# A Petri Net Based Methodology for Business Process Modeling and Simulation

Joseph Barjis, Han Reichgelt

Georgia Southern University
P.O. Box 8150
Statesboro, GA 30460, U.S.A.

**Abstract.** In this paper we introduce a modeling methodology for the study of business processes, including their design, redesign, modeling, simulation and analysis. The methodology, called TOP, is based on the theoretical concept of a transaction which we derive from the Language Action Perspective (LAP). In particular, we regard business processes in an organization as patterns of communication between different actors to represent the underlying actions. TOP is supported by a set of extensions of basic Petri nets notations, resulting in a tool that can be used to build models of business processes and to analyze these models. The result is a more practical methodology for business process modeling and simulation.

## 1 Introduction

Research into business processes has recently seen a re-emergence as evidenced by the increasing number of publications and corporate research initiatives. Research into business processes encompasses a broad spectrum of topics, including business process design, redesign, modeling, simulation and analysis.

The reason for this resurgence in the study of business processes are not hard to fathom. Although organizations have invested heavily in new IT applications, and the software engineering community has worked diligently on producing new methodologies and tools for constructing IT applications, the quality of many IT applications remains poor (see, e.g., the articles in [7]). As organizations struggle to ensure that their IT investments do indeed provide value, their focus is shifting increasingly to the improvement, transformation and management of business processes [9]. In the past few decades businesses, misguided by the belief that IT alone will solve all their corporate woes, have overemphasized the role of IT in gaining a competitive edge while underestimating the importance of a clear understanding and critical analysis of their business processes. However, the tide has shifted and the prevailing standpoint in recent publications suggests a much greater focus on process-centric and process-driven organizations, where, by contrast, IT becomes "second class citizen" and business processes take the lead. An extreme proponent of this proposition is Carr [3], who argues that "the technology potential

Barjis J. and Reichgelt H. (2006).
A Petri Net Based Methodology for Business Process Modeling and Simulation.
In *Proceedings of the 4th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems*, pages 3-15
DOI: 10.5220/0002504700030015

for differentiating one company from the pack inexorably declines as IT becomes accessible and affordable to all". While Carr's position was regarded as too extreme by the majority of CEOs, CIOs, CTOs, as well as prominent academicians, there is a consensus that Carr was right to point out that IT, in and by itself, does not provide value. As Smith & Finger [10] put it, "IT doesn't matter, business processes do."

The renewed focus on business processes has led to the emergence of a large number of computer supported methodologies for modeling, analyzing and designing business processes. After all, following a well-established formal methodology will help an analyst determine which business related information to look for and how to represent this in a format that makes it easy to analyze and communicate with stake-holders, such as end users, managers and so on. Unfortunately, none of the methodologies that have emerged are entirely satisfactorily. Many either lack a clear theoretical foundation. Although they often come with an easy to use notation, the fact that they have no clear theoretical underpinning typically makes it hard to objectively justify the models built in them. Others have a clear conceptual foundation, but often lack an intuitive notation to construct and graphically represent models in.

This paper tries to remedy this situation by proposing a methodology based on a well understood theoretical foundation and containing an intuitive, easy to use notation. Our methodology, which we will refer to as "TOP" for Transaction Oriented Petri net methodology, is based on the concept of a transaction, which we derive from the language action perspective (e.g., [5, 6]). A transaction is regarded as a pattern of interactions between different actors, leading to a set of actions. Transactions can then be strung together to produce business processes, or vice versa, (complex) business processes can be analyzed as consisting of collections of (atomic) transactions. In order to model these potentially complex webs of transactions, we propose to use Petri nets. However, ordinary Petri nets are less likely to adequately represent business processes. To this end, we contribute with a type of Petri nets that has hierarchical structure and process boundary. More detailed discussion will follow in the subsequent sections of the paper.

