

**BUSINESS PROCESS MODELING AND VALIDATION
USING DECISION PROCESS PETRI NETS**

Zvi Retchkiman Konigsberg

Instituto Politécnico Nacional

Centro de Investigación en Computación – CIC

Mineria 17-2, Col. Escandon

Mexico D.F. 11800, MEXICO

e-mail: mzvi@cic.ipn.mx

Abstract: This paper presents a formal framework for business process modeling and validation using partially ordered transition Decision Process Petri Nets (DPPN). The advantage of this approach is its ability to represent the dynamic behavior of the business process. The business process model is supported by an information technology strategic planning (ITSP) model and methodology. The modeling methodology is based on business strategy transformation. High-level business strategies are refined up to the point when they can be described in terms of the activities needed to achieve a certain tactical business strategy given only in terms of goals and strategies. At this point, partially ordered Petri nets are used for business process representation and analysis. DPPN extends the place-transitions Petri nets theoretic approach including the Markov decision process. In this sense, DPPN corresponds to a series of strategies which guide the selection of actions that lead to a final (decision) state. By taking into account different possible courses of action the overall utility of each strategy is considered. The utility function of each business process is represented by a Lyapunov like function. Conditions of equilibrium and stability for the DPPN are recalled. For illustration purposes, two examples are presented.

AMS Subject Classification: 91B06, 93D05, 39A05, 39A11, 62C99

Key Words: business process, information technology strategic planning (ITSP), decision process, Decision Process Petri Nets, strategies and goals, Lyapunov

1. Introduction

Company's success depends on the ability to evolve with the market, not just respond to it. In response to the competitive pressures enforced by the customer demands and the constant changes on the conditions of the environment, many companies are re-thinking the way they do business [3]. The environmental turbulence has created a need for dynamic business processes and companies are looking for models that can evolve and adapt efficiently business processes to the changing conditions and the changing business strategies. As a consequence, research interest in the business process modeling area has increased dramatically over the past decades.

At present, there is a modest formal model support for business processes and it is still done informally. The most critical point in the development of a business process depends largely on the ability to choose a conceptual model to represent the problem domain in a coherent and natural fashion. Formal models that capture and organize knowledge about a business environment can facilitate solutions to this problem.

For instance, in the area of requests processing, government offices are not satisfied fulfilling the well established process. Instead, they are taken as a fundamental job, the re-design of the requests handling process. In this sense, the government offices want to minimize the citizens request responding costs (for obvious budget problems). At the same time, the government offices would like to keep citizens happy, in order to continue sustaining its operational or political position. Therefore, from the business process point of view it is important to be able to describe these operational relationships, and to model solutions from strategic perspectives.

This paper introduces a modeling paradigm for developing business process representation via partially ordered transition Decision Process Petri Nets (DPPN)). It is supported by an ITSP model and methodology, which integrates the business/organizational strategic visions and the information technology (IT) strategic vision in a resulting unified vision (related works in ITSP model are presented in [5] and [1]).

The method is based in business strategy decomposition. High-level business strategies are refined up to the point when they reach a tactical business strategy level, described only in terms of goals and strategies. The importance of being able to clearly link the business processes with the business strategy is highlighted by the concept of business reengineering [3]. The notion of business strategy decomposition is adopted to represent the process of business strategy refinement. Activities are considered as operationalizations of goals and are

applied in accordance with the strategies needed to achieve these goals. Thus, the decomposition process results in a set of primitive actions such as “order a product”. Strategies are expressions that define valid state transitions in the business process. In fact, strategies specify the event occurrences and they represent either integrity rules or control operations. Since the business strategy decomposition determines actions sequence applications, a process can be ordered introducing a partial ordered relation. It is important to note that any business process ultimately ends, because real processes are finite. The method considers a dynamic application domain, since the organizational model is able to modify its structure and respond appropriately to the changes in the business strategy.

The ITSP model considers a dynamic application environment, which integrates the business/organizational strategic visions and the IT strategic vision in a resulting unified vision. Its conceptualization is based in three fundamental concepts: interaction, adaptation and evolution. The ITSP methodology is organized in fifteen modules one of them being the business process.

Partially ordered transition Decision Process Petri Nets are used for business process representation, taking advantage of the well-know properties of this approach namely, formal semantic, graphical display and wide acceptance by practitioners. A partially ordered transition Decision Process Petri Nets model of a business process gives a specific and unambiguous description of the behavior of the process. Its solid mathematical foundation has resulted in different analysis methods and tools. Despite of the formal background, Petri net models are easy to understand.

The rest of the paper is structured in the following manner. The next section presents the necessary mathematical background and terminology needed to understand the rest of the paper. Section 3 describes the basic formalism of the ITSP model and the methodology. Section 4 discusses the issues associated to the business process model method. Section 5 presents two application examples. Finally, Section 6 concludes the paper by giving future research directions.

2. Preliminaries

2.1. Petri Nets (see [8])

Petri nets are a tool for the study of systems. Petri net theory allows a system to

be modeled by a Petri net, a mathematical representation of the system. Analysis of the Petri net then, can hopefully, reveal important information about the structure and dynamic behavior of the modeled system. This information can then be used to evaluate the modeled system and suggest improvements or changes.

A Petri net is a 5-tuple, $PN = \{P, Q, F, W, M_0\}$, where: $P = \{p_1, p_2, \dots, p_m\}$ is a finite set of places, $Q = \{q_1, q_2, \dots, q_n\}$ is a finite set of transitions, $F \subset (P \times Q) \cup (Q \times P)$ is a set of arcs, $W : F \rightarrow N_1^+$ is a weight function, $M_0 : P \rightarrow N$ is the initial marking, $P \cap Q = \emptyset$ and $P \cup Q \neq \emptyset$.

A Petri net structure without any specific initial marking is denoted by PN . A Petri net with the given initial marking is denoted by (PN, M_0) . Notice that if $W(p, q) = \alpha$ (or $W(q, p) = \beta$) then, this is often represented graphically by α , (β) arcs from p to q (q to p) each with no numeric label.

