary theories for their description and analysis.

Modeling by PN and dioid algebra is used in different domains (Vries et al., 1998), (Nait-Sidi-Moh et al., 2009b), and (Boubour et al., 1997). For transportation systems, several works have been done. For example in (Nait-Sidi-Moh et al., 2009a), PN and spectral theory in $(\max,+)$ algebra are used for performance evaluation and optimization of a bus network. (Gaubert, 1992) deals with the control of a transportation system by using the technique of residuation in $(\max,+)$ algebra. Several works have been also realized in modeling and performance evaluation of manufacturing systems. In (Alsaba et al., 2007), these tools are used to model and control manufacturing systems. More precisely, the authors proposed a PN model to represent resource sharing by defining a scheduling rule. (Lahaye et al., 1999) use eigenvalues in dioid algebra associated with conflicted PN to determine the productivity of an assembly line in manufacturing system. (Hamadi and Benatallah, 2003) used PN to model and describe the services composition. In (Yi and Kochut, 2004), colored PNs are used to model the protocols of communication in web services composition.

It is worth noting that, these studies are done taking into consideration some simplifying hypothesis. For example, they are not focused on standard theories incorporating the choice phenomena and the conflict management. To address these issues, some few researches have been proposed (Bouillard et al., 2005) and (Nait-Sidi-Moh et al., 2006).

The purpose of this paper is to study and analyze systems characterized by choice phenomena, by proposing a routing policy to resolve and arbitrate the conflicts using PN and dioid algebra. To do so, we first present, in the next section, an overview of these tools.

3 PETRI NETS AND $(\max,+)$ ALGEBRA: AN OVERVIEW

3.1 Petri Nets

Modeling by PNs provides a natural and precise formal presentation, an intuitive graphical language, and a solid mathematical basis for verifying properties and analysis for modeled systems. Petri nets have been shown to offer significant advantages in modeling DESs. Specially, this tool is useful in verification, validation and performance evaluation.

Different classes of PNs have been developed in the literature to answer for the best the modeling of various systems. Each class is developed to model a specific system (Diaz, 2001). For example, Timed Event Graphs (TEG) are used to describe the behavior of DESs. This class is easy to be implemented, it allows to describe systems characterized by synchronization and parallelism, but not those described with choice phenomena. To address these issues, another class of PNs, which is the Free Choice PN can be used. This class is employed when the choice criteria are imposed. The choice can be done according to some criteria. For example, the priority, the optimization, or the schedulability. The major problem of this class is to know which choice will be done. Unfortunately, the behavior of such models can not be easily expressed using $(\max,+)$ equations. So, the development of a standard theory in dioid algebra, to study systems with conflict, is required. To do so, some routing policies are proposed in the literature, such as Bernoulli routing, race policy and periodic routing (Bouillard and al., 2005).

In the following, two sub-classes of PN used in this paper are introduced.

3.1.1 Free-Choice Petri Net (FCPN)

FCPN is a graphical tool, dedicated to describe DES with conflicts (Diaz, 2001). In a FCPN model, there is at least one place, which has at least two downstream transitions as described in Figure 1.

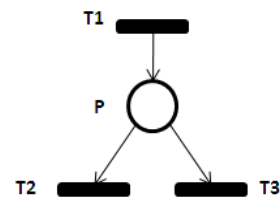


Figure 1: Example of FCPN

In a formal way, a FCPN is a 5-uplet $\mathbf{R} = (P, P_c, T, A, M_0)$ with:

- P is a finite set of places,

- ▶ $P_c \subset \mathbf{P}$ is a finite set of conflicted places,
- ▶ T is a finite set of transitions,
- ▶ $A \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs,
- ▶ $M_0 : P \rightarrow \mathbb{N}$ is the initial marking (token number in places at the initial state).

The conflict is denoted as $\langle P_{c_i}, x_1, \dots, x_n \rangle$. We say that P_{c_i} is a conflicted place and x_1, \dots, x_n are its downstream transitions, called also conflicted transitions.

