Formal Semantics for Grafcet Controlled Systems

JANAN ZAYTOON
Laboratoire d'Automatique et de Microélectronique
Faculté des Sciences
Moulin de la Housse, BP 1039, 51687 Reims cedex 2
FRANCE

Abstract: Grafcet is a widely used model for the specification of logic control in manufacturing systems. Compared to other modelling tools for Programmable Logic Controllers (PLC), it has the advantages of manipulating simple concepts which are commonly used by control agents and engineers. This paper is a first in a series of two papers that present an approach based on the use of the "Timed Transition Model (TTM) / Real-Time Temporal Logic (RTTL)" formalism as a support to the analysis and verification of properties of automated systems whose controllers are specified using Grafcet. In this paper, the modelled system is mapped into a TTM, and the mapping function associates formal semantics to Grafcet and its interactions with the controlled plant in terms of the TTM.

Key-Words : Grafcet, semantics, Timed Transition Model (TTM), mapping function CSCC'99 Proc.pp.5671-5678

1 Introduction

The design of manufacturing systems and the development of their controllers are getting more closely linked as the manufacturing environment is becoming more automated [1], [2]. Controller designers must be able to integrate machines and material handling equipment in accordance with the desired operational decision and control functions. This requires the availability of adequate modelling and validation tools in order to determine whether or not the manufacturing system and its controller will function in the desired manner.

Grafcet [3], [4] o r*sequential function charts* is an international standard used for the specification of sequential control in manufacturing systems. The graphical representation of Grafcet allows a clear modelling of concurrency, synchronisations as well as the inputs and outputs and their relations. This makes Programmable Logic Controllers (PLCs) more tractable and simplifies the simulation of the control logic of the system. Many PLC builders today use the Grafcet as a specification and/or as a programming language. Among the large companies using it widely or recognising it as an internal standard are: Siemens, Renault, Peugeot, Michelin, and others. Recent works also reported the use of Grafcet to implement supervisory control applications and to structure rule-based systems [5]. In spite of its advantages, Grafcet has long been criticised because it was not supported by a formal semantics that allows for unambiguous interpretation of a given model, and that provides means for the analysis and verification of safety, liveness and timeliness properties of a given Grafcet [6], [7].

This paper is the first in a series of two papers that present an integrated approach combining features from Grafcet and the TTM (Timed Transition Model) / RTTL (Real-Time Temporal Logic) formalism in order to provide a global framework for the validation of systems controlled by Grafcet. The aim of the work presented in this paper is to establish formal semantics for Grafcet and for its interactions with the controlled systems. After a review of Grafcet (section 2) and of the TTM/RTTL framework (section 3), a mapping function that associates formal semantics to Grafcet controlled systems in terms of the TTM is presented in section 4.

2 Grafcet

Grafcet is a discrete-event modelling tool that integrates the ability of Petri nets for concurrent modelling, and the use of variables and Boolean functions to represent complex decisions. These features, together with its simplicity to represent the behaviour of control systems and its normative character, explain its wide industrial implementation. Grafcet consists in describing parallel and synchronised sequences of elementary operations applied to the plant with due

consideration to plant's response. The basic concepts of this model are quite clear and simple: the *step*, the *action*, the *transition* and its associated *receptivity* (Fig. 1).