Let $M_k(p_i)$ denote the marking (i.e., the number of tokens) at place $p_i \in P$ at time k and let $M_k = [M_k(p_1), \dots, M_k(p_m)]^T$ denote the marking (state) of PN at time k . A transition $q_j \in Q$ is said to be enabled at time k if $M_k(p_i) \geq W(p_i, q_j)$ for all $p_i \in P$ such that $(p_i, q_j) \in F$. It is assumed that at each time k there exists at least one transition to fire, i.e. it is not possible to block the net. If a transition is enabled then, it can fire. If an enabled transition $q_j \in Q$ fires at time k then, the next marking for $p_i \in P$ is given by

$$M_{k+1}(p_i) = M_k(p_i) + W(q_j, p_i) - W(p_i, q_j).$$

Let $A = [a_{ij}]$ denote an $n \times m$ matrix of integers (the incidence matrix), where $a_{ij} = a_{ij}^+ - a_{ij}^-$ with $a_{ij}^+ = W(q_i, p_j)$ and $a_{ij}^- = W(p_j, q_i)$. Let $u_k \in \{0, 1\}^n$ denote a firing vector, where if $q_j \in Q$ is fired then, its corresponding firing vector is $u_k = [0, \dots, 0, 1, 0, \dots, 0]^T$ with the one in the j -th position in the vector and zeros everywhere else. The matrix equation (nonlinear difference equation) describing the dynamical behavior represented by a Petri net is:

$$M_{k+1} = M_k + A^T u_k, \quad (1)$$

where if at step k , $a_{ij}^- < M_k(p_j)$ for all $p_j \in P$ then, $q_i \in Q$ is enabled and if this $q_i \in Q$ fires then, its corresponding firing vector u_k is utilized in the difference equation (1) to generate the next step. Notice that if M' can be reached from some other marking M and, if we fire some sequence of d transitions with corresponding firing vectors u_0, u_1, \dots, u_{d-1} we obtain that

$$M' = M + A^T u, \quad u = \sum_{k=0}^{d-1} u_k. \quad (2)$$



Figure 1: Routing policy Case 1 Figure 2. Routing policy Case 2

Definition 1. The set of all the markings (states) reachable from some starting marking M is called the reachability set, and is denoted by $R(M)$.

Let (N^n, d) be a metric space where $d : N^n \times N^n \rightarrow \mathbb{R}_+$ is defined by

$$d(M_1, M_2) = \sum_{i=1}^m \zeta_i | M_1(p_i) - M_2(p_i) |; \zeta_i > 0, \quad i = 1, \dots, m$$

and consider the matrix difference equation which describes the dynamical behavior of the discrete event system modeled by a Petri net (2) then we have the following proposition.

Proposition 2. Let PN be a Petri net. PN is uniform practical stable if there exists a Φ strictly positive m vector such that

$$\Delta v = u^T A \Phi \leq 0 \Leftrightarrow A \Phi \leq 0. \tag{3}$$

2.2. Decision Processes Petri Nets, see [9]

Definition 3. A Decision Process Petri Net is a 7-tuple $DPPN = \{P, Q, F, W, M_0, \pi, U\}$, where:

- $P = \{p_0, p_1, p_2, \dots, p_m\}$ is a finite set of places,
- $Q = \{q_1, q_2, \dots, q_n\}$ is a finite set of transitions,
- $F \subset I \cup O$ is a set of arcs, where $I \subset (P \times Q)$ and $O \subset (Q \times P)$ such that $P \cap Q = \emptyset$ and $P \cup Q \neq \emptyset$,
- $W : F \rightarrow N_1^+$ is a weight function,
- $M_0: P \rightarrow N$ is the initial marking,
- $\pi : I \rightarrow \mathbb{R}_+$ is a routing policy representing the probability of choosing a particular transition (routing arc), such that for each $p \in P$,

$$\sum_{(p,q_j):q_j \text{ varying over } Q} \pi((p, q_j)) = 1,$$

- $U : P \rightarrow \mathbb{R}_+$ is a utility function.

Figure 1 and Figure 2 represent partial routing policies π that generates a transition from state p_1 to state p_2 where $p_1, p_2 \in P$:

Case 1. In Figure 1 the probability that q_1 generates a transition from state p_1 to p_2 is $1/3$. But, because q_1 transition to state p_2 has two arcs, the probability to generate a transition from state p_1 to p_2 is increased to $2/3$.

Case 2. In Figure 2 we set by convention for the probability that q_1 generates a transition from state p_1 to p_2 is $1/3$ ($1/6$ plus $1/6$). However, because q_1 transition to state p_2 has only one arc, the probability to generate a transition from state p_1 to p_2 is decreased to $1/6$.

Definition 4. Let $L : \mathbb{R}^n \rightarrow \mathbb{R}_+$ be a continuous map. Then, L is a Lyapunov like function iff satisfies the following properties:

1. $\exists x^*$ such that $L(x^*) = 0$,
2. $L(x) > 0$ for $\forall x \neq x^*$,
3. $L(x) \rightarrow \infty$ when $x \rightarrow \infty$,
4. $\Delta L = L(x_{i+1}) - L(x_i) < 0$ for all $x_i, x_{i+1} \neq x^*$.

Definition 5. The utility function U with respect a Decision Process Petri Net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is represented by the equation

$$U_k^{q_j}(p_i) = \begin{cases} U_k(p_0), & \text{if } i = 0, k = 0, \\ L(\alpha), & \text{if } i > 0, k = 0 \text{ and } i \geq 0, k > 0, \end{cases} \quad (4)$$

where

$$\alpha = [\sum_{h \in \eta_{i j_0}} \Psi(p_h, q_{j_0}, p_i) * U_k^{q_{j_0}}(p_h), \sum_{h \in \eta_{i j_1}} \Psi(p_h, q_{j_1}, p_i) * U_k^{q_{j_1}}(p_h) \\ , \dots, \sum_{h \in \eta_{i j_f}} \Psi(p_h, q_{j_f}, p_i) * U_k^{q_{j_f}}(p_h)] \quad (5)$$

the function $L : D \subseteq \mathbb{R}_+^n \rightarrow \mathbb{R}_+$ is a Lyapunov like function which optimizes the utility through all possible transitions (i.e. trough all the possible trajectories defined by the different q_j s), D is the decision set formed by the j 's ; $0 \leq j \leq f$ of all those possible transitions $(q_j p_i) \in O$, $\Psi(p_h, q_j, p_i) = \pi(p_h, q_j) * \frac{FN(q_j, p_i)}{FN(p_h, q_j)}$, $\eta_{i j}$ is the index sequence of the list of previous places to p_i through transition q_j , p_h ($h \in \eta_{i j}$) is a specific previous place of p_i through transition q_j .

2.2.1. DPPN Mark-Dynamic Properties

Definition 6. An equilibrium point with respect a Decision Process Petri Net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is a place $p^* \in P$ such that $M_l(p^*) = S < \infty$, $\forall l \geq k$ and p^* is the last place of the net.

Theorem 7. *The Decision Process Petri Net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is uniformly practically stable iff there exists a Φ strictly positive m vector such that $\Delta v = u^T A \Phi \leq 0$.*

2.2.2. DPPN Trajectory-Dynamic Properties

Definition 8. A final decision point $p_f \in P$ with respect a Decision Process Petri Net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is a place $p \in P$, where the infimum or the minimum is attained, i.e. $U(p) = 0$ or $U(p) = C$.

Definition 9. An optimum point $p^\Delta \in P$ with respect a Decision Process Petri Net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is a final decision point $p_f \in P$, where the best choice is selected “according to some criteria”.

