A Discrete Time Control Synthesis Tool for Time Discrete Event Systems

Alexandre SAVA*, Zied ACHOUR*, Nidhal REZG*

Ioana LEAHU**

*INRIA Lorraine-MASCI Team / Laboratoire de Génie Industriel et Production Mécanique

ENIM, Ile du Saulcy, 57045 Metz, France

sava@enim.fr, {Zied.Achour, Rezg}@loria.fr,

**University "Al. I. Cuza", Str General Berthlot, nr 16, Iasi, Roumanie

olgai@infoiasi.ro

Abstract: This paper presents a control synthesis software tool for manufacturing systems with a driven dynamic event. This tool is based on an original control synthesis algorithm which solves the forbidden state problem (FSP) under the following hypothesis: 1) each event may occur at time instants specified by an interval or a union of intervals; 2) two types of events are considered: controllable and incontrollable events and 3) the evolution of time is discrete. This tool generates a live and maximally permissive controller which can 1) act on the arriving date of controllable events or 2) completely forbid some controllable events. The aim is to guaranty the respect of the given specifications.

Key words: Forbidden state problem, Petri Nets, control synthesis, time constraints.

Alexandre SAVA is Associate Professor at École Nationale d'Ingénieurs de Metz since september 2002. He obtained his Ph.D. degree in control systems and industrial engineering from the Institut National Polytechnique de Grenoble, in 2001. His research activity deals with control synthesis for timed discrete event systems and simulation based optimisation applied to production distribution systems.

Zied ACHOUR received his M.S. degree in Automatic production from the University Henri-Poincaré of Nancy-France in 2002 and his Ph.D. degree from the University Paul Verlaine-Metz, in 2005. Since September 2006 he is Associated Professor at University Paul Verlaine-Metz. His research activity deals with control synthesis of discrete dynamic event systems.

Nidhal REZG is a Full Professor at University Paul Verlaine-Metz since September 2004. He received the Ph.D. degree in automatic control from the Institut National des Sciences Appliquées de Lyon and the Habilitation à Diriger des Recherches degree from University of Metz, France, in 1996 and 2003, respectively. Between 1997 and 1999 he was Professor in the Industrial Engineering Department at Moncton University, New Brunswick, Canada. His research interests include control synthesis of discrete dynamic event systems, the reliability and maintenance for performance evaluation simulation and optimisation of manufacturing systems. He was Guest Editor of a special issue of International Journal of Production Research in 2004.

Ioana LEAHU received her M.Sc. Degree in Distributed Systems from the "Al. I. Cuza" University of Iasi, Faculty of Computer Science in 2003 and is currently a PhD student at the Faculty of Computer Science in Iasi. Between 2003 and 2006 she was Research Assistant at the Faculty of Computer Science in Iasi. Since October 2006 she is Assistant Professor at the same faculty, in the Computer Science Fundamentals and Distributed Systems Department. Her research activity deals with behavioral properties of distributed systems and control synthesis for discrete event systems.

1. Introduction

The theory of Discrete Event Systems (DES) supervision was initiated by the research work of Ramadge and Wonham [9][10]. This theory has triggered interest in the supervisory control and many approaches have been proposed to design maximally permissive controllers, using in particularly the Petri Net model [8].

Thus, Yamalidou and al. [14], proposed a maximally permissive controller synthesis method based on Petri nets and place invariants. This approach deals with totally controllable and observable plants. In [4] and [11], the authors take into account the uncontrollable nature of some events. The authors use the theory of regions in order to design a maximal permissive live controller made of a set of control places which may disable the firing of some controllable transitions. This approach has been extended to DES modeled by Petri nets with uncontrollable/ unobservable transitions [1].

Taking into account the time information is crucial for realistic systems. This information can be used to compute more permissive control laws [12]. Timing introduces a new dimension of considerable interest in DES control, but also of significant complexity.

A first attempt to take into account the influence of time in the process model is presented in [3]. The authors extend the theory of Ramadge and Wonham by introducing forcible events and time constraints. Generally, discrete time approaches are characterized by a combinatorial explosion of the state space. Meanwhile, the control synthesis methodology, usually inspired by the Kumar algorithm [6], is rather easy to implement.

Some authors proposed approaches based on continuous time models [2][7][13][15]. These approaches are mainly based on Time Petri Nets and Timed Automata models. They are characterized by a compact state space. However, the control synthesis is based on polyhedral operations and it is difficult to implement.