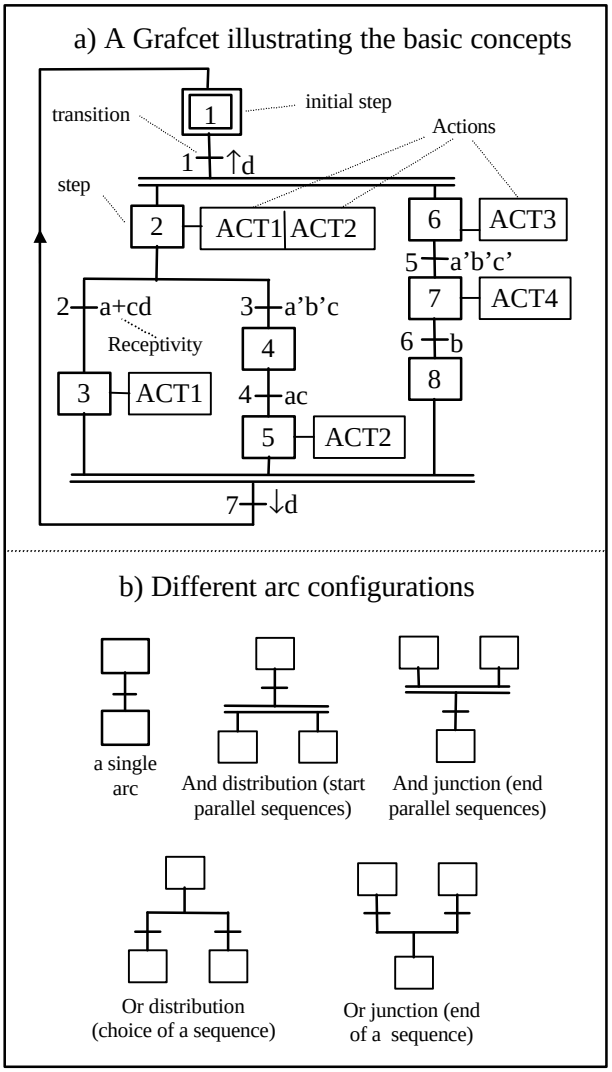

Fig. 1: Basic concepts of Grafcet

The step, drawn as a square, represents a partial state of the controller to which actions can be associated. A step can be active or idle; associated actions are performed when the step is active and remain asleep when it is idle. A situation is given by the set of active steps. The transition, represented as a bar, links one (or several) previous step(s) to one (or several) following step(s). It represents the fact that the actions of the previous steps are followed by the actions of the following ones and figures a decision of changing system state. A logical expression, called receptivity, is associated to each transition. This expression manipulates Boolean variables, corresponding to controller inputs or to the activation state of individual Grafcet steps, and events corresponding to the rising and falling edges of input variables. A rising edge of a variable v is given by ↑v, the falling edge is given by ↓v.

The evolution of Grafcet is traditionally given by the following evolution rules [4]:

- Rule 1: Grafcet initial situation is given by all of its initial steps (drawn by a double square).
- Rule 2: a transition is firable if all of its previous steps are active. A firable transition is fired if its associated receptivity is true.
- Rule 3: the firing of a transition results in the deactivation of its previous steps and the simultaneous activation of its following steps.
- Rule 4: simultaneously firable transitions are simultaneously fired.
- Rule 5: if a step is to be simultaneously activated and deactivated, it remains active.

Many extension, including the introduction of macro-steps and partial Grafcets with forcing orders hierarchy have also been introduced to consolidate the modelling power of Grafcet [8].

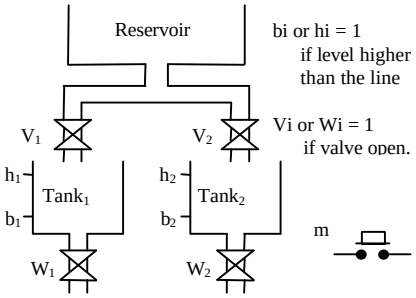

Fig. 2: Example: tank filling

2.1 Example - tank filling

A simple example, taken from David [3], is used to illustrate the basic concepts of Grafcet. The process in question (Fig. 2) includes two tanks used in a similar way. Tank 1 is empty when the level is less than b_1 and is full when the level is greater than h_1. These conditions are given respectively by: b_1=0 and h_1=1. At the initial state, both tanks are empty. If push button m is pressed, both tanks are filled by opening the inflow valves V_1 and V_2. When a tank is full, filling stops (by closing the corresponding inflow valve) and its contents start to be used (by opening the corresponding outflow valve, W_1 or W_2). When a tank is empty, the corresponding outflow valve is closed. Filling may only start up again when both tanks are empty and if the button m is pressed.

The Grafcet corresponding to this specification is given in Fig. 3. Initially, steps 1 and 4 are active. Transition 1 which follows these steps can be fired

as soon as its associated Boolean variable m has the value 1. After this firing, steps 2 and 5 are active and their associated actions (V1 and V2, respectively) are performed. In this situation, transition 2 can be fired if h_1 =1, and transition 4 can be fired if h_2 =1. And so on. Concurrency is explicitly represented in this model. Steps 1, 2 and 3 correspond to the states of tank 1 (empty, during filling, and during emptying, respectively) and steps 4, 5, and 6 correspond to the states of tank 2. The sequence of states (or active steps) and Boolean conditions leading from one state to another are quite apparent

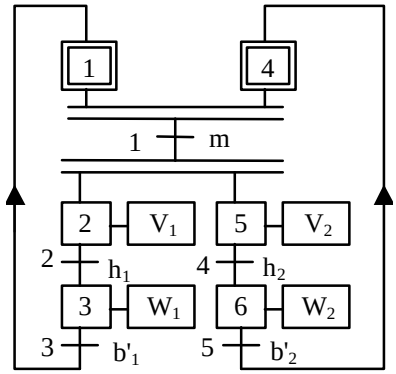

Fig. 3: Grafcet specification of tank filling

2.2 Temporal behaviour of Grafcet

The application of the five evolution rules of Grafcet results in changing a Grafcet situation into another situation. This new situation may be unstable (if some transitions are firable in this situation) and must change again before the occurrence of a new input event. Issues related to reactivity, determinism and stability have therefore been raised and led to the definition of a new temporal framework in which the modelling universe of Grafcet is partitioned into an internal and an external time scales that have no common measure [8], [9]. At the external time scale, all changes of the values of the receptivities associated with the transitions are considered as soon as they occur and their consequential reactions (the actions associated with the next stable situation) are perceived as occurring at the same time instant; this ensures the reactivity of the model. At the internal time scale, a reaction involves a number of consecutive internal evolutions whose durations are as small as necessary; the actions associated with intermediate unstable situations do not affect the controlled plant. This behaviour guarantees the determinism of Grafcet since one and only one output scenario (activation and deactivation of actions) may result from an input scenario.