3.1.2 P-Timed Petri Net (TPN)

TPNs allow to describe systems whose functioning depends on time. In other words, in TPNs the firing of a transition occurs after the end of a given time (equals to τ_i defined in equation (1)), associated to its upstream place. In TPN model, a time (called temporization) is associated with each place as defined as follows:

$$L : P \rightarrow \mathbb{R}^+ \quad (1)$$

$$P_i \rightarrow L(P_i) = \tau_i$$

We denote by $TPN = \{P, T, A, M_0, L\}$, the formal expression of the P-Timed PN.

A P-Timed Free-Choice Petri Net (TCPN), defined formally by $\{P, P_c, T, A, M_0, L\}$, merges the two sub-classes defined previously.

We denote also by T_{Conf} , the set of transitions which are in a conflict situation, such as: $T_{Conf} = \{T_i \in T / T_i \in P_{c_j}^\circ\}$, where $P_{c_j}^\circ$ is the set of all downstream transitions of the conflicted place P_{c_j} .

3.2 Dioid algebra

In our study, we are interested in $(\max, +)$ algebra dedicated to the management and calculus of the occurrence dates of events. In the case we are interested in the occurrence number of events, another class of dioid algebra, which is $(\min, +)$, may be applied (Lahaye and al., 1999). It is proved in several studies (Cohen et al., 1989), (Heidergott et al., 2006), and (Baccelli et al., 1992) that this algebra is efficient and powerful for studying and analyzing DESs behavior.

3.2.1 Definition

A dioid \mathbb{D} is a set endowed with the two laws noted \oplus and \otimes called respectively "addition" and "multiplication" in the sense of dioid algebra, such as :

$\forall a, b, c \in \mathbb{D}$, we have :

\oplus is associative : $(a \oplus b) \oplus c = a \oplus (b \oplus c)$;

\oplus is commutative : $a \oplus b = b \oplus a$;

\otimes admits a neutral element noted as ε : $a \otimes \varepsilon = a$;

\otimes is associative: $(a \otimes b) \otimes c = a \otimes (b \otimes c)$;

\otimes admits a neutral element noted as e : $a \otimes e = a$;

\otimes is distributive over \oplus : $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$;

ε is absorbing for \otimes : $\varepsilon \otimes a = a \otimes \varepsilon = \varepsilon$;

\oplus is idempotent : $a \oplus a = a$.

A dioid is said commutative if the multiplication is commutative.

4 CONFLICT RESOLUTION IN $(\max, +)$ ALGEBRA

To model and analyze conflicted systems, dioid algebra is limited and there is no standard theory for such systems. In this perspective we introduce a routing policy, which allows to arbitrate and resolve conflicts. The objective is to develop a generic method for conflicted systems. To do so, a routing function is defined taking into account some choice criteria.

First, let us give a TCPN model, which will be used to define the routing policy.

4.1 Modeling using TCPN

The Figure 2 describes a TCPN model. In this graph, three transitions are in conflict situation $T_{Conf} = \{x_1, x_2, x_3\}$. So here, we limit ourselves to three conflicted transitions, but the proposed approach may be applied to more than three ones.

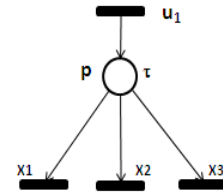


Figure 2: A P-Timed Petri Net with Conflict

When the place p contains one token, we don't know which transition will be fired, because there are a free choice in this place. So, the transition that will be fired can be chosen arbitrarily. This policy could not answer our requirements when some choice criteria are imposed, like optimization purpose or satisfying quality of service. The objective is to analyze and evaluate the performance of conflicted systems, tackling into account the constraints of firing of conflicted transitions, which will be integrated in the mathematical model. To do so, we start by defining some choice criteria. A choice can be done according to the availability of services or resources. The available service is chosen by firing its corresponding transition. For example in the context of services interaction, the choice is done according to the availability criteria. It can be also done according to the optimized service

(cost, response time, QoS, etc.). We have to define a routing policy to choose how the conflicted transitions may be fired, in which order, and according to which criteria. In the next section, a (max,+) model representing the behavior of TCPN model of Figure 2 is given. Furthermore, the suggested routing policy to manage the conflicts is introduced.