The organization of the paper is as follows. We first introduce the notion of a "transaction" and trace its origins back to the language action perspective. We then introduce transaction-oriented Petri nets. Finally, we introduce the TOP methodology using an explanatory example.

## 2 The Transaction Concept

The central theoretical concept in TOP is that of a transaction. The term "transaction" derives from the so-called language action perspective (LAP) as introduced in the DEMO methodology [5]. The transaction concept is based on the idea that an organization and its underlying business processes can be better understood through the observation of communication between the members of the organization and the organization's interaction with its environment. Thus, this concept looks at communication as a tool to capture action patterns that represent the business processes. In this context, the notion of communication is not just an exchange of information, but also includes the negotiation, coordination, agreement, and

commitment that lead to certain actions. In turn, these actions create new facts, deliver results, and accomplish the mission of an organization.

LAP traces its origins to Searle's speech act theory [8], which itself extended earlier work by Austin [2]. Austin's primary insight was that utterances have both a constative aspect, i.e., they express meaning, and a performative aspect, i.e. they are produced to achieve something. Winograd and Flores [11] used speech act theory to develop a modeling approach that introduces the notion of "conversation for action" as a sequence of communicative acts involving two actors. Dietz [5, 6] used the Winograd and Flores approach to derive the concept of a business transaction and made it the central concept in the DEMO methodology (**D**esign & **E**ngineering **M**ethodology for **O**rganizations formerly known as **D**ynamic **E**ssential **MO**deling, or **D**ynamic **E**ssential **M**odeling of **O**rganization).

In order to introduce the concepts of a transaction and business process, consider the following, artificial, example:

> *A visitor calls a hotel's reception to reserve a room. After being asked to provide the details of the required room and the dates of his intended visit, the visitor is asked to hold as the receptionist checks the availability of a room. While the visitor is on hold, a piece of Mozart is played over the phone. After a brief period, the receptionist gets back to the visitor and states that the hotel has an appropriate room available for the dates required. However, in order to confirm the reservation and give the visitor a confirmation number, the visitor has to pay for the room by credit card. The receptionist therefore asks the visitor for credit card information and, after receiving the information, verifies the card and availability of sufficient fund with the credit card company. Once the payment has been approved, the receptionist is able to give the confirmation number to the visitor.*

Business transactions are patterns of actions and interactions, as illustrated in Figure 1a. The *action* is the core of a business transaction and represents an activity that changes the state of the world and creates a new fact. An *interaction* is either the initiation of an action or the communication of a fact as the result of the action. An example of an interaction is a request made by one actor towards another actor that leads to creation of a new fact. Other examples of interactions are clicking an "apply" button, "submit" in an electronic form, or inserting a debit card into ATM to withdraw cash. In the hotel reservation example, the visitor's request for a room reservation and the receptionist's statement of the confirmation number are both *interactions*; and reserving the room by the receptionist is an *action*. Thus a transaction is interaction plus action (cf. Figure 1.b).


**Fig. 1.** a) The business transaction concept, b) The OER diagram (detailed and compact notations).

Accordingly, each business transaction is carried out in three phases, the *order phase*, the *execution phase*, and the *result phase*, corresponding to the first interaction (when an action is requested), the action itself and the second interaction (when the result of an action is communicated to the requester) respectively. The phases are abbreviated as O, E and R correspondingly and are the basis of the OER paradigm, as illustrated in Figure 2. Note that the order (O) phase and result (R) phase are interactions and the execution (E) phase an action, therefore they are represented by different colors.

We distinguish between simple and complex (composite) transactions. Business processes typically consist of numerous transactions that are chained together and nested into each other. *Simple* transactions do not involve, i.e., trigger or cause, other transactions during their execution. In *complex* transactions, on the other hand, one or more phases will trigger further, nested, transactions. Thus, in the hotel reservation example above, the receptionist can only make the reservation after she has received a credit card payment from the visitor. However, this transaction can only be completed once the receptionist has made the necessary checks with the credit card company. Thus, the "receive payment" transaction can only be completed once the receptionist initiates a transaction with a credit card company and this transaction maybe nested inside a larger transaction.

