Business Process Modeling using Petri Nets with Clocks

Daniel Riesco[*], Germán Montejano[*], Gabriel Vilallonga[*], Roberto Uzal[*]

[*] Universidad Nacional de San Luis
Departamento de Informática
Ejército de los Andes 950
5700 – San Luis – Argentina
E-mail: {driesco | gmonte | gvilallo | ruzal}@unsl.edu.ar
Tel: + 54 (0) 2652 – 424027 ext. 251
Fax: + 54 (0) 2652 – 430059

Abstract

Petri Nets are tools for the analysis and design of concurrent systems. There is a formal theory, which supports Petri Nets. We propose Petri Nets with Clocks which has a high expressive power in the concurrent and asynchronous process modeling and gives the possibility to model real time systems.

The Petri Nets with Clocks are useful to model systems with temporal requirements via specification of clocks, using temporal invariants for the places and temporal conditions in the transitions. Also, we have developed an algorithm for the analysis of Petri Nets with Clocks.

For the Business Processes Modeling, we propose to use Petri Nets with Clocks to formalize models, allowing to study the models through a quantitative and qualitative analysis.

Petri Nets with Clocks includes additional temporal elements -clocks-, which are appropriate for the Business Processes Modeling and are not taken into consideration in the literature concerning the extensions of Petri Nets with time. Petri Nets with Clocks allows studying the structural properties of Business Processes Modeling. This study not only allows the simulation but also verifies formally the model. It is oriented to the verification and correction of errors in the modeling of the time variable en Business Processes.

Keywords: Software Engineering, Business Process Modeling, Timed Petri Nets, Real Time Systems, Timed Graph.

1. Introduction

Petri Nets (PN) are tools for system studying and modeling. PN theory allows system modeling, obtaining a mathematical representation of the system [3]. The analysis of the corresponding PN gives important information about the system structure and its dynamic behavior. This information can be used to evaluate the system model and for improving and changing the model.

When studying different types of systems we are confronted with some systems in which time plays an important role: communications protocols, real time systems, business processes, etc. An important part of these system's requirements that must be fulfilled are the temporal requirements. The growing complexity and critical nature of these systems have motivated the search for verification methods [4, 6].

To the power of structure analysis and system behavior that PN have, we have added a structure of an algorithm that allows for temporal analysis of the extended PN [1].

Petri Nets with Clocks (PNwC), proposed here has a high expressive power in the concurrent and asynchronous process modeling and give the possibility to model real time systems. PNwC includes additional temporal elements, clocks, which are not taken into consideration in the literature concerning the extensions of PN with time [6, 16, 17, 18].

Business Process Reengineering has emerged as a modeling and analysis methodology for making the organization of an enterprise more efficient [7, 8, 9].

The Business Process Modeling (BPM) is described as a set of partially ordered steps to reach a business goal [10, 11, 19].

A component of a process is called a process element, representing activities without internal substructures.

An agent is an actor (human or machine) who performs one or more process elements. A coherent set of process elements to be assigned to an agent as a unit of functional responsibility is also called role, and the product created or modified by the execution of a process element is referred to as work. A BMP is an abstract description of a business process constituted by their respective process elements, together with their assignment to agent. The assignment describes the dependencies, coordination and interaction among agents. The most natural way of interaction among agents is the flow of work. Like the assignment of process elements to the responsibility of agent, a BPM must also appropriately describe the input and output relation of work for every process element. The BPM describe work scheduling and management strategy applied by an agent to execute the assigned process element. This is essential especially for agent acting in several roles (involved in more than one process). For a BPM to be adequate with respect to dependencies of processes as determined by their appearance and duration, it must reflect time intervals and duration of process element executions. Consequently, the modeling formalism has to provide sufficient expressiveness to characterize the time dynamics of the system.

The modeling power of PNwC meets those requirements, since they allow formalizing time in the places as well as the transitions.

This work presents an approach to obtain the properties of BPM using PNwC.