4.2 Modeling using (max,+) algebra

Before giving the (max,+) model, let us define the firing policy of conflicted transitions. This policy is based on virtual firing using what we call "virtual token" defined hereafter.

4.2.1 Virtual token

The unique token in conflicted place participates to the real firing of the fired transition, others transitions are not really fired but we assume that they are virtually fired. The introduction of these virtual firings facilitates the description of the behavior of conflicted system with (max,+) linear representations. To represent these virtual firings, we define the following function:

$$f : T \rightarrow \{\epsilon, e\} \quad (2)$$

$$x_i \rightarrow f(x_i) \quad \text{with } i \in \{1, 2, \dots, |T|\}$$

Where $|T|$ is the number of all transitions of the PN model (called also cardinal of T).

To simplify the equation (2), we note $f_{x_i} = f(x_i)$ in what follows.

$$\forall P_l \in \mathbf{P}_c, \forall x_i \in P_l^\circ : \bigoplus_{i=1}^{|P_l^\circ|} f_{x_i} = e \quad (3)$$

In what follows, we will use another parameter called "dater". The associated *dater* with the transition x_i is $x_i(k)$ and represents the date of the k^{th} firing of x_i . We define then the following application:

$$x_i : \mathbb{N}^* \rightarrow \mathbb{R}_{max} \quad (4)$$

$$k \rightarrow x_i(k) \quad \text{with } i \in \{1, 2, \dots, |T|\}$$

When only a transition x_i is really fired for the k^{th} time:

$$f(x_i)(k) = f_{x_i}(k) = e$$

In a formal way, let $P_l \in \mathbf{P}_c$ be a conflicted place, for a given $k \in \mathbb{N}^*$:

$$\begin{cases} \forall x_i \in P_l^\circ, \exists! x_j \in P_l^\circ; & x_j(k) = \alpha \quad (\alpha \in \mathbb{R}^+) \\ \forall x_i \in P_l^\circ \setminus \{x_j\}; & x_i(k) = \epsilon \end{cases}$$

Where x_j is really fired, and x_i is virtually fired. Knowing that the dater is monotone, this virtual token verify this property even if the place P_l contains more than one token.

In Figure 2, the downstream transitions of p are fired according to a priority order. For example, x_1 will be really fired when $k=1$ (u_1 is fired for the first time), at the same time x_2 and x_3 are virtually fired. When the second token arrives at p ($k=2$), x_2 will be really fired and the other transitions will be virtually fired, and so on. This schedulability allows to arbitrate the conflict, without being limited to the arbitrarily firing. Our concern now is to generate a (max,+) linear equations to manage and solve the conflicts. So, formally, the proposed routing function is expressed as follows:

$$\forall P_l \in \mathbf{P}_c, \forall x_i \in P_l^\circ \text{ with } 1 \leq i \leq |P_l^\circ|, \forall k \geq 1;$$

$$f_{x_i}(k) = \begin{cases} e & \text{if } i \equiv k \text{ mod } (|P_l^\circ|) \\ \epsilon & \text{otherwise} \end{cases} \quad (5)$$

with $|P_l^\circ|$ is the cardinal of the set P_l° . For the model described in Figure 2, we have only one conflicted place p, where:

- $|p^\circ| = 3$
- $T_{Conf} = \{x_1, x_2, x_3\}$

4.2.2 (Max,+) linear representation

The analytical behavior of the graphical model presented in Figure 2 is given as follows: $\forall k \geq 1$:

$$\begin{cases} x_1(k) = \tau \otimes u_1(k) \otimes f_{x_1}(k) \\ x_2(k) = \tau \otimes u_1(k) \otimes f_{x_2}(k) \\ x_3(k) = \tau \otimes u_1(k) \otimes f_{x_3}(k) \end{cases} \quad (6)$$

The system (6) can be written using matrix form by:

$$X(k) = F(k) \otimes B \otimes U(k) \quad (7)$$

In a general manner, the system (7) can be written for a complete model, where input variables, state variables, and output variables are defined, by the following system:

$$\begin{cases} X(k) = A \otimes F(k) \otimes X(k) \oplus F(k) \otimes B \otimes U(k) \\ Y(k) = C \otimes X(k) \end{cases} \quad (8)$$