In spite of their simplicity, Grafcet evolution rules and the above temporal framework are not sufficient to guarantee a unique interpretation of a given Grafcet [10]. To remove persisting ambiguities, the semantics of Grafcet have been completed by an algorithmic "Grafcet player" [6] which organises and explains the interactions between the basic evolution rules and the two time scales of the model. The semi-formal semantics provided by this algorithm insures a deterministic interpretation of Grafcet and reinforces its synchronous and reactive nature. The mapping function given in section 4.2 represents a formal definition of this semantics in terms of Timed Transition Model (TTM).

The choice of the TTM formalism as a support for associating semantics to Grafcet is motivated by the fact that the TTM has rich semantics, including the manipulation of different types of variables and time. It presents a flexible model for representing a set of concurrent processes, whether these processes are hardware devices (e.g. pumps, valves and reactors) or originate as programs in various real-time programming languages, Petri nets, Statecharts, or Grafcet. This flexibility allows to support both the synchronous, reactive and deterministic nature of Grafcet, and the asynchronous non-deterministic nature of the controlled plant. Furthermore, the use of temporal intervals is extremely useful for the modelling of plant evolution times, since these times cannot be precisely known during the specification phase. The TTM is also an element of the TTM/RTTL varification framework which will be adapted in the second paper to provide a formal validation support for Grafcet controlled systems.

3 TTM/RTTL Framework

The TTM/RTTL framework is a state-based, linear discrete time, interleaved, asynchronous, and explicit linear logic formalism [11]. It includes the following elements:

Semantic model of time: the notion of a possible behaviour or trajectory of a system is given by an infinite sequence alternating events and states. A discrete notion of time is employed using an explicit clock whose current time is represented by the non-negative integer variable "t". The tick event, which increments "t" by one, occurs infinitely often in the trajectory and is interleaved with other system events. Time bounds on events determine when they may occur relative to the ticks.

Timed Transition Model: TTM is basically an asynchronous model that represents most real-time features such as delays, time-outs, parallel

processing, communication through shared variables, as well as message passing over channels. A TTM is defined as a three-tuple ($V, \Theta, \Im$) where V is the set of variables used, Θ is a predicate asserting an initial condition on the variables and $\Im$ is the set of all transitions (representing events). A transition τ is a 4-tuple (e_τ, h_τ, l_τ, u_τ); where e_τ is an enabling condition, h_τ is a transformation function, l_τ and u_τ are constants representing the lower and upper time bounds respectively. These bounds indicate that a transition which is continuously enabled over an interval of time does not actually occur for l_τ ticks of the clock, but must occur by u_τ ticks of the clock unless it becomes disabled. The tick event corresponds to a distinct transition belonging to $\Im$. A spontaneous transition, with [0, ∞] time bounds represents an event that may occur at any moment. However, it may also delay occurring forever. Spontaneous transitions are useful to represent situations where the designer initially has no knowledge of the time bounds or to model unpredictable behaviour in the plant, such as the failure of a device. When more than one transition are enabled and eligible (by virtue of their bounds) to occur at a point in time, the order of firing is chosen nondeterministically.

Real Time Temporal Logic: RTTL is an extension of Manna and Pneuli [12] untimed temporal logic to timed systems. It is an expressive language, used to specify the properties to be verified in the semantic time model corresponding to the TTM of a plant and its controller. RTTL is an explicit clock logic because its expressions may explicitly use the clock time variable "t". The basic operators used in RTTL are: ○ (next), *U* (until), □ (henceforth), ◊ (eventually), U (unless) and *P* (precedes). These operators allow to specify qualitative temporal properties. Quantitative temporal properties (or timeliness properties), which are used to specify exact time, maximum time, minimal time and periodicity, can be expressed by bounding the time interval of the operators. The reader may refer to [13] for a detailed description of the semantics of these operators.

Verification via proof systems and heuristics: The initial proof system of the TTM/RTTL provides algorithms and an implemented verifier to check whether all legal trajectories of a finite state TTM satisfy a given RTTL specification. If a RTTL property fails to hold, then the failing trajectories are provided, making it possible to debug the system. An advantage of RTTL is that no new temporal operators are introduced. As a result, all the proof rules of Manna-Pneuli temporal logic can be used and other rules are added for the real-time part of the reasoning. For infinite state systems, the RTTL has theorem proving analysis techniques together with heuristics that require interactive user guidance for doing proofs using proof diagrams and weakest preconditions. A proof diagram is an abstract view of a state reachability graph that contains the intuition of system execution without the distracting proliferation of states.