Proposition 10. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri Net and let $p^\Delta \in P$ an optimum point. Then $U(p^\Delta) \leq U(p)$, $\forall p \in P$ such that $p \leq_U p^\Delta$.*

Theorem 11. *The Decision Process Petri Net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is uniformly practically stable iff $U(p_{i+1}) - U(p_i) \leq 0$.*

Definition 12. A strategy with respect a Decision Process Petri Net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is identified by σ and consists of the routing policy transition sequence represented in the $DPPN$ graph model such that some point $p \in P$ is reached.

Definition 13. An optimum strategy with respect a Decision Process Petri Net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is identified by σ^Δ and consists of the routing policy transition sequence represented in the $DPPN$ graph model such that an optimum point $p^\Delta \in P$ is reached.

2.2.3. Convergence of the DPPN Mark-Dynamic and Trajectory-Dynamic Properties

Theorem 14. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri Net. If $p^* \in P$ is an equilibrium point then it is a final decision point.*

Theorem 15. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a finite and non-blocking Decision Process Petri Net (unless $p \in P$ is an equilibrium point). If $p_f \in P$ is a final decision point then it is an equilibrium point.*

Corollary 16. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a finite and non-blocking Decision Process Petri Net (unless $p \in P$ is an equilibrium point).*

Then, an optimum point $p^\Delta \in P$ is an equilibrium point.

Definition 17. Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri Net. A trajectory ω is an (finite or infinite) ordered subsequence of places $p_{\varsigma(1)} \leq_{U_k} p_{\varsigma(2)} \leq_{U_k} \dots \leq_{U_k} p_{\varsigma(n)} \leq_{U_k} \dots$ such that a given strategy σ holds.

Definition 18. Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri Net. An optimum trajectory ω is an (finite or infinite) ordered subsequence of places $p_{\varsigma(1)} \leq_{U_k^\Delta} p_{\varsigma(2)} \leq_{U_k^\Delta} \dots \leq_{U_k^\Delta} p_{\varsigma(n)} \leq_{U_k^\Delta} \dots$ such that the optimum strategy σ^Δ holds.

Theorem 19. Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a non-blocking Decision Process Petri Net (unless $p \in P$ is an equilibrium point) then we have that:

$$U_k^\Delta(p^\Delta) \leq U_k(p), \quad \forall \sigma, \sigma^\Delta.$$

Corollary 20. Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a non blocking Decision Process Petri Net (unless $p \in P$ is an equilibrium point) and let σ^Δ an optimum strategy. Set $L = \min_{i=1, \dots, |\alpha|} \{\alpha_i\}$ then, $U_k^\Delta(p)$ is equal to:

$\sigma_{0j_m}^\Delta(p_{\varsigma(0)})$	$\sigma_{1j_m}^\Delta(p_{\varsigma(0)})$	\dots	$\sigma_{nj_m}^\Delta(p_{\varsigma(0)})$	$U_k(p_0)$
$\sigma_{0j_n}^\Delta(p_{\varsigma(1)})$	$\sigma_{1j_n}^\Delta(p_{\varsigma(1)})$	\dots	$\sigma_{nj_n}^\Delta(p_{\varsigma(1)})$	$U_k(p_1)$
\dots	\dots	\dots	\dots	\dots
$\sigma_{0j_v}^\Delta(p_{\varsigma(i)})$	$\sigma_{1j_v}^\Delta(p_{\varsigma(i)})$	\dots	$\sigma_{nj_v}^\Delta(p_{\varsigma(i)})$	$U_k(p_i)$
\dots	\dots	\dots	\dots	\dots

$\underbrace{\hspace{15em}}_{\sigma^\Delta} \qquad \underbrace{\hspace{2em}}_U$

(6)

where p is a vector whose elements are those places which belong to the optimum trajectory ω given by $p_0 \leq p_{\varsigma(1)} \leq_{U_k} p_{\varsigma(2)} \leq_{U_k} \dots \leq_{U_k} p_{\varsigma(n)} \leq_{U_k} \dots$ which converges to p^Δ .

Definition 21. A Decision Process Petri Net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is said to be symmetric if it is possible to decompose it into some finite number (greater than 1) of sub-Petri nets in such a way that there exists a bijection ψ between all the sub-Petri nets such that

$$(p, q) \in I \Leftrightarrow (\psi(p), \psi(q)) \in I \text{ and } (q, p) \in O \Leftrightarrow (\psi(q), \psi(p)) \in O$$

for all of the sub-Petri nets.

Corollary 22. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a non blocking (unless p is an equilibrium point) symmetric Decision Process Petri Net and let σ^Δ be an optimum strategy. Set $L = \min_{i=1, \dots, |\alpha|} \{\alpha_i\}$ then*

$$\sigma^\Delta U \leq \sigma U, \quad \forall \sigma, \sigma^\Delta,$$

where the σ and σ^Δ are represented by a matrix and U is represented by a vector.

2.2.4. Optimum Trajectory Planning

Given a non-blocking (unless $p \in P$ is an equilibrium point) Decision Process Petri Net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$, the optimum trajectory planning consists in finding the firing transition sequence u such that the optimum target state M_t with the optimum point is achieved. The target state M_t belong to the reachability set $R(M_0)$, and satisfies that it is the last and final task processed by the DPPN with some fixed starting state M_0 with utility U_0 .

Theorem 23. *The optimum trajectory planning problem is solvable.*

3. ITSP Conceptual Model and Methodology, see [2]

In the model represented in Figure 3, the real world is composed by entities representing physical things (people, governments, enterprises, etc.) these entities are related in terms of goals, beliefs, etc. Entities under events generation change the environmental conditions. They take particular strategic positions through the network of relationships with other entities, where they play different roles. The model is based on three fundamental concepts: interaction, adaptation and evolution.

The interaction concept represents the dynamic behavior of the environment, leading to the incorporation or rejection of beliefs and facts related with environment conditions. Interactions are established by the relationships between the roles that each entity plays in the domain of application. The behavior of the environment is induced by the interaction of the entities.

When an incident happens (beliefs, market reactions, etc.), and it changes the environment conditions, it is called an event. Each entity has the option to consider an event occurrence and it incorporates or rejects the facts related to changes in the environment. The acceptance or rejection will depend on the entities interest. Some examples of conditions that can be accepted are:

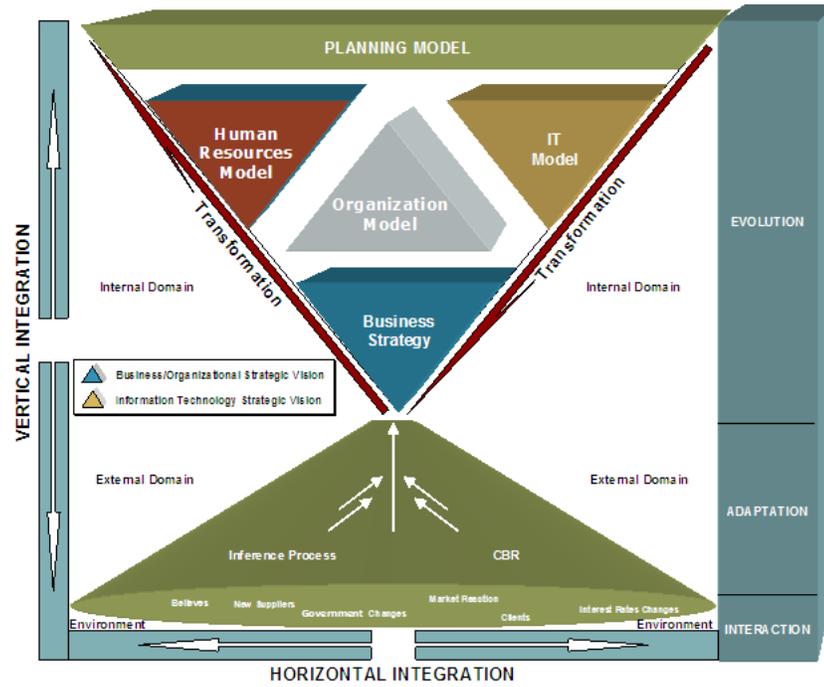


Figure 3: ITSP Model

economic plans changes, political beliefs, new technological tendencies, interest rate grow, etc.

The adaptation includes business strategies using a logic inference method, which uses beliefs and facts in order to generate new business strategies. This is a dynamic process where old business strategies are replaced by those corresponding with the present environmental state. In the real world, there are always assumptions that, if they are proven to be unfounded, they are easily corrected. The environmental changes always take place in the course of events that invalidate previous states. On the other hand, non-monotonic reasoning shows an opposite fact to this problem. It simply allows the retraction of “truth” whenever contradictions arise by forcing the incorporation of new beliefs.

The evolution is a process in which the business strategy is transformed into operative and IT components (the organizational model, the human resources, the IT model and the planning model). It considers a dynamic application domain which integrates the business/organizational strategic visions and the IT strategic vision in a resulting unified vision.

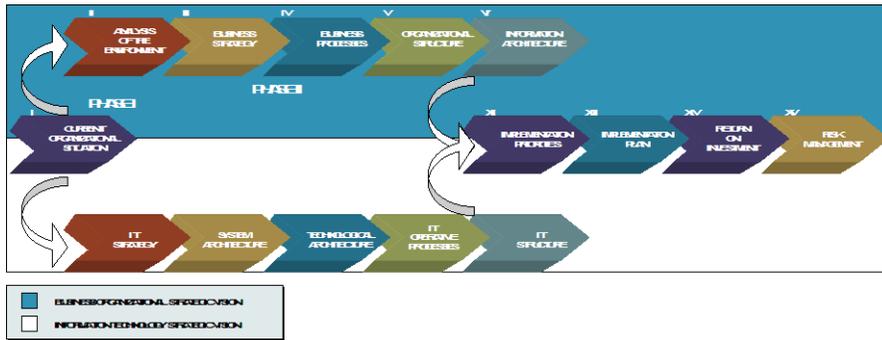


Figure 4: ITSP Methodology

The evolution process is represented by an inverse pyramid where business strategy represents the “axioms” of the archetype of the organization’s. These axioms are considered as true, i.e., fundamental principles, in virtue that they are congruent with the reality of the environment. In every case, the ITSP tries to be in contact with the real world in order to give to its construction, logical coherence. The organization propositions [5] (the IT strategy [6], the organization model, the IT model and the planning model) are deduced from the axioms through a logic inference method. Thus, every proposition is true if it can be deduced from the axioms.

This definition is in agreement with the fact that the efficiency of an enterprise and the effective use of the IT depend on the concordance that exists with the business strategy. If the business strategy is incompatible with the physical structure of the enterprise and the configuration of the IT then, the functionality of the organizational areas will be inefficient. It is important to note that the organizational axioms are not necessarily absolute, but they evolve in accordance with the internal and external changes of the environment.

The ITSP methodology (Figure 4) is organized in fifteen modules which are divided in four phases, and conceived in two visions. In addition, it is concerned with creating a business/organizational vision, which provides the critical information inputs and, it also forms the foundations for later stages of planning. As well as this, it creates a vision of the IT, which exploits new technological solutions and it improves the enterprise situation.

4. Business Process Modeling and Partially

Ordered Transition DPPN

In business process modeling, high-level business strategies are refined up to the point when they reach a tactical business strategy level, described only in terms of goals and strategies¹.

Business strategy decomposition represents a hierarchy of objective/decision-points, varying from the high-level business strategy with the maximum long-term impact to the more refined operational business strategy (goal, strategy) with relative short-term impact.

The business strategy refinement process concludes when a resulting business strategy can be transformed into an executable action. In this sense, the nodes found in the lowest levels of the business strategy decomposition tree are usually mapped into actions.

A business process is regarded as a set of activities. Activities are considered as operationalizations of goals and are applied in accordance with the strategies to achieve the goals. Strategies determine the legal sequential movements that can be made from any activity to another. The structure of each node in the business strategy decomposition is a complex object, defined by the ordered pair goal-strategy.

For completeness let us recall some basic notations of ordering. Given a poset (X, \preceq) a successor of an element $x \in X$ is an element y such that $x \preceq y$, but $x \neq y$ and there is no third element u between x and y . x is a predecessor of y if y is a successor of x . In symbols, for any $x \in X$

Successors of x : $y \in \text{suc}(x)$ iff $x \neq y$, $x \preceq y$ and $\forall u : x \preceq u \preceq y \implies (u = x) \vee (u = y)$

Predecessors of x : $y \in \text{pre}(x)$ iff $y \neq x$, $y \preceq x$ and $\forall u : y \preceq u \preceq x \implies (u = y) \vee (u = x)$

The graph of the ordering is the graph whose vertices are the points in X and each pair (x, y) , where y is a successor of x determines an edge. The graph corresponding to the ordering “ \preceq ” defined is a directed acyclic graph (*DAG*).

The minimal elements are those with no predecessors, i.e. nodes with null inner degree in the *DAG*. The maximal elements are those with no successors, i.e. nodes with null outer degree in the *DAG*. In this ordering the conditions with no input transitions correspond to the minimal elements, and the conditions with no output transitions correspond to the maximal elements.

Since the business strategy decomposition determines actions sequence applications, a process can be ordered as follows.

¹For simplification, we decompose the business strategy in goals and strategies, which we consider is adequate from an operational point of view

Definition 24. Let X be a process and $x, y \in X$ two activities. We say that the activity y “depends on” the activity x , and we denoted it by $x \preceq y$, if the corresponding decomposed node of x is upper than that of y in the business strategy decomposition tree.

Property 25. Clearly, “ \preceq ” establishes a partial ordering.

The partial order concept guarantees that the nodes found in the lowest levels of the business strategy decomposition tree, are already partially ordered and ready to be mapped into what next, is defined to be a partially ordered *DPPN*.