2. Petri Nets with Clocks

PN are limited in their modeling and design power when used for systems where time is part of the system's specification. Timed Graphs (TG) [5,12,13, 14] on the other hand are a useful tool to specify system time constraints.

The power inherent of TG and PN has motivated us to extend PN theory with temporal requisites using TG. To the power of structure analysis and system behavior that PN have, we have added a structure of an algorithm that allows the temporal analysis of the extended PN [2].

The definition of PNwC is presented here. By extending PN using TG we profit from the advantage in modeling systems with asynchronous and concurrent behavior (PN) and the possibilities of formalizing systems with temporal requisites.

Definition 1. *A Clock* **x**

A Clock **x** is a positive real variable, i.e., $\mathbf{x} \in \mathbf{R}^{+0} = \{ \mathbf{z} / \mathbf{z} \in \mathbf{R}^{+} \vee \mathbf{z} = \mathbf{0}\}$; where $\mathbf{R}^{+}$ are all real positives.

Definition 2. *Set of all Clocks* ***X***

$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ is the *set of all Clocks*.

Definition 3. *Restricted predicates* Ω_X

Ω_X is a set of restricted predicates on the places defined as a Boolean combination of atoms that take the form **x # c**, where $\mathbf{x} \in \mathbf{X}$, **#** is a binary relation on the set $\{<, >, =, \geq, \leq\}$ and $\mathbf{c} \in \mathbf{R}^{+0}$.

Definition 4. *Restricted predicates* Ψ_X

Ψ_X is a set of restricted predicates on the transitions defined as a Boolean combination of atoms that take the form **c # x # c'** or the form **x # c'** where $\mathbf{x} \in \mathbf{X}$, **#** is a binary relation on the set $\{<, =, \leq\}$, $\mathbf{c}, \mathbf{c'} \in \mathbf{R}^{+0}$. Every clock in Ψ_x must have an upper limit **c'**.

Definition 5. *Valuation* **VaL**

VaL is a set of all vectors of dimension **k**, where **k** is the cardinality of set **X**, and each element of the vector belong to $\mathbf{R}^{+0}$,

$$\mathbf{VaL} = \{\mathbf{v} / \mathbf{v} = (\mathbf{v}_1,\ldots,\mathbf{v}_k) \wedge \mathbf{k} = |\mathbf{X}| \wedge \forall \mathbf{j}, 1 \leq \mathbf{j} \leq \mathbf{k} \cdot \ \mathbf{v}_j \in \mathbf{R}^{+0}\}$$

Definition 6. *A Petri Net with Clocks* **PNwC**

A Petri Net with Clocks is a PN extended based on TG, with a finite set of clocks whose values are incremented uniformly with time. The restrictions associated with the system are expressed by invariants on places and the association of an enabling condition with each transition. A clock can be reset in each transition. Also, the firing of a transition shall be an instantaneous action that does not consume time. Time runs only at places and for no more than what is established by the associated invariant.

Formally the structure of a PNwC is a t-uple:

PNwC = < P, T, I, O, X, Inv, C, A > where:

- **P, T, I** and **O** [3],
- **X** is in **def 2**,

- **Inv:** $\mathbf{P} \rightarrow \Omega$, associates to each place $\mathbf{p_i} \in \mathbf{P}$, a restricted predicate $\Omega \in \Omega_X$ (**def 3**) called **place invariant**.
- **C:** $\mathbf{T} \rightarrow \Psi$, associates to each transition $\mathbf{t} \in \mathbf{T}$, a restricted predicate $\Psi \in \Psi_X$ (**def 4**) called **transition condition.**
- **A:** $\mathbf{T} \rightarrow \mathbf{w}$, set of clocks of the transition that are initialized to zero $\mathbf{w} \subseteq \mathbf{X}$.