4 Associating Semantics to Grafcet in terms of TTM

This section presents the formal definition of the function that maps Grafcet into an equivalent TTM. The TTM transitions structure resulting from the application of the mapping function is depicted in Fig. 4. This structure insures a correct temporal behaviour of Grafcet in terms of synchronism, determinism and reactivity [6]. Synchronism is achieved by using a zero-time-bounds TTM transition for each possible internal evolution of Grafcet, whether this evolution corresponds to the firing of a single Grafcet transition or to a number of Grafcet transitions simultaneously. Determinism is guaranteed by associating exclusive conditions to TTM transitions related to Grafcet evolutions, and therefore only one internal Grafcet evolution may be possible at a given instant. Reactivity is maintained by giving priority to internal Grafcet evolutions (due to [0, 0] time bounds) over input and output communicating transitions (having [0, 1] time bounds) which, in turn, have higher priority over plant evolution transitions (whose lower time bounds are given by a positive integer). Therefore, when a Grafcet situation resulting from an internal evolution is unstable, the TTM transition corresponding to the next internal Grafcet evolution will occur before any enabled TTM communicating transition. This cycle continues until a stable situation is reached when all TTM transitions corresponding to Grafcet evolutions are disabled. In this case, the enabled output communicating transitions will occur to transmit Grafcet actions associated with the stable situation to the plant, and to receive the next inputs from the plant.

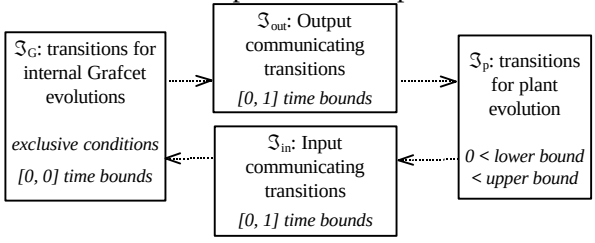

Fig. 4: TM transitions structure

Timed actions, stored actions, pulsed actions, and forcing actions are not addressed in this paper because a consensual interpretation of these actions, within the frame of deterministic, synchronous, and reactive semantics of Grafcet is not yet established.

4.1 Definition of Grafcet structure and preliminary analysis

The mapping function, which is given in the following sub-section, is based on a 4-tuple definition of Grafcet structure: G = (V, X, T, I), where:

- V is the set of variables given by: V = Vin ∪ Vout, where Vin is the set of variables originating from the plant and Vout is the set of binary variables representing Grafcet actions.
- X is the set of steps, X = {X_1, X_2, X_3, ...}. The actions associated to a step X_i , which are given by the set "$action_i \subseteq$ Vout", should be set to 1 when X_i is active.
- T is the set of Grafcet transitions, T = {t_1, t_2, t_3, ...}. A Grafcet transition t ∈ T is defined by the 3-tuple ($X_{PR}(t)$, $X_{FO}(t)$, φ(t)), where $X_{PR}(t)$ is the set of previous steps of t, $X_{FO}(t)$ is the set of following steps of t and φ(t) is the receptivity associated to t.
- I is the set of initial steps, I⊆ X.

For example, the Grafcet of Fig. 1-a is defined by:

- Vin = {a, b, c, d};
- Vout = {ACT1, ACT2, ACT3, ACT4};
- X = {X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8}, $action_1$ = ∅, $action_2$ = {ACT1, ACT2}, $action_3$ = {ACT1}, $action_4$ = ∅, $action_5$ = {ACT2}, $action_6$ = {ACT3}, $action_7$ = {ACT4}, $action_8$ = ∅.
- T = {t_1, t_2, t_3, t_4, t_5, t_6, t_7}, t_1 = ({X_1}, {X_2, X_6}, ↑d), t_2 = ({X_2}, {X_3}, a+cd), t_3 = ({X_2}, {X_4}, a'b'c), t_4 = ({X_4}, {X_5}, ac), t_5 = ({X_6}, {X_7}, a'b'c'), t_6 = ({X_7}, {X_8}, b), t_7 = ({X_3, X_5, X_8}, {X_1}, ↓d).
- I = {X_1}.

Based on the set of Grafcet transitions, T, the following definitions are introduced to be used in the Grafcet to TTM mapping function:

1) $T^* = 2^T - \emptyset$, is the set of all the sets, excluding the empty set, in which each transition t∈T occurs at most once. Example: if T = {t_1, t_2, t_3} then T^* = { {t_1}, {t_2}, {t_3}, {t_1, t_2}, {t_1, t_3}, {t_2, t_3}, {t_1, t_2, t_3} }.
2) {t_i , t_k} ∈ *sim* IFF t_i, t_k ∈T are correlated and non-contradicting transitions, where:
   - t_i, t_k ∈T are correlated if Grafcet structure and plant dynamics allow them to fire simultaneously.
   - t_i, t_k ∈T are contradicting if (((φ(t_i)=true) ⇒ (φ(t_k)=false)) and ((φ(t_k)=true) ⇒(φ(t_i)=false))).