Definition 26. A partially ordered transition Decision Process Petri Net is a duple $(DPPN, \preceq)$, where *DPPN* is a Decision Process Petri Net and \preceq_q is the partial order defined on the elements of the set of transitions Q such that the following conditions hold:

- $q_1 \prec_q q_2$ iff $q_1 \preceq_q q_2$ and $\neg(q_2 \preceq_q q_1)$.
- $q_1 \sim_q q_2$ iff $q_1 \preceq_q q_2$ and $q_2 \preceq_q q_1$.

Note that the order of the *DPPN* is the order established by the “depends on” relationship (see the definition of \preceq).

Events are actions which take place in a process. The occurrence of these events is controlled in part by the state of the process. The state of a process can be described as a set of conditions. The minimal elements of the net are those conditions associated to the initial marking. Since events are actions, they may occur, for an event to occur, it may be necessary that certain preconditions hold. Each transition has associated a strategy that determines the preconditions to hold or not and may cause post-conditions to become true.

Proposition 27. Let us suppose that all the condition of Theorem 19 are satisfied and let us suppose that the *DPPN* is a partially ordered Decision Process Petri Net. Then, equation (6) reduces to:

1	0	$U_k(p_0)$
$\sigma_{0j_m}^\Delta(p_{\varsigma(1)})$	0	0	$U_k(p_1)$
...	$\sigma_{0j_n}^\Delta(p_{\varsigma(2)})$	0
...	0	...	$U_k(p_i)$
...	$\sigma_{nj_v}^\Delta(p_{\varsigma(n)})$	0	...
$\underbrace{\hspace{15em}}_{\sigma^\Delta}$						$\underbrace{\hspace{2em}}_U$

(7)

Proof. It follows from Corollary 20 and the fact that the Decision Process Petri Net is partially ordered. □

5. Application Examples

The aim of this section is to present two business process application examples, represented by the partially order (DPPN), where the optimum strategy and its correspondent tracking are described.

Example 28. Let us consider an insurance broker agency. As a broker, the agency sells policies for different companies. The main products are life and automobile policies. For selling and advertising the insurance company obtains detailed information from potential customers (C), and from private and governmental agencies (A). This information is distributed between the company's agents (AG) which contact potential clients via phone and try to set up a conference call; however, they also have their own sources of information. At the interview, the agent examines the client's current insurance coverage and tries to find an opportunity for a policy that will best fit the customer's needs. Before obtaining an insurance policy, the new client suffers an identity investigation. In the case of a life insurance, the client has, in addition, to approve a physical examination test in an accredited hospital (H). In the case that the investigation is positive both parts sign a policy and keep a copy of the contract. If during the investigation irregularities are found, the agent is informed, who meets with the client in order to find new options. The insurance policy is in effect when the client makes the first insurance premium payment. Every policy carries with a schedule of premiums, which varies with the type and coverage. Each policy provides a commission for the agency. The commission varies with the insurance company, policy type and coverage. The insurance company management (M) defines the commissions politic, which varies from agency to agency. The agency splits the commission received for each policy with the agent who sold it; the rate depends on the seniority of the agent. Once a policy has been sold, the agency submits premium bills to the client, collects payment and sends the payment, minus its commission, to the insurance company. If a client fails to pay premiums, the agent who sold the policy is informed, so that he can contact the client. Claims can be made on insurance policies as specified in the policy itself. Clients or beneficiaries (B) contact the agent to file such claims. Life insurance claims may be made by the beneficiaries on the death of the insured. In both cases, the insurance company sends an adjustor (AD) to legitimate the claim and arrange the final insurance details. For an automobile insurance policy, claims are made when the car is involved in an accident, damaged or stolen. For simplification, we will consider just the organizational strategy of the insurance company. Let us construct the organizational strategy like in [6]. In the business strategy decomposition tree

Objectives	Goals	Strategies	CSEs
O1 Achieve Market Leadership position (M)	G1 Reduce Operation Costs 5% (O1, O2)	S1 Improvements in Marketing Practices (O3, O4, M, A)	C1 Improve Cash Flow Management (O1, G3)
O2 Improve Service Quality (A, C, AG, H, M, E, AD)	G2 Achieve 30% Market Participation (O1, O4, M)	S2 Penetration into New Markets (O1, O2, G3)	C7 Growth Through Acquisition (O1, S2)
O3 Gain Competitive Advantage (M)	G3 Improve Stockholders Gains 15% (O1, O5, M)		C3 Improve Human Resources Training (O4)
O4 Enhance Company Image (M)	G4 Introduce IS/IT Solutions within two year (O2, O3)	S3 Improvements in New Product Generation (O2, O3)	C5 Improve Client Service (O2, G4, S3)
			C4 Introduce Strategic Distinction (O1, O3, S1)

Figure 5:

each node has a complex structure as follows: objective, goal, strategy, critical success factors (CSF). The complete business strategy decomposition is out of the scope of this article. However, we are following the decomposition process presented in [7]. The decomposition is as follows:

(Achieve Market Leadership position, Reduce Operation Costs 5%, Penetration into New Markets, Growth Through Acquisition)

(Achieve Market Leadership position, Reach 30% of Market participation, Penetration into New Markets, Growth Through Acquisition)

(Achieve Market Leadership position, Reach 30% of Market participation, Penetration into New Markets, Improve Cash Flow Management)

↓
...

The business strategy decomposition tree is shown in Figure 5.

Next, the partially ordered *DPPN* net model ($DPPN, \preceq$) is constructed by mapping the activities in the business strategy decomposition tree. Notice that the goals are represented by the places while the transitions represent the activities. The partially ordered *DPPN* (Figure 6) has the following specifications:

Places: P_0 : order requested, P_1 : investigated client, P_2 : examined client health, P_3 : rejected order, P_4 : declined order, P_5 : consented physical examination and admitted antecedents, P_6 : authorized policy, P_7 : delivered policy.

Transitions: q_1 : investigate client antecedent, q_2 : review physical condition, q_3 : deny physical examination, q_4 : refuse antecedents, q_5 : accept order, q_6 : sign contract, q_7 : send life policy.

i) Stability: From the incidence matrix of the partially ordered Petri net shown in Figure 6 given by

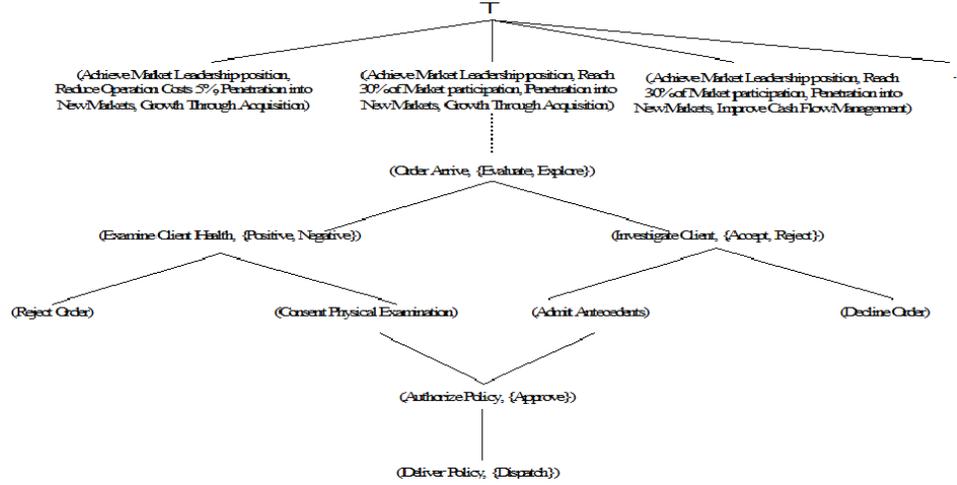


Figure 6:

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

choosing $\Phi = [1, 1, 1, 1, 1, 1, 1, 1]$, $\Phi > 0$, we obtain that $A\Phi \leq 0$ concluding stability.

ii) Optimum Strategy Analysis: Define the Lyapunov like function L in terms of the entropy $H(p_i) = -p_i \ln p_i$ as $L = \max_{i=1, \dots, |\alpha|} (-\alpha_i \ln \alpha_i)$ then:

$$\begin{aligned} U_{k=0}(p_0) &= 1, \\ U_{k=0}^{\sigma_{hj}}(p_1) &= L[\sigma_{01}(p_1) * U_{k=0}^{\sigma_{01}}(p_0)] = L[7/10 * 1] = \max H[7/10 * 1] = 0.249, \\ U_{k=0}^{\sigma_{hj}}(p_2) &= L[\sigma_{02}(p_2) * U_{k=0}^{\sigma_{02}}(p_0)] = L[3/10 * 1] = \max H[3/10 * 1] = 0.361, \\ U_{k=0}^{\sigma_{hj}}(p_3) &= L[\sigma_{23}(p_3) * U_{k=0}^{\sigma_{23}}(p_2)] = L[8/10 * 0.361] = \max H[8/10 * 0.361] = \\ &0.358, \\ U_{k=0}^{\sigma_{hj}}(p_4) &= L[\sigma_{14}(p_4) * U_{k=0}^{\sigma_{14}}(p_1)] = L[8/10 * 0.249] = \max H[8/10 * 0.249] = \\ &0.321, \\ U_{k=0}^{\sigma_{hj}}(p_5) &= L[\sigma_{15}(p_5) * U_{k=0}^{\sigma_{15}}(p_1) + \sigma_{25}(p_5) * U_{k=0}^{\sigma_{25}}(p_2)] = \max H[4/10 * 0.249 + \\ &4/10 * 0.361] = 0.286, \\ U_{k=0}^{\sigma_{hj}}(p_6) &= L[\sigma_{56}(p_6) * U_{k=0}^{\sigma_{56}}(p_5)] = L[1 * 0.286] = \max H[0.286] = 0.358, \end{aligned}$$

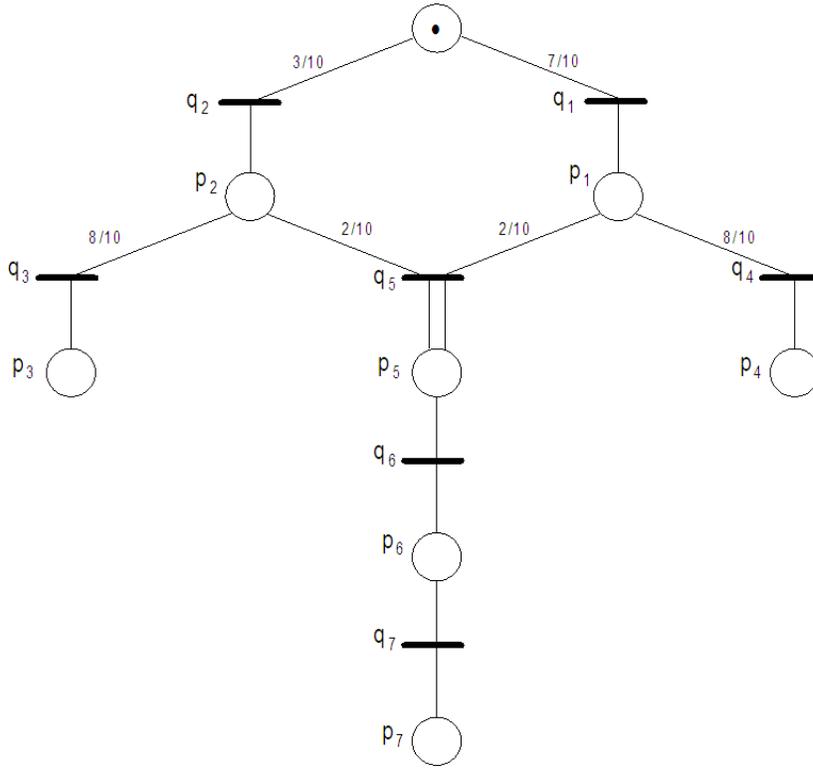


Figure 7:

$$U_{k=0}^{\sigma_{hj}}(p_7) = L[\sigma_{67}(p_7) * U_{k=0}^{\sigma_{67}}(p_6)] = L[2 * 0.358] = \max H[2 * 0.358] = 0.239,$$

the firing transition vector is u , which tracks the optimum strategy σ^Δ for selling a policy, is given by transitions q_1, q_2, q_5, q_6 and q_7 as follows $u = [1, 1, 0, 0, 1, 1, 1]$

In this case the strategy is optimum because the selling is obtained. In case that the policy is rejected the operative costs are supported by the insurance company. Alternatively, we have:

$$U_{k=0}^{\sigma_{hj}}(p_3) = L[\sigma_{23}(p_3) * U_{k=0}^{\sigma_{23}}(p_2)] = L[8/10 * 0.361] = \max H[8/10 * 0.361] = 0.358,$$

$$U_{k=0}^{\sigma_{hj}}(p_4) = L[\sigma_{14}(p_4) * U_{k=0}^{\sigma_{14}}(p_1)] = L[8/10 * 0.249] = \max H[8/10 * 0.249] = 0.321.$$

Example 29. Continuing with the insurance broker agency we have that in case of a car accident, the insurance company depends on the adjustor appraisal to evaluate the damages. To maintain company profitability the adjustor must evaluate the case so that only the minimal necessary repairs will be con-

sidered. In this sense, the adjustor evaluation is expected to be in favor of the insurance company because of his dependence on the latter. However, the adjustor must be careful, because the insurance company wants to offer a good service in order to keep the client. As a result, the automobile owner depends on the appraisal of the adjustor for an appropriate accident evaluation. The automobile owner can also be assisted by an authorized garage to obtain a fair evaluation of the car's damage. Notice that, the garage must satisfy both the client and the insurance company, given that the garage income depends on the car owner and on the insurance company. In case that the accident includes physical damage, the client and passengers must be directed to an accredited hospital for medical treatment. Three different strategies can be presented to manage a car accident in order to optimize the company's profitability ([4]). To improve the operation cost, small accidents can be directly evaluated by the adjustor or the authorized garage, and reported to the insurance company. Accidents of considerable size must be managed centrally by the insurance company.