With :

$$\begin{cases} U(k) : \text{Input vector} \\ X(k) : \text{State vector} \\ Y(k) : \text{Output vector} \\ F(k) : \text{Routing matrix} \end{cases} \quad (9)$$

$A \in \mathcal{M}_{n \times n}(\mathbb{R}_{max})$, $B \in \mathcal{M}_{n \times m}(\mathbb{R}_{max})$ and $C \in \mathcal{M}_{q \times n}(\mathbb{R}_{max})$ are characteristic matrices with: n (respectively m and q) is the number of internal (respectively input and output) transitions of TCPN model. The routing matrix $F(k) \in \mathcal{M}_{n \times n}(\mathbb{R}_{max})$ allows to arbitrate the conflict whatever $k \in \mathbb{N}^*$. It is defined as follows:

$$\mathbf{F}(k) = \begin{pmatrix} f_{x_1}(k) & \epsilon & \epsilon \\ \epsilon & f_{x_2}(k) & \epsilon \\ \epsilon & \epsilon & f_{x_3}(k) \end{pmatrix} \quad (10)$$

The components of the matrices A , B , C are expressed according to the system data. F is expressed in terms of k , and allows to determine the transition which will be really fired for each time. These firing are explained in the equation (5). After defining the state equation (8), we will proceed to its resolution in the next paragraph.

4.2.3 Resolution of (max, +) model

In this section, we determine all the states of the system at any time, by resolving the first equation of the system (8). In order to resolve this equation, we replace in an iterative way $X(k)$ by its expression.

By verifying the corresponding precedence graph of the matrix A , $\mathcal{G}(A)$, which is strongly connected, we can write: $\forall i \geq n, A^i = \epsilon$. The star of Kleene is expressed by $A^* = Id \oplus A \oplus \dots \oplus A^{n-1}$.

The first equation of (8) can then expressed as:

$\forall k \geq 1$:

$$\begin{aligned} X(k) &= A \otimes F(k) \otimes X(k) \oplus F(k) \otimes B \otimes U(k) \\ &= A \otimes F(k) \otimes [A \otimes F(k) \otimes X(k) \oplus F(k) \otimes B \otimes U(k)] \oplus F(k) \otimes B \otimes U(k) \\ &= \dots \\ &= [A \otimes F(k)]^n \otimes X(k) \oplus [(A \otimes F(k))^{(n-1)} \oplus (A \otimes F(k))^{(n-2)} \oplus \dots \oplus I] \otimes F(k) \otimes B \otimes U(k) \end{aligned}$$

In order to simplify this written expression, we take $A' = A \otimes F(k)$, we obtain:

$$\begin{aligned} X(k) &= A'^n \otimes X(k) \oplus [I \oplus A' \oplus \dots \oplus A'^{(n-1)}] \\ &\quad \otimes F(k) \otimes B \otimes U(k) \end{aligned} \quad (11)$$

While using star of Kleene, The equation (11) becomes:

$$X(k) = A'^* \otimes F(k) \otimes B \otimes U(k) \quad (12)$$

4.2.4 Routing policy algorithm

The algorithm defined hereafter allows to solve the equation (7), by defining for each k , the transitions that will be really and virtually fired. This algorithm can be extended to general model represented by the equation (8). The complexity of this algorithm is $O(N \times |P| \times |T|)$ (Figure 2) where N is the firing number of the input transition u .

Algorithm 1 Routing Policy Algorithm

Require:

- 1: $u(k) \in \mathbb{R}^+, \forall k \in 1, \dots, N \subset \mathbb{N}^*, N \in \mathbb{N}^*$
- 2: $\tau \in \mathbb{Q}^+$

Ensure:

- 3: BEGIN
 - 4: **for** $k = 1$ **to** N **do**
 - 5: **for** $P_l \in \mathbf{P}$ **do**
 - 6: **if** $|P_l^\circ| < 2$ **then**
 - 7: $f_{P_l^\circ} = e$
 - 8: **else**
 - 9: **for** $j = 0$ **to** $|P_l^\circ| - 1$ **do**
 - 10: **if** $j \equiv k \otimes \text{mod}(|P_l^\circ|)$ **then**
 - 11: $f_{x_{j+1}}(k) = e$
 - 12: **else**
 - 13: $f_{x_{j+1}}(k) = \epsilon$
 - 14: **end if**
 - 15: $x_{j+1}(k) = \tau \otimes u(k) \otimes f_{x_{j+1}}(k)$
 - 16: **end for**
 - 17: **end if**
 - 18: **end for**
 - 19: **end for**
 - 20: END
-

All non-conflicted transitions will be really fired if the upstream places contains one token (line 6 and 7 of the algorithm). To illustrate our approach, we apply in the next section the proposed methodology to model services interoperability as a case study.

5 CASE STUDY

The interaction of services within a system can be made in a state discrete space, we can talk about a class of DES. So, it is possible to apply the developed theory, in the literature, for studying DESs to model, analyze and evaluate the process of services interaction. The evolution of this process is determined by the presence of certain events such as synchronization, parallelism, concurrency, choice and conflicts.

5.1 Studied system

The studied scenario is defined in Figure 3, it is composed of four services S_1, S_2, S_3 and S_4 . Each service represents an independent entity, it can represent for example, a web service, a service in a manufacturing chain, etc. Figure 3 represents the TCPN model of the considered scenario. The interaction between

services is described as follows: the service S_1 send its request to the three other services. This request is modeled by the firing of the transition u_1 and the added token in the place P_1 models the request. Once the request is sent, S_2, S_3 and S_4 are able to answer the request, but only one and only one can give the answer hence the conflict. This means that the three transitions x_1, x_2 and x_3 are enabled but only one can be fired by the only token of the place P_1 . Our proposal allows then to determine which service is selected to answer the request. This selection is based on the proposed routing policy, and the three services S_2, S_3 and S_4 are invoked according to a defined order as given in equation (5). This is modeled by the sequential firing of transition x_1, x_2 and x_3 to answer the different requests coming from S_1 (see Figure 3). The firing of x_4 (resp. x_5 and x_6) represents the processing end of the request by the service S_2 (resp. S_3 and S_4). Finally, the token added to the place P_5 models the answer of the request. To study and eval-

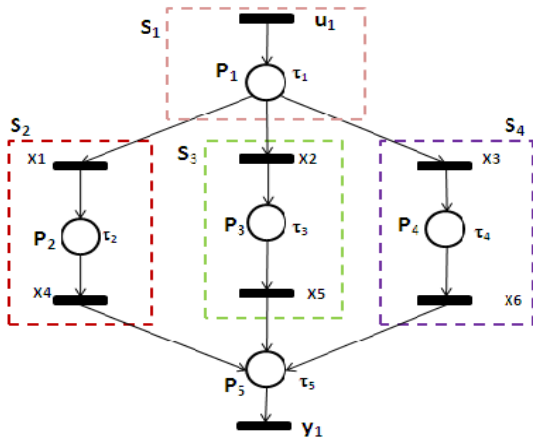


Figure 3: Services composition scenario

uate the performance of the considered scenario, we use the results obtained in the previous section. The firing of transitions is made according to the routing function defined in equation (5). The analytical behavior of the scenario presented in Figure 3 is: $\forall k \geq 1$:

$$\begin{cases} x_1(k) = \tau_1 \otimes u_1(k) \otimes f_{x_1}(k) \\ x_2(k) = \tau_1 \otimes u_1(k) \otimes f_{x_2}(k) \\ x_3(k) = \tau_1 \otimes u_1(k) \otimes f_{x_3}(k) \\ x_4(k) = \tau_2 \otimes x_1(k) \otimes f_{x_4}(k) \\ x_5(k) = \tau_3 \otimes x_2(k) \otimes f_{x_5}(k) \\ x_6(k) = \tau_4 \otimes x_3(k) \otimes f_{x_6}(k) \\ y_1(k) = \tau_5 \otimes (x_4(k) \oplus x_5(k) \oplus x_6(k)) \end{cases} \quad (13)$$