   This definition implies that: ∀t_i ∈T : {t_i, t_i} ∈ *sim*. The calculation of correlated and non-contradicting transitions will be illustrated by means of the Grafcet given in Fig. 1-a. The structure of this Grafcet implies that when step 1 is active, all other steps are idle. Therefore, each of the transitions t_1 and t_7 (which deactivates and activates step 1, respectively) can not fire simultaneously with any other transition. When step 1 is idle, the distribution structure of this Grafcet implies that:
   - steps 6, 7 and then 8 are activated sequentially. Therefore, transitions t_5 and t_6 can not fire simultaneously,
   - steps 4 and 5 are activated sequentially. Therefore, transitions t_3 and t_4 can not fire simultaneously.

   Thus, correlated Grafcet transitions are given by the couples : (t_2, t_3), (t_2, t_4), (t_2, t_5), (t_2, t_6), (t_3, t_5), (t_3, t_6), (t_4, t_5), (t_4, t_6). Among these couples, (t_2, t_5), (t_3, t_5), (t_3, t_6), and (t_4, t_5) represent contradicting transitions because the two receptivities of each of these transition couples can never become true simultaneously. Therefore, the couples of Grafcet transitions that can be simultaneously firable are given by: *sim* = { {t_1, t_1}, {t_2, t_2}, {t_2, t_3}, {t_2, t_4}, {t_2, t_6}, {t_3, t_3}, {t_4, t_4}, {t_4, t_6}, {t_5, t_5}, {t_6, t_6}, {t_7, t_7} }.
3) T ∈ $\widetilde{sim}$ if T ∈T^* and (∀ t_i ∈ T , ∃ t_k ∈ T such that {t_i, t_k} ∈ *sim*). Each element of $\widetilde{sim}$ is a set that contains either a single Grafcet transition, or a number of Grafcet transitions each of which is simultaneously firable with at least another transition of the same set.
   For the above example, $\widetilde{sim}$ = { {t_1}, {t_2}, {t_3}, {t_4}, {t_5}, {t_6}, {t_7}, {t_2,t_3}, {t_2,t_4}, {t_2,t_6}, {t_4,t_6}, {t_2,t_3,t_4}, {t_2,t_3,t_6}, {t_2,t_4,t_6}, {t_2,t_3,t_4,t_6} }.
4) T^{MAX} = {T∈ $\widetilde{sim}$: ∄ T_i∈ $\widetilde{sim}$ such that T⊂T_i}; this is the set containing the maximum elements of $\widetilde{sim}$. For the previous example, T^{MAX} = { {t_1}, {t_5}, {t_7}, {t_2,t_3,t_4,t_6}}.
5) ∀Tm ∈ T^{MAX}: Tm^{MAX*}= {T ∈ 2^{Tm} such that ∀t_i, t_k ∈ T : {t_i, t_k} ∈ *sim* }. Each element of Tm^{MAX*} is a set that contains either a single Grafcet transition or a number of simultaneously firable Grafcet transitions that are contained in Tm. For the same example: T_1^{MAX*} ={{t_1}}, T_2^{MAX*} = {{t_5}}, T_3^{MAX*} = { { t_7}}, and T_4^{MAX*} = { {t_2},

$\{t_3\}, \{t_4\}, \{t_6\}, \{t_2, t_3\}, \{t_2, t_4\}, \{t_2, t_6\}, \{t_4, t_6\}, \{t_2, t_4, t_6\}$ }.

This preliminary analysis of Grafcet structure allows the mapping function to limit the number of TTM transitions corresponding to the internal evolution of Grafcet to only those evolutions that may potentially occur. For the above example, this analysis results in identifying 13 possible evolutions (a single evolution for each of: T_1^{MAX*}, T_2^{MAX*} and T_3^{MAX*}, and 9 evolutions for T_4^{MAX*}). If this analysis is not carried out, then the synchronous and parallel nature of Grafcet would imply the calculation of 2^7=128 TTM transitions to cater for all evolution possibilities of Grafcet. This economy in the number of calculated TTM transitions also allows a substantial reduction in proofs' complexity (see the second paper).

4.2 Mapping function

The global TTM of the system under development includes a TTM equivalent to Grafcet, $TTM_G = (V_G, \Im_G, \Theta_G)$, and a TTM representing the plant, TTMp = (Vp, $\Im$p, Θp). These two concurrent TTMs communicate by means of input and output TTM transitions which are generated systematically. Input transitions (given by the set $\Im_{in}$) allow the Grafcet to receive the values of input variables from the plant. Output transitions (given by the set $\Im_{out}$) transmit Grafcet actions to the plant. Figure 4 gives a layout of TTM transitions structure obtained by applying the mapping function and the rules presented in the following two sub-sections.