In this paper we present a control synthesis tool for time discrete event systems (TDES). This tool is based on an original approach which solves the forbidden state problem (FSP) under the following hypothesis: 1) each event may occur at time instants specified by an interval or a union of intervals; 2) two types of events are considered: controllable and incontrollable events and 3) the evolution of time is discrete. This control synthesis technique is based on Time Controlled Petri Net model (TCPN), which is derived from the Controlled Petri Net model [5] by including a new time firing condition. Therefore, each transition of the TCPN is characterized by a firing condition made of : 1) a time constraint and 2) a logic constraint. An enabled transition can be fired if both time constraint and logic constraint are satisfied.

The proposed control synthesis approach consists in building the state space of the plant behavior under control by computing new time constraints and new logic constraints such that the forbidden states are no longer reachable and the controlled plant is deadlock free. The aim is to build a most permissive controller which guarantees the respect of the given specifications under the liveness constraint. A controller is said to be maximal permissive if it has the ability to restrict the behavior of the plant so that only the forbidden states are avoided. In our case, the maximal permissivity is measured by the number of admissible states under control. Liveness is the property of the controller to avoid deadlocks.

The rest of the paper is organized as follows. Section 2 introduces the control synthesis approach. In Section 3 we present the software tool that we developed. Section 4 discusses the complexity of the algorithm. Some conclusions and further directions are given in Section 5.

2. Control Synthesis Approach

Let us consider a plant, and a set of control specifications. Solving a control synthesis problem consists in building a controller so that the behavior of the closed loop system obtained by coupling the controller to the plant respects the given specifications.

In this paper we present a formal synthesis method of a time controller based on the TCPN model of a plant. The control specifications are given by General Mutual Exclusion Constraints (GMEC) [4]. Each GMEC gives a condition on the markings of TCPN places that has to be respected.

Definition 1: A TCPN model is a tuple N=(P, T, Pre, Post, M0, Is, C):

- P is a finite set of places;
- T is a finite set of transitions, $P \cap T = \phi$;
- Pre : $PxT \rightarrow \mathcal{N}$, is the pre-incidence function, which defines the weight of arcs from places to transitions, where $\mathcal{N}$ denotes the set of natural numbers;
- Post : $TxP \rightarrow \mathcal{N}$, is the post-incidence function, which defines the weight of arcs from transitions to places;
- M_0, is the initial marking;
- C is the set of control variables. A control variable C_j is associated to each controllable transition T_j;
- U(Cj) : $C \rightarrow Z$, defines a control function which associates a control value to each control variable Cj, where Z is the set of integer numbers;
- $I_s : T \rightarrow Q^+ x Q,^+$ is a function which associates a time interval to each transition, where Q denotes the set of positive rational numbers.

$$I_s(T_i)=[a_i, b_i]$$

$$a_i \le b_i\ ;\ 0 \le a_i < \infty\ ;\ 0 \le b_i < \infty$$

where ai and bi are positive rational numbers. ■

The set of input (resp. output) transitions of a place Pi is denoted by •Pi (resp. Pi•). The set of input (resp.

output) places of a transition Tj is denoted by •Tj (resp. Tj•). A Petri net can also be represented by the incidence matrix W defined as W(p, t)=Post(Pi,Tj)-Pre(Pi,Tj). A marking of the Petri net, $M:P\rightarrow\mathcal{N}$, is a mapping that assigns to each place a non negative integer number of tokens. A transition Tj is enabled by a marking M if and only if M(Pi)≥ Pre(Pi, Tj), ∀ Pi∈•Tj. A enabled transition can be fired if both the associated time condition and the logic condition are satisfied. Firing a transition Tj yields a new marking M' such that M'=M+W(., Tj). The time condition Is(Tj)=[aj, bj] is satisfied if the amount of time elapsed since the last enabling moment of Tj belongs to the interval [aj, bj]. The logic condition of Tj is satisfied if the control function value U(Cj) ≥1.

Let us consider the TCPN illustrated in figure 1. The transitions T1, T3, T4 and T6 are controllable. Therefore, we associate a control variable Cj to each transition Tj, j∈{1, 3, 4, 6}. Initially only the places P1 and P2 are marked. Therefore, the enabled transitions are T1 and T4. For instance, T1 can be fired if: 1) its firing is allowed by the controller and 2) an amount of time between 0 and 2 time units has elapsed since its last enabling moment. An example of GMEC is : M(P2)+M(P5)<2. Each marking that does not satisfy this constraint is considered to be forbidden. The behavior of a TCPN is modeled by a Time Reachability Graph (TRG).