Definition 7. *Affectation* α

An affectation α is a function $\alpha\mathbf{:}\ \mathbf{VaL\ x\ T} \rightarrow \mathbf{VaL}$

$$\alpha(\mathbf{v, t}) = \mathbf{v'} \text{ iff } \forall \mathbf{x_i} \in \mathbf{X} \cdot\ [\mathbf{x_i} \in \mathbf{A(t)} \Rightarrow \mathbf{v'(i) = 0} \wedge\ \mathbf{x_i} \notin \mathbf{A(t)} \Rightarrow \mathbf{v'(i) = v(i)}]$$

Definition 8. Marked Timed Petri Net **MPNwC**

A Marked PNwC is defined as **MPNwC = < P, T, I, O, X, Inv, C, A,** μ **>** where **P, T, I, O, X, Inv, C** and **A** are in **def 6**, and $\mu \in \mathbf{M}$.

The marking μ is as an n-vector $\mu = (\mu_1, \mu_2 \ldots \mu_n)$, with $\mathbf{n} = |\ \mathbf{P}\ |$ and $\mu_i \in \mathbb{N}\mathbf{at_0}$ with $\mathbf{1 \leq i \leq n}$.

The set of all marking **M** is the set of all vectors of dimension **n**, $\mathbf{M} \subseteq \mathbf{Nat_0^{n\cdot}}$,

$$\mathbf{M} = \{\mu = (\mu_1, \ldots, \mu_n) \wedge \mathbf{n} = |\ \mathbf{P}\ | \wedge \forall \mathbf{i}, \mathbf{1 \leq i \leq n} \cdot\ \mu_i \in \mathbb{N}\mathbf{at_0}\}$$

Definition 9. *Invariant of the Marking* **InvM(**μ**)**

Invariant of the Marking **InvM(**μ**)** is the conjunction of the invariants of the places where the number of tokens is greater than zero.

$$\mathbf{InvM}(\mu) \text{ iff } \bigwedge \mathbf{Inv(p_i)} \text{ where } \mathbf{p_i} \in \mathbf{P} \wedge \mu(\mathbf{p_i}) > \mathbf{0}$$

Definition 10. *A Predicate applied to Valuation* Φ**[v]**

$\Phi[\mathbf{v}]$ iff $((\mathbf{x_i\ \#\ c})$ is in $\Phi) \Rightarrow \mathbf{v(i)\ \#\ c}$ holds,

where $\mathbf{x_i} \in \mathbf{X}$, **#** is a binary relation on the set $\{<, >, =, \geq, \leq\}$, $\mathbf{c} \in \mathbf{R^{+0}}$.

Definition 11. *A State* **q**

A state of a **PNwC** is a pair $\mathbf{q} = (\mu, \mathbf{v})$, $\mu \in \mathbf{M}$ and $\mathbf{v} \in \mathbf{VaL}$, where the valuation **v** of the clocks satisfy the invariants of the net's places, i.e., **InvM(**μ**)[v]** holds.

Definition 12. *The set of all states* **Q**

The set of all possible states of a PNwC is represented by $\mathbf{Q} \subseteq \mathbf{M\ x\ VaL}$, such that:

$$\mathbf{Q} = \{(\mu, \mathbf{v})\ |\ \mu \in \mathbf{M} \wedge \mathbf{InvM}(\mu)[\mathbf{v}] \text{ holds } \}$$

3. Execution of MPNwC

The execution of the MPNwC is done using the successor marking and the system state changes.

Definition 13. *Enabled Transition in MPNwC* **E(t, q)**

Let $\mathbf{q} = (\mu, \mathbf{v})$ be a possible state of a PNwC, where μ is a marking and **v** the valuation of the clocks. In **q**, the valuation of the clocks satisfies the associated invariants to each place in the marking by **def 11**.