From the TCPN model defined in Figure 3, when the transition x_1 is really fired for the k^{th} time, then x_4 will be also really fired for the k^{th} time. It is the same for (x_2, x_5) and (x_3, x_6) . From this remark, we can obtain:

$$\forall i \in \{4, 5, 6\}, \forall k \in \mathbb{N}^* : f_{x_i}(k) = f_{x_{i-3}}(k); \quad (14)$$

The system (13) can be written like the system (8) by:

$$\forall k \geq 1;$$

$$\begin{cases} X(k) = A \otimes F(k) \otimes X(k) \oplus F(k) \otimes B \otimes U(k) \\ Y(k) = C \otimes X(k) \end{cases} \quad (15)$$

With :

$$\begin{cases} U(k) = u_1(k) \text{ Input variable} \\ X(k) = [x_1(k), x_2(k), \dots, x_6(k)]^t \text{ State vector} \\ Y(k) = y_1(k) \text{ Output variable} \end{cases}$$

The solution of the system (15) is given by equation (12). The second equation of the system (15), can be expressed by:

$$\forall k \geq 1;$$

$$Y(k) = C \otimes A'^* \otimes F(k) \otimes B \otimes U(k) \quad (16)$$

With $A \in \mathbb{R}_{max}^{6 \times 6}$, $B \in \mathbb{R}_{max}^{6 \times 1}$, $C \in \mathbb{R}_{max}^{1 \times 6}$, $F \in \mathbb{R}_{max}^{6 \times 6}$ given hereafter.

5.2 Performance evaluation

The firing of the transition u_1 follows an uniform law, for example it is fired every 10 min, thus :

$$\forall k > 1;$$

$$\begin{cases} u_1(1) = 10 \\ u_1(k) = 10 \otimes u_1(k-1) \end{cases}$$

All model temporizations are given by:

Temporizations	τ_1	τ_2	τ_3	τ_4	τ_5
Numerical values	3	8	12	6	1

Table 1: Temporizations values

According to these values, we give the expressed data of the characteristics matrices and vectors as follow:

$$\mathbf{A} = \begin{pmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 8 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 12 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 6 & \epsilon & \epsilon & \epsilon \end{pmatrix} \quad (17)$$

$$\mathbf{B} = (3 \ 3 \ 3 \ \epsilon \ \epsilon \ \epsilon)^t \quad (18)$$

$$\mathbf{C} = (\epsilon \ \epsilon \ \epsilon \ 1 \ 1 \ 1) \quad (19)$$

By applying the proposed algorithm for $1 \leq k \leq \mathbf{N} = N_{max} = 20$, $F(k)$ can be expressed as follows:

$$F(1) = \begin{pmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{pmatrix} \quad (20)$$

$$F(2) = \begin{pmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{pmatrix} \quad (21)$$

$$F(3) = \begin{pmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{pmatrix} \quad (22)$$

Since we have for all $k > 3 : F(k) = F(k - 3)$, we give only $F(1)$, $F(2)$ and $F(3)$ (see equation (14)). $A' = A \otimes F(k)$ will be computed according to $F(1)$, $F(2)$ and $F(3)$. The obtained results hereafter are expressed by resolving the equation (15) and by applying the proposed algorithm till $k = N_{max} = 20$, and using the system data.

k	u	x_1	x_2	x_3	x_4	x_5	x_6	y
1	10	13	ϵ	ϵ	21	ϵ	ϵ	22
2	20	ϵ	23	ϵ	ϵ	35	ϵ	36
3	30	ϵ	ϵ	33	ϵ	ϵ	39	40
4	40	43	ϵ	ϵ	51	ϵ	ϵ	52
5	50	ϵ	53	ϵ	ϵ	65	ϵ	66
6	60	ϵ	ϵ	63	ϵ	ϵ	69	70
7	70	73	ϵ	ϵ	81	ϵ	ϵ	82
8	80	ϵ	83	ϵ	ϵ	95	ϵ	96
9	90	ϵ	ϵ	93	ϵ	ϵ	99	100
10	100	103	ϵ	ϵ	111	ϵ	ϵ	112
11	110	ϵ	113	ϵ	ϵ	125	ϵ	126
12	120	ϵ	ϵ	123	ϵ	ϵ	129	130
13	130	133	ϵ	ϵ	141	ϵ	ϵ	142
14	140	ϵ	143	ϵ	ϵ	155	ϵ	156
15	150	ϵ	ϵ	153	ϵ	ϵ	159	160
16	160	163	ϵ	ϵ	171	ϵ	ϵ	172
17	170	ϵ	173	ϵ	ϵ	185	ϵ	186
18	180	ϵ	ϵ	183	ϵ	ϵ	189	190
19	190	193	ϵ	ϵ	201	ϵ	ϵ	202
20	200	ϵ	203	ϵ	ϵ	215	ϵ	216