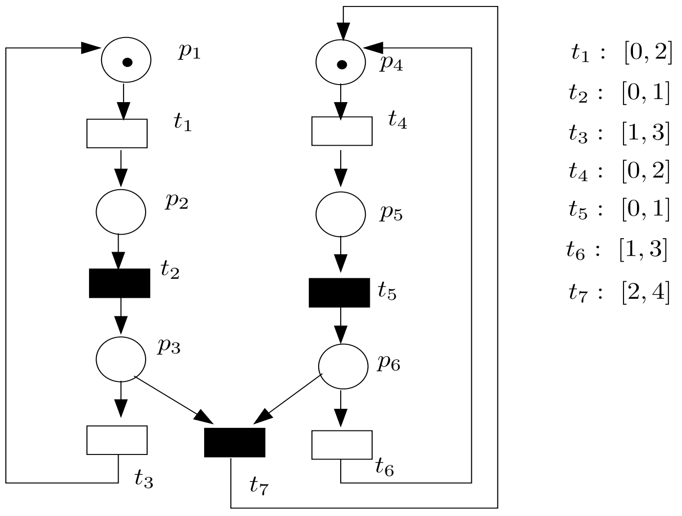

Figure 1. A Time Controlled Petri Nets

Definition 2: A TRG is tuple G=(X, x, Σ, X0, τ), where:

- X, is the set of macro-states;
- x= $\cup$ x_i , where $x_i \subset X_i$, is the set of timed micro-states belonging to a given macro-state X_i;
- $\Sigma=\Sigma_c\cup\Sigma_u$ is the set of events occurring in the system;
- X_0, is the initial macro-state;
- τ is the event time. ■

Each node of the TRG, called a macro-state, represents a particular marking of the TCPN. A macro-state Xi corresponds to a TCPN marking Mi. Each macro-state contains a set of timed micro-states xik which memorizes the evolution of the system due to the elapsing of time inside the same macro-state. Each timed micro-state xki corresponds to particular value of the clocks associated to transitions enabled by the marking Mi. The set of forbidden macro-states is denoted XF. Obviously, all the timed micro-states belonging to a forbidden macro-state are also forbidden. The set of forbidden timed micro-states is denoted xF.

Lemma 1: Let us consider a TRG G=(X, x, Σ, X0, τ), associated with a plant. If G is strongly connected, then the plant has a reversible behavior. ■

According to Lemma 1, it is always possible to return to the initial state if the TRG associated to the plant is strongly connected [4].

We consider that the set of events Σ occurring in the system is divided into two subsets: Σc, the set of controllable events and Σu, the set of uncontrollable events, such that Σ=Σc∪Σu. The controller can act only on controllable events.

Definition 3: A firing sequence is said to be uncontrollable, if and only if each of the timed micro-states reached during its firing are characterized by one of the following output configurations:

1. only the time event τ,
2. at least an uncontrollable event. ■

Definition 4: Let us consider a TRG G=(X, x, Σ, X0, τ). A timed micro-state xik is said to be dangerous, if there exists at least one uncontrollable firing sequence starting in xik and leading to a forbidden state. ■

The set of dangerous timed micro-states is denoted xD. If all the micro-states xik ∈ Xi are dangerous, then the macro-state Xi is dangerous. The set of dangerous macro-states is denoted XD.

In our approach, we impose the reversibility condition. In other words, from every state of the system, there must exist at least a firing sequence leading to the initial state. Therefore, the behavior of the closed loop system respecting the control specifications and the reversibility condition is defined as follows.

Definition 5: The set of admissible timed micro-states of the system, denoted XA, is the biggest set of reachable timed micro-states such that:

- XA∩XF∩XD = ϕ,
- it is possible to reach the initial state of system starting from any reachable state,
- the evolution from an acceptable state to an unacceptable one is done by the execution of a controllable event. ■

A macro-state Xi is considered to be acceptable if and only if all the timed micro-states xik ∈Xi are admissible. The set of admissible macro-states is denoted XA.

The set XB= X-XA-XF-XD defines the deadlock macro-states. They represent TCPN markings from which no transition can be fired. The set xA (resp. XA) represents the biggest set of acceptable timed micro-states (resp. macro-states). Thus, the controlled time reachability graph (TRGc) Gc=(XA, xA, Σ, X0, τ) models the most permissive behavior.

Definition 6: A controller is called most permissive if every macro-state Xi∈XA is reachable during the evolution of the controlled system, and no macro-state Xj∈XF∪XD∪XB is reachable. ■

We propose the following control synthesis algorithm:

Step 1: Initialization

Build the TRG associated to the TCPN model of the plant.

Step 2: The cleaning

Define the set of forbidden macro-states XF.

Identify the set of dangerous timed micro-states xD and the set of dangerous macro-states XD.

Build the controlled time reachability graph (TRGc) by eliminating every macro-state Xi∈XF∪XD∪XB and every timed micro-state xik∈XD;

If the TRGc is strongly connected, then go to Step 3.