A transition $\mathbf{t} \in \mathbf{T}$ in a **MPNwC** is enabled **E(t, q)** in the state $\mathbf{q} = (\mu, \mathbf{v})$,

$\mathbf{E(t, q)}$ iff $\forall\ \mathbf{p_i} \in \mathbf{I(t)} \cdot\ \mu(\mathbf{p_i}) \geq \#(\mathbf{p_i}, \mathbf{I(p_i)}) \wedge \mathbf{C(t)[v]}$ holds. /* **defs 6, 10 */**

Definition 14.: *System State Changing* $\rightarrow$

The System State Changing is represented by the following expression:

$$\mathbf{SC} = \langle\ \mathbf{Q}, \rightarrow\ \rangle$$

where **Q** is the set of all states (**def 19**).

The changing relation $\rightarrow\ \subseteq \mathbf{Q}\ \mathbf{x}\ (\mathbf{T} \cup \mathbf{R}^{+})\ \mathbf{x}\ \mathbf{Q}$ has two types of changing: temporals and instantaneous. The notation is $\mathbf{q} \rightarrow^{\mathbf{time}} \mathbf{q'}$ for temporal changing and $\mathbf{q} \rightarrow^{\mathbf{t}} \mathbf{q'}$ for instantaneous changing, where $\mathbf{q}, \mathbf{q'} \in \mathbf{Q}$ and $\mathbf{time}, \mathbf{t} \in \mathbf{T} \cup \mathbf{R}^{+}$.

Definition 15.: *Temporal State Changing* $\rightarrow^{\mathbf{time}}$

Temporal state changing represents the elapse of time by a changing labeled **time** from the state $(\mu, \mathbf{v})$ to $(\mu, \mathbf{v+time})$, $(\mu, \mathbf{v}) \in \mathbf{Q}$, $\mathbf{time} \in \mathbf{R}^{+0}$.

$(\mu, \mathbf{v}) \rightarrow^{\mathbf{time}} (\mu, \mathbf{v+time})$ iff $\forall \mathbf{y}, \mathbf{y} \in \mathbf{R}^{+0} \cdot\ 0 \leq \mathbf{y} \leq \mathbf{time} \Rightarrow \mathbf{InvM}(\mu)[\mathbf{v+y}]$ holds

/* def 9 */

Definition 16.: *Instantaneous or discrete State Changing (by transition)* $\rightarrow^{\mathbf{t}}$

An instantaneous state changing is given by the execution of a transition $\mathbf{t} \in \mathbf{T}$, where the changing is labeled **t,** from the state $(\mu, \mathbf{v})$ to state $(\mu', \mathbf{v'})$ as **def 14.**

$(\mu, \mathbf{v}) \rightarrow^{\mathbf{t}} (\mu', \mathbf{v'})$ iff $\mathbf{E}(\mathbf{t}, (\mu, \mathbf{v})) \wedge \alpha(\mathbf{v}, \mathbf{t}) = \mathbf{v'}$ **/* defs 7, 14 */**

4. Analysis Method

Analysis of a PNwC is based on the exploration of the symbolic execution of the system being analyzed. This execution is studied using the PN reachability graph. When the reachability graph is constructed each graph node represents a symbolic state. The following definitions are necessary for the formalizing of the process model that employs PNwC. The analysis algorithm allows model checking to detect errors in the structure as well as modeling of the time variable. In contrast with the algorithm described in [20] this algorithm allows the analysis of concurrent and asynchronous process as well as the time variable.

4.1. Symbolic Execution of PNwC

A symbolic state of a PNwC is a generalization of the concept of state. The state is a pair $(\mu, \mathbf{v})$ as in **def 11.** The symbolic state is a pair $\langle \mu, \Omega \rangle$ as in **def 17.** The difference between state and symbolic state is that state uses a valuation and the symbolic state uses a predicate $\mathbf{\Omega}$.

Definition 17. *Symbolic States* $\langle \mu, \Omega \rangle$

The analysis verification technique that follows are based on symbolic execution for a given set of symbolic states [14].

A symbolic state of a PNwC is a pair $\langle \mu, \Omega \rangle$ where $\mu \in \mathbf{M}$, and $\Omega \in \Omega_{\mathrm{X}}$.