Places: P_0 : claim settled, P_1 : handled accident info centrally, P_2 : handled accident info by authorized garage, P_3 : handled accident info by adjustor, P_4 : verified policy covering centrally, P_5 : verified policy covering by authorized garage, P_6 : verified policy covering by adjustor, P_7 : corroborated accident details, P_8 : evaluated damage centrally, P_9 : got medical treatment cost, P_{10} : determined accident in range, P_{11} : send info to be handle centrally, P_{12} : got accident info by adjustor, P_{13} : assessed client antecedents, P_{14} : determined accident covering centrally, P_{15} : got accident info by authorized garage, P_{16} : evaluated damage by authorized garage, P_{17} : determined accident in range, P_{18} : send info to be handle centrally, P_{19} : adjusted policy and made covering offer centrally, P_{20} : made covering offer by authorized garage, P_{21} : evaluated damage by adjustor, P_{22} : made covering offer by adjustor.

Transitions: q_1 : handle accident info centrally, q_2 : handle accident info by authorized garage, q_3 : handle accident info by adjustor, q_4 : verify policy covering centrally, q_5 : verify policy covering by authorized garage, q_6 : verify policy covering by adjustor, q_7 : corroborate accident details, q_8 : evaluate damage, q_9 : get medical treatment cost, q_{10} : determine accident in range, q_{11} : send info to be handle centrally, q_{12} : get accident info by adjustor, q_{13} : assess client antecedents, q_{14} : determine accident covering centrally, q_{15} : get accident info by authorized garage, q_{16} : evaluate damage by authorized garage, q_{17} : determine accident in range, q_{18} : send info to be handle centrally, q_{19} : adjust policy and make covering offer centrally, q_{20} : make covering offer by authorized garage, q_{21} : evaluated damage by adjustor, q_{22} : make covering offer by adjustor.

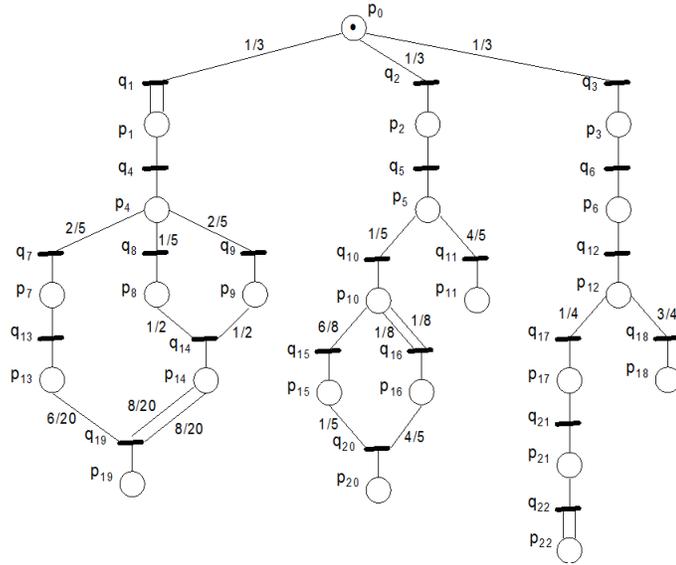


Figure 8:

i) Stability: From the incidence matrix of the partially ordered Petri net shown in Figure 7 choosing $\Phi = [1, 1/2, 1, 1, 1/2, 1, 1, 1/2, 1/2, 1/2, 1, 1, 1, 1/2, 1, 1, 2, 1, 1, 5/2, 3, 1, 1/2]$, $\Phi > 0$ we obtain that $A\Phi \leq 0$ concluding stability (the details of this computation are omitted).

ii) Optimum Strategy Analysis: Define the Lyapunov like function L in terms of the Entropy $H(p_i) = -p_i \ln p_i$ as $L = \max_{i=1, \dots, |\alpha|} (-\alpha_i \ln \alpha_i)$ then:

a) The optimum strategy σ^Δ for accidents of considerable size that must be manage centrally by the assurance company is represented by

$$\begin{aligned}
 U_{k=0}(p_0) &= 1, \\
 U_{k=0}^{\sigma_{hj}}(p_1) &= L[\sigma_{01}(p_1) * U_{k=0}^{\sigma_{01}}(p_0)] = L[1/3 * 1] = \max H[2/3 * 1] = 0.270, \\
 U_{k=0}^{\sigma_{hj}}(p_4) &= L[\sigma_{14}(p_4) * U_{k=0}^{\sigma_{14}}(p_1)] = L[1 * 0.270] = \max H[0.270 * 1] = 0.353, \\
 U_{k=0}^{\sigma_{hj}}(p_7) &= L[\sigma_{47}(p_7) * U_{k=0}^{\sigma_{47}}(p_4)] = L[2/5 * 0.353] = \max H[2/5 * 0.353] = 0.276, \\
 U_{k=0}^{\sigma_{hj}}(p_8) &= L[\sigma_{48}(p_8) * U_{k=0}^{\sigma_{48}}(p_4)] = L[1/5 * 0.353] = \max H[1/5 * 0.353] = 0.187, \\
 U_{k=0}^{\sigma_{hj}}(p_9) &= L[\sigma_{49}(p_9) * U_{k=0}^{\sigma_{49}}(p_4)] = L[2/5 * 0.353] = \max H[2/5 * 0.353] = 0.276, \\
 U_{k=0}^{\sigma_{hj}}(p_{13}) &= L[\sigma_{7,13}(p_{13}) * U_{k=0}^{\sigma_{7,13}}(p_7)] = L[1 * 0.276] = \max H[1 * 0.276] = 0.355,
 \end{aligned}$$

$$U_{k=0}^{\sigma_{hj}}(p_{14}) = L[\sigma_{8,14}(p_{14}) * U_{k=0}^{\sigma_{8,14}}(p_8) + \sigma_{9,14}(p_{14}) * U_{k=0}^{\sigma_{9,14}}(p_9)] = L[1/2 * 0.187 + 1/2 * 0.276] = \max H[1/2 * 0.187 + 1/2 * 0.276] = 0.338,$$

$$U_{k=0}^{\sigma_{hj}}(p_{19}) = L[\sigma_{13,19}(p_{19}) * U_{k=0}^{\sigma_{13,19}}(p_{13}) + \sigma_{14,19}(p_{19}) * U_{k=0}^{\sigma_{14,19}}(p_{14})] = L[6/20 * 0.355 + 8/20 * 0.338] = \max H[6/20 * 0.355 + 8/20 * 0.338] = 0.343.$$

the firing transition vector is $u^\Delta = [1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0]$.