ELSE

Compute the strongly connected component (SCC) of the TRGc;

Replace XF by the set of macro-states outside the SCC.

Go to Step 2.1.

Step 3: Computing new firing conditions so that the forbidden states, the dangerous ones and the deadlock states are no longer reachable.

Compute new time conditions for controllable TCPN transitions;

Compute the control variables to completely forbid the execution of some controllable TCPN transitions.

In the sequel we present the tool that we developed to validate this control synthesis technique.

3. Control Synthesis Tool For TDES

The implementation of the algorithm was carried on using Microsoft Visual C++. For the graphic interface we used the MFC classes and for solving the linear inequations system we used the CON'FLEX software. The user interface is intended to be as intuitive, consistent and easy to use as possible. This software tool is available at http://hp.agip.sciences.univ-metz.fr/qsl/interface-rdptempv1.exe.

The architecture of the application is based on the object oriented paradigm. The Time Controlled Petri is implemented as class *tpn*, while the TRG and the TRG^c are implemented in the same class *rgt*, by using adjacency lists. They are both composed of macrostates. A macro state corresponds to a marking and contains a vector of the valid transitions for the marking and the time micro states. For implementation reasons, we introduced micro states as the set of time micro states that have the same time pattern. The list of all macrostates is implemented in class *List*. For the application of the theory of regions, we used separate classes for each type of constraints: separation conditions, reachability conditions, cycle conditions. Each of these classes is a list of nodes containing the index of the node (with different signification for each class) and the vector of coefficients.

The main program asks for an input file containing the TCPN and it takes the following steps: 1) create the TCPN structure ; 2) build the *time reachability graph* ; 3) perform the cleaning ; 4) compute the control variables ; 5) build the controlled time reachability graph with control variables.

3.1. Create the TCPN structure

The time controlled Petri net structure is constructed based on an input file of this form:

- image: the name of the image file for the Petri net, if exists, else blank;
- no of places: $|P|$
- no of transitions: $|T|$
- time limits; state of transition: for each transition T_i, $I_s(T_i)$ and *0* for controllable transitions, *1* for uncontrollable ones
- arcs $P \rightarrow T$ and their weight: $|Pre(P\times T)|$, followed by the matrix;
- arcs $T \rightarrow P$ and their weight: $|Post(P\times T)|$, followed by the matrix;
- the initial marking, as a vector;

the Generalized Mutual Exclusion Constraints, written as a vector, or *0* if none.

3.2. Build the TRG

Building the TRG is done using the following algorithm:

1. Initialization:
    a. Macrostate ms_0 = macrostate corresponding to the initial marking;
    b. ms_0 is added to the TRG and the List;

```
2. For each macrostate *ms* not visited
    a. for each microstate *mss* of *ms* not visited
        i. for each time microstate *tms* of *mss* not visited
            i.1. compute in *firableTrs* the firable transitions for this time microstate;
            i.2. for each firable transition *k*:
                i.2.1. compute new macrostate *nms*;
                i.2.2. compare new macrostate with existing ones in the graph;
                i.2.3. if new macrostate, then add it as a node in the graph and in the list of macrostates;
                i.2.4. add the arc from *tms* to *nms* labeled by *k*
            i.3. add an arc labeled *time* from *tms* to the next time micro state.
```

The TRG and the list of macrostates will be written in two output text files.

3.3. Perform the cleaning

This operation consists of these main steps:

```
1.0. Define the set of forbidden macrostates $X_F$.
2.0. Identify the set of dangerous time microstates $x_D$ and the set of dangerous macrostates $X_D$.
3.0. Build the controlled time reachability graph (TRG$^c$) by eliminating every macrostate $X_i \in X_F \cup X_D \cup X_B$
     and every time microstate $x_i^k \in x_D$;
4.0. IF the TRG$^c$ is strongly connected, THEN exit the cleaning.
     ELSE
     a. Compute the strongly connected component (SCC) of the TRG$^c$ containing the initial
          macrostate;
     b. Replace $X_F$ by the set of macrostates outside the SCC.
5.-1.          Go to Step 1.
```

The initial forbidden macrostates XF are computed during the building of the TRG, as those macrostates that do not respect the GMECs.

The dangerous time microstates are computed by searching in the time reachability graph for all those time micro states that lead through uncontrollable transitions to forbidden macrostates or to dangerous time microstates. If we reach a dangerous or forbidden time microstate or macrostate through a controlled transition, then we perform one of the two following actions: i) if it is a time microstate, then we change the time interval for the transition such that this time microstate could not be created again; or ii) if it is a macrostate, then we add the transition and the source macrostate to the list of transitions to cut (the separation instances).