Definition 18.: *Set of all States* $\mathbf{|[\, \langle \mu, \Omega \rangle \,]|}$

$\mathbf{|[\, \langle \mu, \Omega \rangle \,]|}$ is the set of all states $\mathbf{(\mu, v)}$ of the symbolic state $\langle \mu, \Omega \rangle$ such that $\mathbf{\Omega[v]}$ holds.
/* **defs 10, 11** */

Definition 19.: *Temporal Successors of Symbolic States* $\langle \mu, \mathbf{succ_{time}^{\mu}(\Omega)} \rangle$

The temporal successors of symbolic state that are in $\mathbf{|[\, \langle \mu, \Omega \rangle \,]|}$ are characterized by the symbolic state $\langle \mu, \mathbf{succ_{time}^{\mu}(\Omega)} \rangle$, where

$$\mathbf{(\mu, v)} \in |[\, \langle \mu, \mathbf{succ_{time}^{\mu}(\Omega)} \rangle \,]| \text{ iff } \mathbf{(\mu, v\text{-}time)} \rightarrow^{\mathbf{time}} \mathbf{(\mu, v)} \wedge \mathbf{\Omega[v\text{-}time]} \text{ holds.}$$

/* **defs 15, 10** */

Definition 20.: *Successors of Symbolic States by a Transition* $\langle \mu', \mathbf{succ_t(\Omega)} \rangle$

The successors of symbolic state by a transition **t** from $\langle \mu, \Omega \rangle$ to $\langle \mu', \mathbf{succ_t(\Omega)} \rangle$ are:

$$\mathbf{(\mu', v')} \in |[\, \langle \mu', \mathbf{succ_t(\Omega)} \rangle \,]| \text{ iff } \exists\, \mathbf{(\mu, v)} \in |[\, \langle \mu, \Omega \rangle \,]| \cdot \ \mathbf{(\mu, v)} \rightarrow^{\mathbf{t}} \mathbf{(\mu', v')}$$

/* def 16 */

Definition 21.: *Symbolic Sequence* $\langle \mu_0, \Omega_0 \rangle\ \mathbf{t_0}\ \langle \mu_1, \Omega_1 \rangle\ \mathbf{t_1}\ \langle \mu_2, \Omega_2 \rangle \ldots$

The analysis of PNwC is done exploring sequences of symbolic states. We define the symbolic sequences as follows.

A symbolic sequence with start $\langle \mu_0, \Omega_0 \rangle$ is a succession of symbolic states:

$$\langle \mu_0, \Omega_0 \rangle\ \mathbf{t_0}\ \langle \mu_1, \Omega_1 \rangle\ \mathbf{t_1}\ \langle \mu_2, \Omega_2 \rangle \ldots$$

where μ_0 is the initial marking, Ω_0 are the all clocks initialized in zero and

$$\langle \mu_{i+1}, \Omega_{i+1} \rangle = \langle \mathbf{succM}(\mu_i, \mathbf{t}), \mathbf{succ_{ti}}(\mathbf{succ_{time}^{\mu\, i}}(\Omega_i)) \rangle$$

Definition 22.: *Symbolic Reachability Graph*

The Reachability Graph (RG) of a PNwC is composed of sequences which are constructed from the initial state $\langle \mu_0, \Omega_0 \rangle$. Let $\langle \mu_i, \Omega_i \rangle$ be any symbolic state, if $\forall \mathbf{t_{jr}} \in \mathbf{T}, \mathbf{1} \le \mathbf{r} \le \mathbf{k} \cdot\ \mathbf{E}(\mu_i, \mathbf{t_{jr}})$ then the symbolic RG will be composed of the symbolic sequences with the following characteristics:

$$\langle \mu_i, \Omega_i \rangle\ \mathbf{t_1}\ \langle \mu_{j1}, \Omega_{j1} \rangle$$

$$\ldots$$

$$\langle \mu_i, \Omega_i \rangle\ \mathbf{t_k}\ \langle \mu_{jk}, \Omega_{jk} \rangle$$

where **k** is the number of enabled transitions in $\langle \mu_i, \Omega_i \rangle$, and where **k** new sequences are generated that have the particularity that up to the symbolic state $\langle \mu_i, \Omega_i \rangle$ of the sequences they are all equal, and from then on they might be different.