The hotel reservation example also illustrates that each transaction involves two actors. By two actors we mean two interacting parties. The actor that initiates the transaction is called the *initiator* of the transaction, while the actor that executes the transaction is called the *executor* of the transaction. Actors can be human actors, software agents or machines. For example, if the hotel reservation application is submitted online, a software agent, rather than a human receptionist, will make the hotel reservation.

This leads to the following series of definitions of the concept of a business transaction:

**Definition 1:** A *business transaction* is a generic pattern of activity carried out between two actors, an initiator and an executor. The activity is carried out in three phases, the order phase, the execution phase, and the result phase, (OER) in which the first and last phases are interactions and the execution phase is an action. The activity creates a new fact and changes the state of the world.

**Definition 2:** A *business transaction* comprises two types of actions: communicative action, which represents an *interaction*, and productive action, which represents an *action*. An *interaction* is coordinating or negotiating the essence of an *action* and communicating the result of an action. Thus an interaction takes place twice, before an action is carried out and after an action is carried out and a result is achieved.

**Definition 3:** A *business transaction* can have even deeper structure when it nests further transactions. Thus the primary transaction is called *composite* (or nesting) transaction and the embedded ones are called *nested* transactions.

**Definition 4:** A *business transaction* has a single start point and a single end point. A transaction starts with a request by an actor and ends with a result accepted by the same actor.

As the example above illustrates, the initiation, execution, or completion of a business transaction may lead to initiation and execution of new transactions. In this way transactions are chained into arbitrarily large structures, called *business processes* [5].

This then leads to the following definition of the concept of a business process:

**Definition 5:** A *business process* is a network of interrelated business transactions that delivers value to an external agent through the production of products (goods or service).

## 3 Transaction-Oriented Petri Nets

Earlier attempts made to map a business transaction into Petri nets were made using classical Petri nets [4]. However, the principles and features of the Transaction-Oriented Petri net adhere to the same of standard (classic) Petri net as well as features of hierarchical Petri nets. The extensions made are more towards suitability of Petri nets as a modeling technique for language action based business process modeling, where processes have deep structure (nested transactions).

### 3.1 Transaction-Oriented Petri Net Extensions

Although especially hierarchical Petri nets are eminently suitable for business process modeling [1], we add a few minor extensions. The primary extension derives from the fact that basic Petri nets do not allow one to model the fact that different parts of the process being modeled take place in units that are somehow distinguishable. Clearly, the ability to do is potentially of great importance in business process modeling, as one wants to be able to distinguish between departments, and indeed organizations involved in the same business process. Second, by introducing these extensions it is made feasible to distinguish between an interaction (or communicative action) and an action (or productive action). However, the most important reason of the proposed extension is the very nature of business processes that have deep structure (nested transactions). We therefore provide a simple extension to Petri nets that allows analysts to address the mentioned issues. Figure 2 represents the set of basic elements that we use in building Transaction-Oriented Petri Nets, and Table 1 gives description of each of the elements.