The TRGc in step 3 is built using the same algorithm as above, with the difference that the new time limits and the separation instances are taken into account. The cleaning operation is performed until XF is the empty. The TRGc for the TCPN in Fig. 1 is presented in Fig. 2.

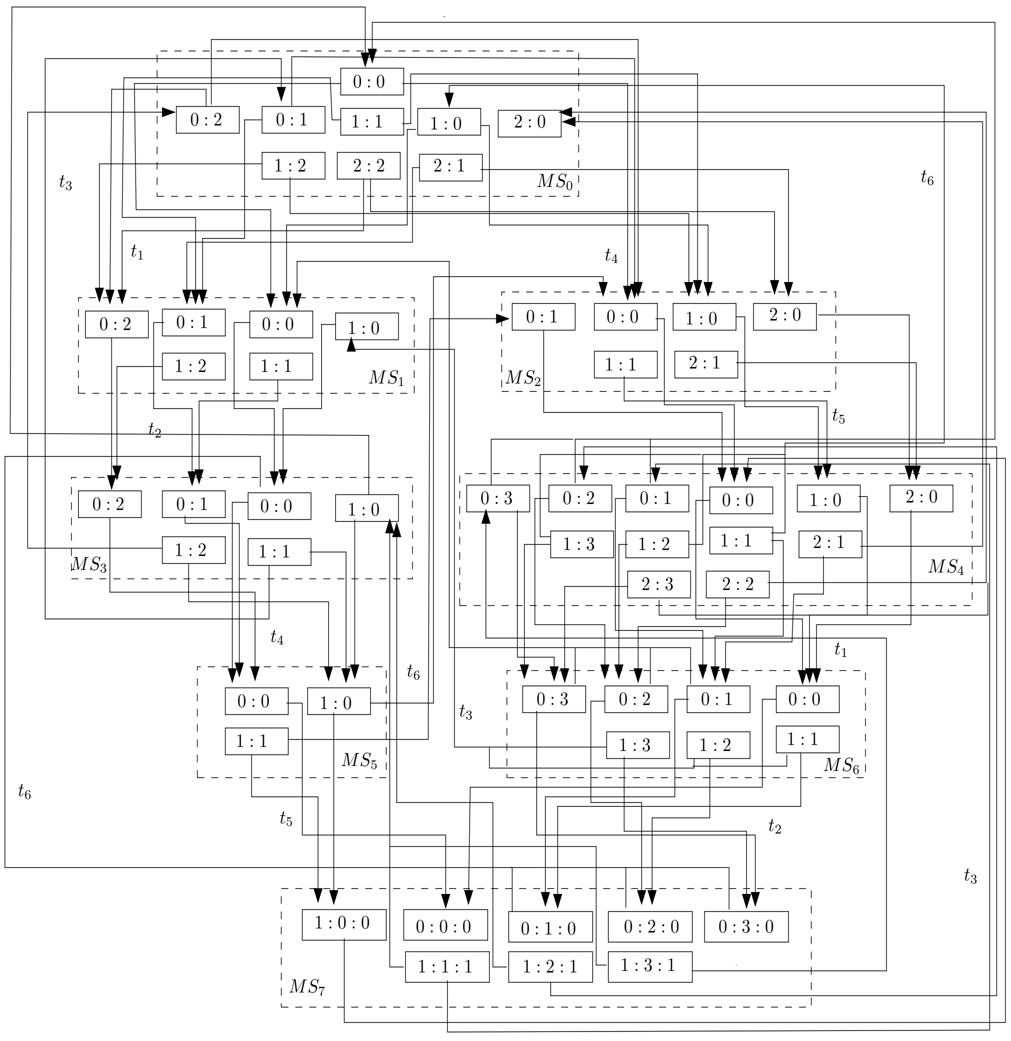

Figure 2. A Time Controlled Reachability Graph

3.4. Compute the control variables

The aim of introducing control variables is to inhibit the firing of controllable transition leading outside the desired behavior. The control variables are determined by using the theory of regions [4].

A transition which is absolutely not allowed to fire from a given TCPN marking (i.e. a macro-state of the TRG) defines a separation instance. The set separation instances is denoted by Ω:

$$\Omega = \left\{ \begin{array}{r} w_k = \left(X_i \xrightarrow{T_m} X_j \right) / T_n \in \Sigma^c, X_i \in TRG^c \\ , X_j \in TRG - TRG^c \end{array} \right\}$$

This technique consists in associating a control variable C_k to each separation instance $w_k \in \Omega$. We also define a control function $U(C_k)$ as follows:

$$U(C_k)/X_j = U(C_k)/X_i + C_k^m,$$

where $U(C_k)/X_j$ denotes the control function value in state X_j; $U(C_k)/X_i$ denotes the control function value in state X_i and $C_k^m \in \mathcal{Z}$ is the incrementing value of the control function due to the occurring of the firing of transition T_m.