5. Modeling Business Process System with PNwC

In [15] the timed Petri nets has been vital for the preanalysis of the structural properties in the decomposition of BPM for improve a distributed simulation methods to accelerate the evaluation. Here we present a method for model analysis and verification that goes beyond simulation.

In a correspondence among PNwC and business process concepts, the static structure of the business process topology is expressed in a PNwC graph, whereas the behavioral aspect is encoded in the initial marking and the transition firing rules of the PNwC. The relationship between PNwC and BPM respect of theirs main concepts can more formally be summarized as follows: A BPM is a MPNwC **/* def. 8 */**

BPM = **<S, X, Inv, C, A, μ_0>**, where:

- S is the structure of standard PN, **S = <P, T, I, O>**. P is the set of places, corresponds to "work deposits", stores for pieces of work represented by tokens.
  T is the set of transitions T = {t1, t2, …} stands for process elements, i.e. abstract, atomic processes to be conducted upon invocation (enabling). Each transition has associated a condition of firing. Transition t_i don't consume time for their enabling. The transition t_i is also refer as process elements. Work is seen as a product created or modified by the enactment of a process element, i.e. the firing of a transition. A process element may require a resource or a set of resources for their execution; it may or may not consume time.
  The functions I and O model the flow or work among process elements.
- **X** is the set of clocks **/* def 2** */ represent all the variables what models the time in the systems. Each element of X can model time restriction in invariants and conditions.
- **Inv: P** $\rightarrow$ **Ω**, /* **def 6** */, the work deposit's invariant, that stand how many time may stay a work, represented by tokens, in a work deposit.
- **C: T** $\rightarrow$ **Ψ**, /* **def 6** */, the process element's condition, that stands the enabling interval time. Establishes what is the necessary time for a job to pass to another deposit.
- **A: T** $\rightarrow$ **w**, **/* def 6 */**, the process element's affectation, stand the new values of clocks, a new initialization of the set of clocks. The affectation reflects the end of a task mensuration and the beginning of another one. Clocks are reset to be able to time a new set of activities.
- μ_0 is the initial assignment of works to work deposits.

6. Dynamic behavior

Analogously, we describe the rules for the definition of the dynamic behavior of BMP with PNwC.

Process element enabling rule: A process element t_i is enabled, **E(t_i, q),** in some state **q = (μ, v)**, iff each of its input deposits contains a "sufficient" amount of work, and the temporal restriction associated to the process holds **/*def 13*/.**

Process element execution rule: When a process element $t_i \in$ T executes in the state **(μ, v)** produce a state changing to **(μ', v')**, **/*def 16*/.** That means changes in the marking removing a certain amount of work from its input deposits and creating a certain amount into its output deposits, **μ'**, and the new valuation **v'**.

A business process, or equivalently a process, finally appears as a set of partially ordered process elements, described as a spatial region of a PNwC graph.

7. Conclusions

We have presented here a formalism to model systems with time restrictions. We believe that is very important the verification of the system's integrity regarding its structure and its temporal specifications. Deadlock detection and temporal blockings as well as the consistency of the restrictions have been proposed in previous work.

PNwC has a high expressive power in the concurrent and asynchronous process modeling and have the possibility to model real time systems. PNwC includes additional temporal elements, clocks, which are not taken into consideration in the literature concerning the extensions of PN with time. We have developed:

- a formalism for PNwC,
- a CASE tool, supported by the above formalism, that allows the design and verification of concurrent and asynchrony processes.

PN theory has been used for developing the PNwC formalism, including the analysis. On the other hand, TG theory has been used for the temporal aspects of PNwC.