Table 2: Numerical results obtained by solving equation (15).

In the table (2), we have, for example, when $k=1$ the transition x_1 and x_4 are really fired but the others internal transitions are virtually fired. The date of each virtual firing equals to ϵ , and the date of each real firing is expressed by a nonnegative integer (see table (2)). For example, for $k = 15$ only the transitions x_3 and x_6 are really fired.

In order to represent only the real dates of different firing of all transitions, we removed all virtual dates, which are given by ϵ , from the result table (2). The table (3) represents then these real firing dates.

k	u	x_1	x_2	x_3	x_4	x_5	x_6	y
1	10	13	23	33	21	35	39	22
2	20	43	53	63	51	65	69	36
3	30	73	83	93	81	95	99	40
4	40	103	113	123	111	125	129	52
5	50	133	143	153	141	155	159	66
6	60	163	173	183	171	185	189	70
7	70	193	203	-	201	215	-	82
8	80	-	-	-	-	-	-	96
9	90	-	-	-	-	-	-	100
10	100	-	-	-	-	-	-	112
11	110	-	-	-	-	-	-	126
12	120	-	-	-	-	-	-	130
13	130	-	-	-	-	-	-	142
14	140	-	-	-	-	-	-	156
15	150	-	-	-	-	-	-	160
16	160	-	-	-	-	-	-	172
17	170	-	-	-	-	-	-	186
18	180	-	-	-	-	-	-	190
19	190	-	-	-	-	-	-	202
20	200	-	-	-	-	-	-	216

Table 3: Dates of real firing of model transitions without taking into account virtual firings.

In the table (3), we remark that for the 20th firing of u_1 , x_1, x_2, x_4 and x_5 are fired seven times, and x_3 and x_6 are fired six times, the output of the system is fired twenty times.

6 CONCLUSION AND PERSPECTIVES

Despite the relevance, the importance, and the power of Petri Nets and dioid algebra jointly for modeling, analyzing, and evaluating the performance of DESs, few researches have been dedicated to the use of these formalisms for studying DESs with conflicts.

The proposed approach focuses on the development of a theory and a generic method to model and analyze system with conflicts in $(\max,+)$ algebra. This methodology allows to solve and arbitrate the conflicts encountered. A description of a DES with conflict by $(\max,+)$ linear equations is proposed. The obtained equations are easy to handle and solve thanks to the defined routing policy. To illustrate the pro-

posed approach and test the defined routing policy, modeling and evaluating study of services interaction is considered.

Future works can include the introduction of more significant criteria for the management of conflicts. Other routing policies will be defined in a general manner. The generated (max,+) system will consider and integrate all criteria and the proposed routing policies.

ACKNOWLEDGMENTS

This work is supported by the ASSET project (Advanced Safety and Driver Support for Essential Road Transport, 2008-2011) of the EU 7th Framework Program (FP7) (Asset, 2008).