The control law denoted $\tilde{U}$ is defined by:

$$\tilde{U}(X_i \xrightarrow{T_m} X_j) = \begin{Bmatrix} 1, if\, U(C_k^m)/X_j \geq 1 \\ 0, if\, U(C_k^m)/X_j < 1 \end{Bmatrix}$$

where C_k^m is the control value associated to firing the transition T_m.

The control variables computing, based on the theory of regions, consists in solving a system of equations which represent the reachability conditions, the cycle equations and the separation conditions. We explain this technique using a simple example.

Let us consider the *TRG* in figure 3. All transitions are supposed to be controllable and the macro-state X_3 is forbidden. There is only one separation instant $w_1 = \left(X_1 \xrightarrow{T_3} X_3 \right)$. Consequently, only one control variable C_1 is necessary.

The macro-states X_0, X_1 and X_2 must be reachable. Thus, the corresponding control function must be positive. The reachability conditions are:

$U(C_1)/X_0 \geq 1,$ (1)

$U(C_1)/X_1 = U(C_1)/X_0 + C_1^1 \geq 1,$ (2)

$U(C_1)/X_2 = U(C_1)/X_0 + C_1^1 + C_1^2 \geq 1.$ (3)

There is also only one elementary cycle. As the control function characterizes the associated macro-state, the cycle equation is:

$C_1^1 + C_1^2 + C_1^3 = 0.$ (4)

Furthermore, as the system must not reach the forbidden macro-state X_3, we introduce the following separation condition :

$U(C_1)/X_3 = U(C_1)/X_0 + C_1^1 + C_1^3 < 1.$ (5)

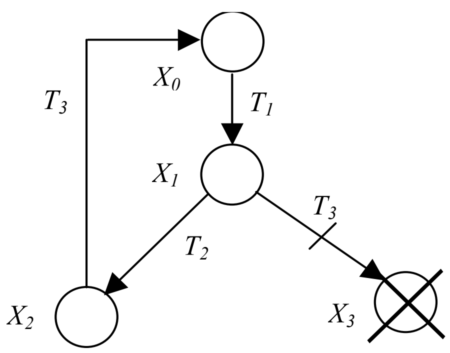

Figure 3. A simple TRG

If at least a solution exists, solving the equations system (1)-(5) allows computing the control variable associated to each event and the control function of the initial macro-state. One of the solutions for the TRG given in figure 3 is

$U(C_1)/X_0 = 1,$

$C_1^1 = 0,\ C_1^2 = 1,\ C_1^3 = -1.$

The most permissive TRG^c and the evolution of the control function is shown in figure 4.

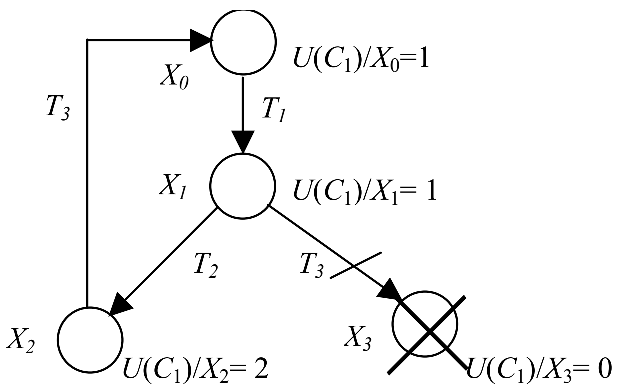

Figure 4. The corresponding *TRG*^{c}

The control function $U(C_1)$ is updated by the Petri net controller given in figure 5, according to the occurrence of the events T_2, and T_3. As C_1^1=0, the transition T_1 has no effect on $U(C_1^1)$.

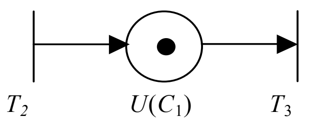

Figure 5. Petri net controller

Let us consider now the TCPN illustrated in figure 1, with the control specification given by the CMEC $M(P_2)+M(P_5)<2$.

The TRG^c associate to this control synthesis problem is shown in fig. 2. For simplicity, we illustrate only the macrostates in fig. 6. One can notice two separation instances: firing transition T_1 from the macrostate X_2 and firing T_4 from X_1. With each separation instance we associate a control variable C_i, i=0,1. Furthermore, for each control variable we write the separation condition, the reachability conditions and the cycle conditions. In the sequel we give the conditions associated with the control variable C_1.