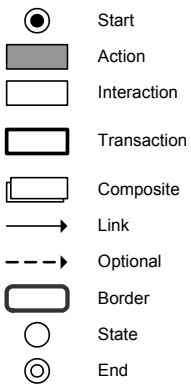


**Fig. 2.** Basic elements of TOP.

**Table. 1.** Elements in Transaction-Oriented Petri Nets.

| Element | Description |
|---|---|
| Start | A start represents a starting point for a process that leads to the creation of a new result or a previously unknown fact. A start place is characterized with a token. |
| Interaction | An interaction represents the transfer of some information from one actor to another, e.g., a request to perform some action, or a communication of the result of some action. In terms of the OER model, both the Order and the Result phases are interactions. |
| Action | An action represents a production act when a certain result is created. The Execution phase in the OER model is an action. |
| Transaction | A transaction encompasses the entire OER cycle. The introduction of a symbol for an entire transaction shows that our notation is an example of a hierarchical Petri net. The reason for the introduction is that it allows the modeler to abstract away from certain details that he or she considers irrelevant to the model. |
| Composite Transaction | A composite transaction is a set of transactions, and thus summarizes a sub-net. It is useful if the modeler wants to hide certain details. |
| Sequence Flow | A sequence flow indicates the order in which interactions and actions are initiated and performed. |
| Optional | An optional link represents a link that is neither a condition for proceeding nor it is essential, but it usually takes place. E.g., in order to apply for a policy usually customers requests a quote; relation of requesting a quote and applying for a policy is an optional one. |
| Boundary | A boundary illustrates the boundary of a business process (department or organization). It is this feature that allows to model intra- and inter-departmental and intra- and inter-organizational processes. |
| Intermediate state | An intermediate state represents a result or state achieved after an interaction or action. |
| End | An end notation represents a termination point for a process, and thus represents the final result of the process. |

As Table 1 illustrates, the Transaction-Oriented Petri nets contain distinct graphical elements for actions and interaction. Moreover, they include short-cut notations for a complete transactions and composite transactions, consisting of a set of transactions, and Transaction-Oriented Petri nets are therefore an example of hierarchical Petri nets. The reason for introducing short-cut notations for complete and composite transactions is to enable the modeler to (temporarily) hide details of the model under construction. This feature is useful both when it comes to constructing the model (modelers can temporarily ignore the details of certain transactions as they concentrate on other transactions within the business process) and

when it comes to seeking feedback on a model under construction from a stake-holder (hiding irrelevant information from stake-holders makes it easier for them to provide feedback on those aspects of the model that are of more direct relevance to them.) Since these are transitions in the sense in which the term is used in Petri nets, all are represented by rectangles.

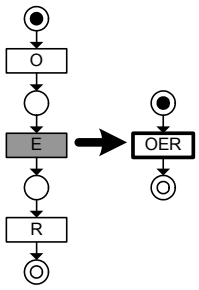


**Fig. 3.** The OER diagram using Petri net (detailed and compact).

A further extension is the inclusion of two special places, namely *start place* and *end place*, to indicate where a process starts and terminates respectively. Because the start and end place are special places, we represent them by additional marking on circles.

A third extension is the inclusion of the *boundary* element. It is this element that allows us to distinguish between intra-organizational and inter-organizational processes. This element allows an analyst to model the interaction of one process with another process within an organization or with the environment. This interaction can be modeled with a set of places between the process boundaries.

The final extension concerns the introduction of an *optional link.* In standard Petri nets, all input places must hold in order for a transition to be executed; otherwise said, execution of a transition is guaranteed only whenever all its input places hold a token. Optional links weaken this assumption and therefore allow the analyst to represent situations where the transition is executed, even when a state represented by its input places does not hold. For instance, in order to apply for a policy, usually a customer requests a quote; however, it doesn't prevent the customer to apply for a policy, if she hasn't requested a quote before. Thus, the relation of requesting a quote and applying for a policy is an optional one.

In concluding this section, Figure 3 represents a business transaction both in detailed and compact (compressed) forms and its three phases known as OER using elements of TOP that we just introduced.

### 3.2 An Example of a Transaction-Oriented Petri Net

In order to illustrate Transaction-Oriented Petri nets, we build a partial model of the hotel reservation example. Figure 4 represents the hotel reservation process at a very high level, and simply shows the process as having a starting place and an end place, and being conducted within the confines of the hotel, at least as far as the initiator is concerned. Given the fact that the reservation process is a composite transaction

(involving further transactions), we use the notation of composite transaction to represent this high level model.