The defined theoretical basis allowed developing a CASE tool, which models, verifies and corrects automatically errors in the modeling of PNwC time variable. The tool generates information about temporal unreachable state and process deadlocks with temporal blocks. Also, it corrects places invariants and transitions conditions.

Modeling and analysis of business organization is required for the purpose of redesigning an enterprise's business process to make the organization more effective (Business Process Reengineering), as well as for the purpose of establishing advanced coordination technology in the organization. Here is presented an abstract frame of business process systems and a modeling formalism based on PNwC. This formalism allows the quantitative and qualitative analysis of BPM.

References

[1] G. Montejano, D. Riesco, G. Vilallonga, A. Dasso, L. Favre "An Analysis Algorithm for Timed Petri Nets" Software Engineering (SE'98). International Association of Science and Technology for Development. Las Vegas, USA. 1998.

[2] D. Riesco, G. Montejano, G. Vilallonga, A. Dasso, R. Uzal. "Underlying Formalism for a Timed Petri Net Algorithm", IASTED International Conference Software Engineering and Applications, Octubre de 1999, Scottsdale, Arizona, USA.

[3] J. Peterson, Petri Net Theory and the Modelling of Systems. Prentice Hall, Englewood Cliffs, New Jersey, 1981.

[4] Alur, Courcoubetis and Dill. Model checking for real-time systems, in "Proceedings of the Firth Symposium on Logic in Computer Science", p. 414-425, IEEE C. S. P., Los Alamitos, CA.

[5] Sifakis and Yovine. Compositional Specifications of Timed Systems. STACS '96.

[6] Ghezzi, Mandrioli, Morascas and Pezze -1991 "A unified high-level Petri Net model for time critical system". IEEE Transactions on Software Ingeniering 17(2), 160-172.

[7] Hammer, M, "Beyond Reengineering: How the process-centered organization is changing out work and our lives", Harper Collins, 1996.

[8] Hammer, M. and Champy, J. "Reengineering the Corporation: A Manifesto for Business Revolution", Harper Collins Publishing, Inc., 1993

[9] Jacobson, I. "Objectifying Business Process Reengineering", Addison Wesley, 1996

[10] Hollingsworth, D. Workflow Management Coalition. The Workflow Reference Model. Document Number TC00-1003. Issue 1.1. Jan-95. http://www.wfmc.org

[11] Jacobson, I, Booch, G., Rumbaugh, J, "The Unified Software Development Process", Addison-Wesley, 1999.

[12] Olivero. Modélisation et analyse de systémes temporisés et hybrides. Thése, Institut National Polytechnique de Grenoble, France, September 1994.

[13] Yovine, S. Méthodes et outils pour la vérification symbolique de systémes temporises, Phd thesis, Institut National Polytechnique de Grenoble, France, May 1993.
[14] Daws, A. Olivero, S. Tripakis, S. Yovine. The tool Kronos. VERIMAG associated with IMAG. 1996.
[15] A. Ferscha. “Optimistic Distributed Execution of Business Process Models”. 31st Hawaii International Conference on System Sciences (HICSS'98), published by the IEEE.
[16] M. Zuberek. Timed Petri nets: Definitions, Properties, and applications. Microelectronics and Reliability, 31(4):627-644, 1991.
[17] Ghezzi, Mandrioli, Morascas and Pezze -1989 “A general way to put time in Petri nets”. Proceeding 5th International Workshop on Software Specification and Design, Pittsburgh, May 89, pages 60-67, IEEE Computer Society Press, 1989.
[18] Buy and Sloan. Analysis of Real-Time Programs with Simple Time Petri Nets. ACM 1994.
[19] Aguilar, Marc; Rautert, Tankred; Pater, Alexander; "Business Process Simulation: A Fundamental Step Supporting Process Centered Management", Proceeding of the 1999 IEEE, Winter Simulation Conference.
[20] Henzinger, Nicollin, Sifakis and Yovine. “Symbolic model checking for real-time systems”, Information and Computation, 111(2):193-244, 1994