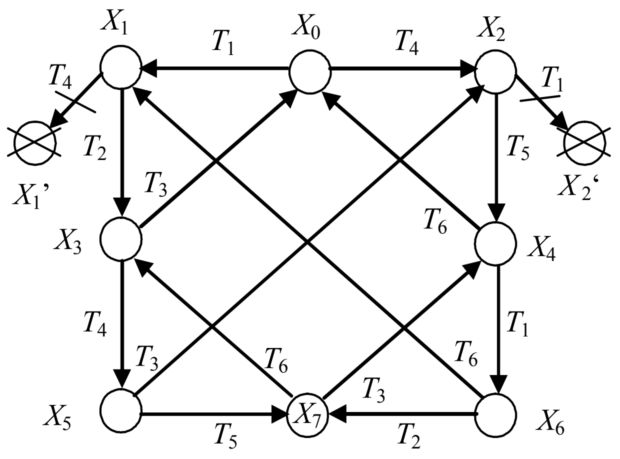

Figure 6. The graph of macrostates

A macrostate is legal only if the value of the associated control function is more than 1. As the macrostate X_1' is forbidden, this constraint is specified by the separation condition:

$$U(C_1)/X_0 \ + C_1^1 + C_1^4 < 1$$

On the other hand, the macrostates X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7 must be reachable. This requirement is modeled by the reachability conditions:

$$U(C_1)/X_0 \geq 1$$

$U(C_1)/X_0 \; + C_1^{1} \geq 1$

$U(C_1)/X_0 \; + C_1^{4} \geq 1$

$U(C_1)/X_0 \; + C_1^{1} + C_1^{2} \geq 1$

$U(C_1)/X_0 \; + C_1^{4} + C_1^{5} \geq 1$

$U(C_1)/X_0 \; + C_1^{1} + C_1^{2} + C_1^{4} \geq 1$

$U(C_1)/X_0 \; + C_1^{4} + C_1^{5} + C_1^{1} \geq 1$

$U(C_1)/X_0 \; + C_1^{1} + C_1^{2} + C_1^{4} + C_1^{5} \geq 1$

Finally, for each elementary cycle is described by a cycle equation:

$C_1^{1} + C_1^{2} + C_1^{3} = 0,$

$C_1^{4} + C_1^{5} + C_1^{6} = 0.$

All the inequations and equations are solved using the CON'FLEX software. If at least a solution exists, solving the equations system allows computing the control variable associated to each transition and the control function of the initial macro-state. For our example, one of the solutions is:

$U(C_0)/X_0 = U(C_1)/X_0 = 2$; $C_0^{1} = C_1^{1} = -1$; $C_0^{2} = C_1^{2} = 1$; $C_0^{3} = C_1^{3} = 0$; $C_0^{4} = C_1^{4} = -1$; $C_0^{5} = C_1^{5} = 1$; $C_0^{6} = C_1^{6} = 0$; $C_0^{7} = C_1^{7} = 0$.

It is to be noticed that, in this case, the control variables C_0 and C_1 are redundant. Thus, one control variable is enough to control the system. The Petri net controller is given in fig. 7.

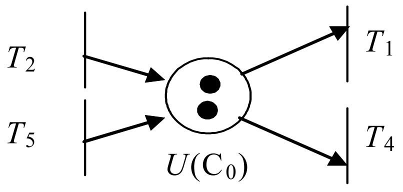

Figure 7. The Petri net controller

3.5. Case study

The application was tested on an AMD platform computer, with AMD Athlon Processor 1.54 GHz, 512 MB RAM running Windows XP operating system. The following table presents the results obtained testing various TCPN examples. We have excluded the amount of time taken by CON'FLEX to solve the linear system.

	No. macrostates *before*	No. macrostates *after*	No. Control Var.	Time – time(CON'FLEX)
Ex. 1	261	215	3	14.20 sec
Ex. 2	261	232	3	13.72 sec
Ex. 3	187	135	6	10.26 sec

4. Algorithm Complexity

Given a TCPN N=(P, T, Pre, Post, Is, C) and its initial marking M0, we consider r as the maximum value of matrices Pre, Post, Is, M0. Then we can define the size of the TCPN as $n = \Theta(|P| \times |T| \times r)$. The algorithm complexity will be computed according to n. We denote by u and v the number of vertices (corresponding to macrostates) and arcs in the reachability graph.

The building the TRG is based on the algorithm of building the reachability graph for a general Petri net to which we added the time interval information. The algorithm is exponential in time and space, related to the size of the Petri net. Given that the Petri nets considered are bounded, the TRG is finite.