4.2.1 Grafcet to TTM mapping function

The function 'f: G $\rightarrow$ TTM_G' is defined by the following mappings:

- **variables mapping**: The set of variables of TTM_G is defined by V_G = {♥, ✖, Edge}, where:

- ♥ and ✖ have the same definition as in §4.1. Each $X_i \in$ ✖ is a binary variable that is set to 0 when the corresponding step is idle and to 1 when the step is active.
- Edge is an integer variable. Each of the possible values of Edge corresponds to the occurrence of a rising or a falling edge of a distinct input variable; the correspondence is defined by associating two constants to each input variable as follows:
  - $\forall v \in$ ♥in: define $v_{re} \in N^+$ and $v_{fe} \in N^-$ such that $v_{re} = -v_{fe}$
  - $\forall v, v' \in$ ♥in : if $v \neq v'$ then $v_{re} \neq v'_{re}$.

The interpretation of the different values of Edge is as follows:

- Edge=0, no input event has occurred since the last evolution of Grafcet;
- Edge= v_{re}, input event corresponding to the rising edge of the variable v has occured;
- Edge= v_{fe}, input event corresponding to the falling edge of the variable v has occured.

Only one value can be assigned to Edge at a given instant because the reactivity of Grafcet and the asynchronous nature of the controlled plant imply that two input events cannot occur simultaneously [8]. For the Grafcet of Fig. 1-a, the values of Edge range between -4 and 4; the correspondence between these values and the edges of input-variables may be defined as follows: a_{re}=1, a_{fe}=-1, b_{re}=2, b_{fe}=-2, c_{re}=3, c_{fe}=-3, d_{re}=4, d_{fe}=-4.

- **initial state mapping:** The initial condition Θ_G is given by the following mappings:

1) $\forall X_i \in$ ✖ : X_i = 1 if $X_i \in$ ▮ , else X_i = 0; step variables corresponding to the initial steps are set to one and the others to zero.
2) $\forall v \in$ ♥out , $\forall X_i \in$ ▮ : v=1 if $v \in action_i$, otherwise v=0. Actions of initial steps are performed.
3) Edge=0; no input event occurs during initialisation.

For Grafcet of Fig. 1-a, the state of the corresponding TTM after initialisation is given by: X_1=1, $\forall$i=2 to 8: X_i =0, ACT1=0, ACT2=0, ACT3=0, ACT4=0, Edge=0.

- **transitions and actions mapping:** Each transition of TTM_G represents a single possible evolution of Grafcet and corresponds to an element of Tm^{MAX*} (defined in §4.1). Such transitions are generated as follows :

$\forall Tm \in T^{MAX}$, $\forall S \in Tm^{MAX*}$: generate the transition T = (e_T , h_T , 0, 0) $\in \Im_G$, where:

- the bounds [0, 0] allow to guarantee the synchronism of Grafcet evolution.
- $$e_T = \bigwedge_{t \in S} \left[\left(\bigwedge_{Xi \in X_{PR}(t)} X_i \right) \wedge \varphi'(t) \right] \wedge \left[\bigwedge_{t \in (Tm - S)} \neg \left[\left(\bigwedge_{Xi \in X_{PR}(t)} X_i \right) \wedge \varphi'(t) \right] \right]$$

where $\varphi'(t)$ corresponds to a rewriting of the expression of the receptivity $\varphi(t)$ in which each of the rising edges $\uparrow$v is replaced by the logical test

(Edge=v_{re}) and each of the falling edges $\downarrow$v is replaced by the logical test (Edge=v_{fe}). The condition e_T represents Grafcet situation that leads to the firing of all the transitions of a set $S \in Tm^{MAX*}$. This condition is true when all of these transitions are firable (their previous steps are active and the associated conditions are true) provided that all the other transitions belonging to the corresponding maximum element in T^{MAX} are disabled. The case in which one of these other transitions is also enabled implies that another superior set of Tm^{MAX*} is firable and hence the evolution under consideration is disabled by virtue of the second term of e_T. Therefore, TTM_G is rendered deterministic since only one of its transitions is enabled at a given situation and corresponds to the simultaneous and immediate firing (due to the zero time bounds) of all the firable Grafcet transitions

- The transformation function h_T is given by three consecutive mappings:

  1) $\forall X_i \in$ ✖ :
  - $X_i = 1$, if $\exists$ t$\in$ S such as $X_i \in X_{FO}(t)$,
  - $X_i = 0$, if ($\nexists$ t$\in$ S such as ($X_i \in X_{FO}(t)$) $\wedge$ ($\exists$t' $\in$ S such as ($X_i \in X_{PR}(t')$))
  2) $\forall$v$\in$ ⮟out, $\forall$t$\in$ S :
  - v=1 if $\exists X_i \in X_{FO}(t)$ such that v$\in$ $action_i$,
  - v=0 if (($\exists X_i \in X_{PR}(t)$ such that v$\in$ $action_i$) and ($\nexists X_j \in X_{FO}(t)$ such that v$\in$ $action_j$)) ,
  - v is not modified , otherwise.
  3) Edge = 0.