For this case the adjustor or the garage must abort the process because the accident is out of their range obtaining that:

$$U_{k=0}^{\sigma_{hj}}(p_{11}) = L[\sigma_{5,11}(p_{11}) * U_{k=0}^{\sigma_{5,11}}(p_5)] = L[4/5 * 0.367] = \max H[4/5 * 0.367] = 0.359,$$

$$U_{k=0}^{\sigma_{hj}}(p_{18}) = L[\sigma_{12,18}(p_{18}) * U_{k=0}^{\sigma_{12,18}}(p_{12})] = L[3/4 * 0.367] = \max H[3/4 * 0.367] = 0.355,$$

concluding

$$U_{k=0}^{\sigma_{hj}}(p_{19}) < U_{k=0}^{\sigma_{hj}}(p_{11}) < U_{k=0}^{\sigma_{hj}}(p_{18}),$$

i.e. $U_{k=0}^{\sigma_{hj}}(p_{11}), U_{k=0}^{\sigma_{hj}}(p_{18})$ are more expensive than $U_{k=0}^{\sigma_{hj}}(p_{19})$.

b) The optimum strategy σ'^Δ for small accidents that must be manage ideally by the garage is represented by

$$U_{k=0}^{\sigma_{hj}}(p_2) = L[\sigma_{02}(p_2) * U_{k=0}^{\sigma_{02}}(p_0)] = L[1/3 * 1] = \max H[1/3 * 1] = 0.366,$$

$$U_{k=0}^{\sigma_{hj}}(p_5) = L[\sigma_{25}(p_5) * U_{k=0}^{\sigma_{25}}(p_2)] = L[1 * 0.366] = \max H[1 * 0.366] = 0.367,$$

$$U_{k=0}^{\sigma_{hj}}(p_{10}) = L[\sigma_{5,10}(p_{10}) * U_{k=0}^{\sigma_{5,10}}(p_5)] = L[1/5 * 0.367] = \max H[1/5 * 0.367] = 0.191,$$

$$U_{k=0}^{\sigma_{hj}}(p_{15}) = L[\sigma_{10,15}(p_{15}) * U_{k=0}^{\sigma_{10,15}}(p_{10})] = L[6/8 * 0.191] = \max H[6/8 * 0.191] = 0.278,$$

$$U_{k=0}^{\sigma_{hj}}(p_{16}) = L[\sigma_{10,16}(p_{16}) * U_{k=0}^{\sigma_{10,16}}(p_{10})] = L[1/8 * 0.191] = \max H[1/8 * 0.191] = 0.089,$$

$$U_{k=0}^{\sigma_{hj}}(p_{20}) = L[\sigma_{15,20}(p_{20}) * U_{k=0}^{\sigma_{15,20}}(p_{15}) + \sigma_{16,20}(p_{20}) * U_{k=0}^{\sigma_{16,20}}(p_{16})] = L[1/5 * 0.278 + 4/5 * 0.089] = \max H[1/5 * 0.278 + 4/5 * 0.089] = 0.261,$$

the firing transition vector is $u'^\Delta = [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0]$.

Intuitively the result is correct, because the best option for the insurance company is that after a car accident happens, the customer takes the car to the garage and the company does not have to send an adjustor.

c) The strategy σ'' for small accidents that must be manage by the adjustor is represented by

$$U_{k=0}^{\sigma_{hj}}(p_3) = L[\sigma_{03}(p_3) * U_{k=0}^{\sigma_{03}}(p_0)] = L[1/3 * 1] = \max H[1/3 * 1] = 0.366,$$

$$U_{k=0}^{\sigma_{hj}}(p_6) = L[\sigma_{36}(p_6) * U_{k=0}^{\sigma_{36}}(p_3)] = L[1 * 0.366] = \max H[1 * 0.366] = 0.367,$$

$$U_{k=0}^{\sigma_{hj}}(p_{12}) = L[\sigma_{6,12}(p_{12}) * U_{k=0}^{\sigma_{6,12}}(p_6)] = L[1 * 0.367] = \max H[1 * 0.367] = 0.367,$$

$$U_{k=0}^{\sigma_{hj}}(p_{17}) = L[\sigma_{12,17}(p_{17}) * U_{k=0}^{\sigma_{12,17}}(p_{12})] = L[1/4 * 0.367] = \max H[1/4 * 0.367] = 0.219,$$

$$U_{k=0}^{\sigma_{hj}}(p_{21}) = L[\sigma_{17,21}(p_{21}) * U_{k=0}^{\sigma_{17,21}}(p_{17})] = L[1 * 0.219] = \max H[1 * 0.219] = 0.332,$$

$$U_{k=0}^{\sigma_{hj}}(p_{22}) = L[\sigma_{21,22}(p_{22}) * U_{k=0}^{\sigma_{21,22}}(p_{21})] = L[2 * 0.332] = \max H[2 * 0.332] = 0.271.$$

The firing transition vector is $u' = [0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1]$.

Notice that since $U_{k=0}^{\sigma_{hj}}(p_{22})$ is strictly bigger than $U_{k=0}^{\sigma_{hj}}(p_{20})$ small accidents must be handled by the garage, whenever it is possible.

6. Conclusions and Future Work

A formal framework for business process modeling and validation, using partially ordered Decision Process Petri Nets, has been presented. The paper was motivated by the fact that it is essential to combine decision process systems and business processes. The business process model was supported by an information technology strategic planning (ITSP) model and methodology. The modeling methodology was based on business strategy transformation. Application examples where decision process properties were shown to hold were addressed.

Acknowledgements

I would like to thank Ph.D student Julio Clempner for working out the application examples.

References

- [1] W. Beats, Aligning information system with corporate strategy, *Journal of Strategic Information Systems*, **1**, No. 4 (1992).
- [2] Z. Retchkiman Konigsberg, Business process modeling and verification, *International Journal of Computational and Numerical Analysis and Applications*, **1**, No. 4 (2002), 333-352.
- [3] M. Hammer, *Reengineering Work Don't Automate Obliterate*, Harvard Business Review (July-August 1990), 104-112.

- [4] M. Hammer, J. Champy, *Reengineering the Corporation: A Manifesto for Business Revolution*, Harper Business (1993).
- [5] J. Henderson, N. Venkatraman, Strategic alignment: Leveraging information technology for transforming organization, *IBM System Journal*, **32**, No. 1.
- [6] W.R. King, Strategic planning for management information systems, *MIS Quarterly*, **2**, No. 1 (1978), 27-37.
- [7] A.A.I. McCracken, W.M. Potts, C. Potts, Goal decomposition and scenario analysis in business process reengineering, *LNCS*, **811** (1994), 94-104.
- [8] Z. Retchkiman Konigsberg, Stability analysis of difference equations and its application to discrete event systems modeled with Petri nets, *International Mathematical Journal*, **5**, No. 4 (2004).
- [9] Z. Retchkiman Konigsberg, Decision process Petri nets: Analysis and optimization, *International Journal of Pure and Applied Mathematics*, To Appear.