The cleaning algorithm presented in section 3.3 has four main steps and their complexity will be discussed in the following. The first two steps has each an *O(u+v)* time complexity, as they involve a binary search in the graph. We presume that the vector comparing is performed in *O(1)* time complexity. The construction of the TRG^c is done by rebuilding the reachability graph with the new constraints (markings and transitions to cut, new time intervals for the controlled transitions) and it takes exponential time in the size of the net. Computing the strongly connected component of the graph obtained at step 3 that contains the initial macrostate is performed in $O(u^2)$ time. Thus the complexity of the cleaning algorithm is of exponential time in the size of the Petri net.

Computing the control variables involves computing the spanning tree of the controlled time reachability tree for building the separation condition and the reachability inequations and the basis cycles for the cycle equations. These operations are done in polynomial time ($O(u^2)$) with respect to the size of the graph. Solving the linear inequation system is performed by the simplex algorithm implemented in the CON'FLEX software. The simplex algorithm takes polynomial time. The complexity of this step is polynomial in the size of the graph.

The memory management for the application is done by dynamic allocation. For implementation efficiency reasons we grouped the time microstates into microstates, according to their time patterns. Thus, a macrostate contains the associated marking, the indexes of the valid transitions that can be applied to the marking and the list of microstates. The graphs are implemented using adjacency lists. Each node of the graph contains only the indexes of the macrostate, microstate, time microstate, the information about these components being memorized in the list of macrostates.

5. Conclusion

In this paper we proposed control synthesis software tool for discrete event systems based on an original approach. This method is based on TCPN model. The controller can 1) fix the firing instant of controllable events, by acting on the corresponding time constraints or 2) it can completely forbid the firing of some controllable transitions using control variables. This approach provides a live controller which is maximally permissive in the sense of Petri nets marking. The complexity of the algorithm is also discussed. Our further research work deals with taking into account the unobservable nature of some events.

REFERENCES

1. ACHOUR, Z., REZG, N. and XIE, X., **On the existence of Petri net controller for discrete event systems under partial observation**, IFAC'05, Prague, 2005.
2. ALTISEN, K., GOESLER, G., PUNELI, A., SIFAKIS, J., TRIPAKIS, S. and YOVINE, S., **A framework for scheduler synthesis**, Proceedings of the 1999 IEEE Real–Time Systems Symposium, Phoenix AZ, USA, 1999.
3. BRANDIN, B., and WONHAM, W.M., **Supervisory Control of Timed Discrete Event Systems**, IEEE Transactions on Automatic Control, 39(2): 329-341, 1994.
4. GHAFFARI, A., REZG, N. and XIE, X., **Design of Live and Maximally Permissive Petri Net Controller Using the Theory of Regions**, IEEE Transactions on Robotics and Automation, Vol. 19, pp; 137-142, 2003.
5. HOLLOWAY, L.E. and KROGH, B.H., **Synthesis of feedback control logic for a class of controlled Petri nets**, IEEE Trans. On Automatic Control, 35(5), pp. 514-523, 1990.
6. KUMAR, R., **Supervisory Synthesis Techniques for Discrete Event Dynamical Systems**, PhD thesis, University of Texas AT , 1991.
7. MALER, O., PUNELI, A. and SIFAKIS, J., **On the synthesis of discrete controllers for timed systems**, Proc. STACS'95, LNCS, 900:229-242, 1995.
8. MURATA, T., **Petri nets: properties, analysis and application**, Proceedings of IEEE, 44(4), pp. 541-579, 1989.

9. RAMADGE, J.G. and WONHAM, W.M., **Supervisory Control of a Class of Discrete Event Processes**, SIAM J., Control and Optimization, 25: 206-230, 1987.

10. RAMADGE, J.G. and WONHAM, W.M., **The Control of Discrete Event Systems**, Proceedings of the IEEE, 77(1): 81-97, 1989.

11. REZG, N., XIE, X. and GAFFARI, A., **Supervisor Control in DES Using The Theory of Regions**, DES: Analysis and Control, WODES'00, Ghent, Belgium, 2000.

12. SAVA, A.T., **Sur la synthèse de la commande des systèmes à événements discrets temporises**, PhD thesis, Laboratoire d'Automatique de Grenoble, Institut National Polytechnique de Grenoble, 2001.

13. SAVA, A.T. and ALLA, H., **A Control Synthesis Approach for Time Discrete Event Systems**, CESA'03, CDROM, Lille, France, july 2003.

14. YAMALIDOU, K., MOODY, J., LEMON, M. and ANTZAKLIS, P., **Feedback Control of Petri Nets Based on Place Invariants**, Automatica, 32(1), pp. 15-28, 1996.

15. YOVINE, S., **Méthodes et outils pour la vérification symbolique des systèmes temporises**, PhD thesis, VERIMAG, Institut National Polytechnique de Grenoble, 1993.