The first mapping represents the activation and deactivation of Grafcet steps (according to *rules 3 and 5* of Grafcet), whereas the second mapping sets the actions of the activated (deactivated) Grafcet steps to 1 (to 0). The fact that the internal evolutions take zero time in the external time scale and that delays are associated to the output communicating transitions (Fig. 4) guarantees that the actions updated during those evolutions will not directly influence the plant; only the actions corresponding to a stable situation will be effectively transmitted to the plant. The third mapping resets the variable Edge to reflect the fact that events can only be observed at the instant when they occur and then they immediately disappear.

The first part of table 1 shows the TTM transition set $\Im_G$ corresponding to the possible evolutions of the Grafcet depicted in Fig. 1-a. In this table, a TTM transition labelled Tt_x corresponds to the firing of Grafcet transition t_x. A TTM transition labelled Tt_xt_y corresponds to the simultaneous firing of Grafcet transitions t_x and t_y.

4.2.2 Plant and communicating transitions

The plant can be modelled using extended automata which correspond to a graphical representation of TTMp [13]. Other asynchronous models such as Petri nets can also be used to model the plant since they can be easily translated into TTM. The only imposed constraint within the frame of our approach is that the enabling intervals and the lower time bounds of plant transitions should not be equal to zero (Fig. 4), and that each input variable and action of Grafcet has a corresponding image variable in TTMp.

For a given TTMp to interact with TTM_G, the following communicating transitions are systematically generated:

- $\forall V \in$ ⮟out : create the transition T_V = (true, [v : V], 0, 1) $\in \Im_{out}$, where v $\in$Vp is the image variable of V in the plant.
- $\forall u \in$ ⮟in : create transitions T_{u0} , $T_{u1} \in \Im_{in}$, where: T_{u0} = (U=0$\wedge$u=1, [u:0, Edge:u_{fe}], 0, 1) and T_{u1} = (U=1$\wedge$u=0, [u:1, Edge:u_{re}], 0, 1). The plant's variable U $\in$Vp represents the image of input variable u. T_{u0} is dedicated to the reception of the falling edge of u; it sets u to zero. In a similar way, T_{u1} sets u to 1 upon the occurrence of the rising edge of u. These transitions also update the variable Edge so as to indicate the occurrence of the relevant edge.

The second and third parts of table 1 give the communicating input "$\Im_{in}$" and output "$\Im_{out}$" TTM transitions, respectively, for the Grafcet of Fig. 1-a. For a given system under development, a fourth part must be added to this table to represent the transitions of TTMp.

Transition sets	label	condition	transformation	l	u
$\Im_G$	Tt_1	$X_1 \wedge$Edge = 4	X_1:0, X_2:1, X_6:1, ACT1:1, ACT2:1, ACT3:1, Edge:0	0	0
	Tt_2	$X_2 \wedge (a \vee (c \wedge d)) \wedge \neg (X_2 \wedge (\neg a \wedge \neg b \wedge c) \vee (X_4 \wedge (a \wedge c)) \vee (X_7 \wedge b))$	X_2:0, X_3:1, ACT1:1, ACT2:0, Edge:0	0	0
	Tt_3	$X_2 \wedge (\neg a \wedge \neg b \wedge c) \wedge \neg (X_2 \wedge (a \vee (c \wedge d)))$	X_2:0, X_4:1, ACT1:0, ACT2:0, Edge:0	0	0
	Tt_4	$X_4 \wedge (a \wedge c) \wedge \neg (X_2 \wedge (a \vee (c \wedge d)) \vee (X_7 \wedge b))$	X_4:0, X_5:1, ACT2:1, Edge:0	0	0
	Tt_5	$X_6 \wedge (\neg a \wedge \neg b \wedge \neg c)$	X_6:0, X_7:1, ACT3:0, ACT4:1, Edge:0	0	0
	Tt_6	$X_7 \wedge b \wedge \neg (X_2 \wedge (a \vee (c \wedge d)) \vee (X_4 \wedge (a \wedge c)))$	X_7:0, X_8:1, ACT4:0, Edge:0	0	0