REFERENCES

- Akian, M., G. Cohen, S. Gaubert, R. Nikoukhah and J.P. Quadrat, 1990. Linear systems in (max, +) algebra. *Proceedings of the 29th Conference on Decision and Control, Honolulu*.
- Alsaba, M., J.L. Boimond and S. Lahaye, 2007. Sur la commande des systèmes flexibles de production manufacturière par l'algèbre des dioïdes. *Revue e-STA, Sciences et Technologies de l'Automatique, vol. 4, n° 2*.
- Asset, 2008-2011. The European ASSET project. <http://www.project-asset.com>.
- Boubour, R., C. Jard, A. Aghasaryan, E. Fabre and A. Benveniste, 1997. Petri net approach to fault detection and diagnosis in distributed systems: application to telecommunication networks, motivations, and modeling. *In Proceeding of the 36th IEEE Conference on Decision and Control, San Diego, CA*.
- Baccelli, F., G. Cohen, G. Olsder and J. Quadrat, 1992. Synchronization and linearity: an algebra for discrete event systems. *J. Wiley, New York*.
- Bakhouya, M., S. Subah, J. Gaber and T. El-Ghazaoui, 2009. Analytical modeling and Evaluation of on-chip Interconnects using Network calculus. *The 3rd ACM/IEEE International Symposium on Networks-On-Chip (NOC09), 74-79*.
- Bouillard, A., B. Gaujal and J. Mairesse, 2005. Throughput in stochastic free-choice nets under various policies. *European Control Conference. CDC-ECC '05. 44th IEEE Conference*.
- Cohen, G., P Moller, J.P. Quadrat, LC Inria and FM Viot, 1989. Algebraic tools for the Performance Evaluation of discrete event systems. *IEEE Proceedings: Special issue on Discrete Event Systems*.
- Diaz, M., 2001. Les réseaux de Petri - Modèles fondamentaux. *Hermès science publications*.
- Gaubert, S., 1992. Systèmes Dynamiques à événements discrets. *Lecture notes, ENSMP, Automation and Master degree ATS Orsay. Inria, Rocquencourt, France*.
- Hamadi, R. and B. Benatallah , 2003. A Petri Net-based Model for Web Service Composition. *presented at Database Technologies 2003, 14th Australian Database Conference*.
- Hayat S. and C. Barret, 1997. Introduction du critère Flux de Passagers dans la Régulation de Trafic de Lignes de Métros basée sur des Contrôleurs Flous. *INRETS-ESTAS. Rapport de recherche, septembre 1997-40*.
- Heidergott, B., G.J. Olsder and J.V.D. Woude , 2006. Max Plus at work - Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications. *Princeton University Press*.
- Lahaye, S., Boimond J.L. and Hardouin L. ,1999. GET and Ajout/Retrait dynamique de jetons comportement asymptotique dans l'algèbre (min,+). *In proceedings of Modélisation des système réactifs (MSR'99), Cachan, France, p. 27-37*.
- Murata, T., 1989. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE, volume 77(4), p. 541-580*.
- Nait-Sidi-Moh, A., M.A. Manier, A.E. Moudni, M. Wack, 2006. Petri net with conflict and (max, plus) algebra for transportation systems. *In proceedings of the 11th IFAC symposium on control in transportation systems, Delft, The Netherlands*.
- Nait-Sidi-Moh, A., M.-A. Manier and A. El Moudni, 2009a. Spectral analysis for performance evaluation in a bus network. *In european journal of operational research, vol. 193, Issue 1, p. 289-302*.
- Nait-Sidi-Moh, A., W. Ait-Cheik-Bihi and M. Wack, 2009b. Modeling and Analysis of a Non-Synchronized Transport network using Petri Nets and (max,+) algebra. *In proceedings of 39th International Conference on Computers and Industrial Engineering (CIE39), Troyes, France*
- Olsder, G.-J., S. Subiono, M.-M. Gettrick, 1998. On large scale max-plus algebra model in railway systems. *In: 26^{ème} école de printemps d'informatique, Algèbre (Max,+) et applications en informatique, Ile de Noirmoutier, France, p. 177-192*.

- Vries, D.R., B. De Schutter and B. De Moor, 1998. On max-algebraic models for transportation networks. *Proceedings of the 4th International Workshop on Discrete Event Systems (WODES'98), Cagliari, Italy, p. 457-462.*
- Wang, J., C. Jin and Y. Deng, 1999. Performance Analysis of Traffic Network Based on Stochastic timed Petri Net Models. p. 77-85 IEEE.
- Yi, X. and K.J. Kochut, 2004. Process Composition of Web Services with Complex Conversation Protocols : a Colored Petri Nets Based Approach. *In Proceedings of the Design, Analysis, and Simulation of Distributed Systems Symposium, pp. 141-148, Arlington, VA, USA.*