	Tt_7	$X_3 \wedge X_5 \wedge X_8 \wedge Edge=-4$	X_1:1, X_3:0, X_5:0, X_8:0, ACT1:0, ACT2:0, Edge:0	0	0
	Tt_2t_3	$X_2 \wedge (a \vee (c \wedge d)) \wedge X_2 \wedge (\neg a \wedge \neg b \wedge c)$	X_2:0, X_3:1, X_4:1, ACT1:1, ACT2:0, Edge:0	0	0
	Tt_2t_4	$X_2 \wedge (a \vee (c \wedge d)) \wedge X_4 \wedge (a \wedge c) \wedge \neg (X_7 \wedge b)$	X_2:0, X_3:1, X_4:0, X_5:1, ACT1:1, ACT2:1, Edge:0	0	0
	Tt_2t_6	$X_2 \wedge (a \vee (c \wedge d)) \wedge X_7 \wedge b \wedge \neg (X_4 \wedge (a \wedge c))$	X_2: 0, X_3: 1, X_7: 0, X_8: 1, ACT1: 1, ACT2: 0, ACT4: 0, Edge:0	0	0
	Tt_4t_6	$X_4 \wedge (a \wedge c) \wedge X_7 \wedge b \wedge \neg (X_2 \wedge (a \vee (c \wedge d)))$	X_4:0, X_5:1, X_7:0, X_8:1, ACT2:1, ACT4:0, Edge:0	0	0
	$Tt_2t_4t_6$	$X_2 \wedge (a \vee (c \wedge d)) \wedge X_4 \wedge (a \wedge c) \wedge X_7 \wedge b$	X_2:0, X_3:1, X_4:0, X_5:1, X_7:0, X_8:1, ACT1:1, ACT2:1, ACT4:0, Edge:0	0	0
$\mathfrak{I}_{in}$	T_{a0}	$A=0 \wedge a=1$	a :0, Edge :-1	0	1
	T_{a1}	$A=1 \wedge a=0$	a :1, Edge : 1	0	1
	T_{b0}	$B=0 \wedge b=1$	b :0, Edge :-2	0	1
	T_{b1}	$B=1 \wedge b=0$	b :1 Edge : 2	0	1
	T_{c0}	$C=0 \wedge c=1$	c :0 Edge :-3	0	1
	T_{c1}	$C=1 \wedge c=0$	c :1 Edge : 3	0	1
	T_{d0}	$D=0 \wedge d=1$	d :0 Edge :-4	0	1
	T_{d1}	$D=1 \wedge d=0$	d :1 Edge : 4	0	1
$\mathfrak{I}_{out}$	T_{ACT1}	true	act1 : ACT1	0	1
	T_{ACT2}	true	act2 : ACT2	0	1
	T_{ACT3}	true	act3 : ACT3	0	1
	T_{ACT4}	true	act4 : ACT4	0	1

Table 1 : TTM transitions corresponding to Grafcet of Fig. 1-a together with its inputs and outputs

5 Conclusion

Grafcet is a widely used model for the specification and the implementation of sequential controllers. This paper is the first in a series of two papers that propose an integrated approach combining features from Grafcet and the TTM/RTTL formal framework. The objective is to allow users to specify system controller through a user friendly interface given by Grafcet and to provide rapid feedback on system properties for candidate designs. The approach presented in the paper is based on the use of a set of rules for mapping Grafcet logical model specifications into TTM. The mapping rules allow to generate a Timed Transition Model that brings together both the synchronous, reactive and deterministic semantics of Grafcet, and the asynchronous non-deterministic semantics of the plant. Preliminary analysis of Grafcet structure and receptivities enables the mapping function to limit the number of generated TTM transitions. The second paper will present a proof system that is dedicated to the verification of TTM corresponding to Grafcet controlled systems.

References:

[1] Ramadge, P.J. and W.M. Wonham, The control of discrete-event systems, *Proc. IEEE*, Vol.77, 1989, pp. 81-97.

[2] Zaytoon, J. and G. Villermain Lecolier, Two methods for the engineering of manufacturing systems, *Control Engineering Practice*, Vol.5, 1997, pp. 185-198.

[3] David, R., Grafcet: A powerful tool for specification of logic controllers, *IEEE Trans Control Systems Technology*, Vol.3, 1995, pp.253-268.

[4] IEC, *Preparation of function charts for control systems*, International Electrotechnical Commission: Publication 848, 1988.

[5] Arzén, K.E., Grafcet for intelligent supervisory control applications, *Automatica*, Vol.30, 1994, pp.1513-1525.

[6] Lhoste, P., J.M. Faure, J.J. Lesage and J. Zaytoon, Comportement temporel du Grafcet, *European Journal of Automation*, Vol.31, 1997, pp. 675-711 (in French).

[7] Zaytoon, J., J.J. Lesage, L. Marcé, J.M. Faure and P. Lhoste, Vérification et validation du Grafcet, *European Journal of Automation*, Vol.31, 1997, pp. 713-740 (in French).

[8] UTE, *Function charts GRAFCET - extension of basic principles*. Union technique d'Electricité: Publication UTE C03-191, 1993.

[9] Frachet, J.P. and G. Colombari, Elements for a semantics of the time in Grafcet and dynamic systems using non-standard analysis, *Automatic Control Production Systems A.P.I.I.*, Vol.27, 1993, pp. 107-125.

[10] Lhoste, P., H. Panetto and M. Roesch, Grafcet: from syntax to semantics, *Automatic Control Production Systems A.P.I.I.*, Vol.27, 1993, pp. 127-141.

[11] Ostroff, J.S. and W.M. Wonham, A framework for real-time discrete event control, *IEEE Trans Automatic Control*, Vol.35, 1990, pp. 386-397.

[12] Manna, Z. and A. Pnueli, *The temporal logic of reactive and concurrent systems*, Springer, Berlin, 1992.

[13] Ostroff, J.S., *Temporal logic for real time systems*, Wiley, London, 1989.