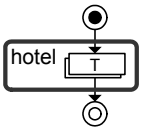


**Fig. 4.** The high-level "Room Reservation" model.

However, a little thought will quickly show that the analysis is too simple. While the visitor will initiate the business process, in order for the process to be completed, the hotel must receive payment from the visitor. Paying is an action initiated by the receptionist, who asks for the payment, but executed by the visitor. We therefore have two transactions:

| | |
|---|---|
| **Transaction 1:** | Reserving a room |
| **Initiator:** | Visitor |
| **Executor:** | Receptionist |
| **Fact:** | A room is reserved |
| | |
| **Transaction 2:** | Making payment |
| **Initiator:** | Receptionist |
| **Executor:** | Visitor |
| **Fact:** | Payment is made |

Since the second transaction is triggered while the first transaction is being completed, we have to expand our representation of the first transaction to make its three phases explicit. Figure 5a illustrates:

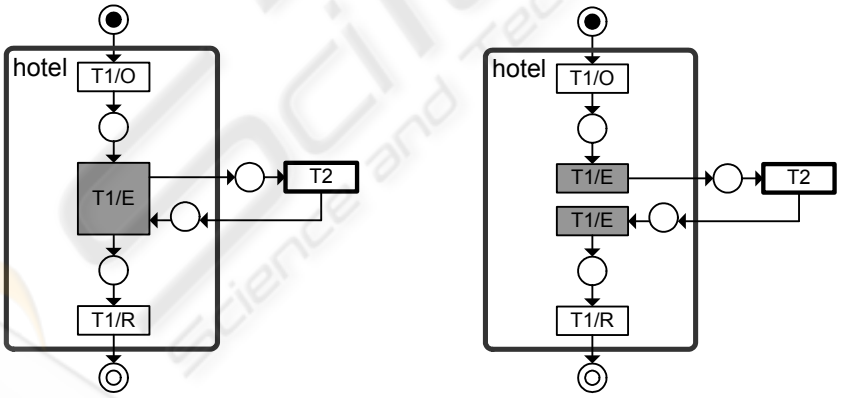


**Fig. 5.** a) The detailed "Room Reservation" model, b) Low level Petri net model of the "Room Reservation".

There is one further, but fundamental, clarification that is the core of TOP. This clarification concerns the nested structure of TOP that makes it a suitable technique for business process modeling based on the language action perspective. At first glance, it may seem that the model in Figure 5a is defective because it does not

represent a clear sequence of actions. After all, in order for the process to be completed successfully, Transaction 2 has to be completed before transition T1/E can be completed. We therefore assume that if any transition calls a nested transaction and the result of the nested transaction is needed for the completion of the nesting transaction, the nested transaction has to be completed before the transition itself can complete. Figure 5b illustrates this with more accurate syntax. In this figure TOP model is mapped into low level Petri net as a proof of syntactic and semantic soundness. The figure illustrates that T1/E phase starts immediately after T1/O is completed; however for its completion T2 should be entirely executed.

Since the purpose of this explanatory example is to introduce TOP, let add some more flavors of complexity and alterations to the hotel example. For instance, let think that the receptionist should cancel an existing reservation before requesting payment for the new reservation. In order to cancel an existing reservation, the receptionist should interact with the reservation system (e.g., database). This will change list of the transactions involved in the "Room Reservation" process:

| | |
|---|---|
| **Transaction 1:** | Reserving a room |
| **Initiator:** | Visitor |
| **Executor:** | Receptionist |
| **Fact:** | A room is reserved |
| | |
| **Transaction 2:** | Canceling a reservation |
| **Initiator:** | Receptionist |
| **Executor:** | Database |
| **Fact:** | Cancellation is confirmed |
| | |
| **Transaction 3:** | Making payment |
| **Initiator:** | Receptionist |
| **Executor:** | Visitor |
| **Fact:** | Payment is made |

Modified models of the "Room Reservation" incorporating the new transaction is illustrated in Figure 6. In this figure both TOP model (a) and low-level Petri net model (b) are illustrated.

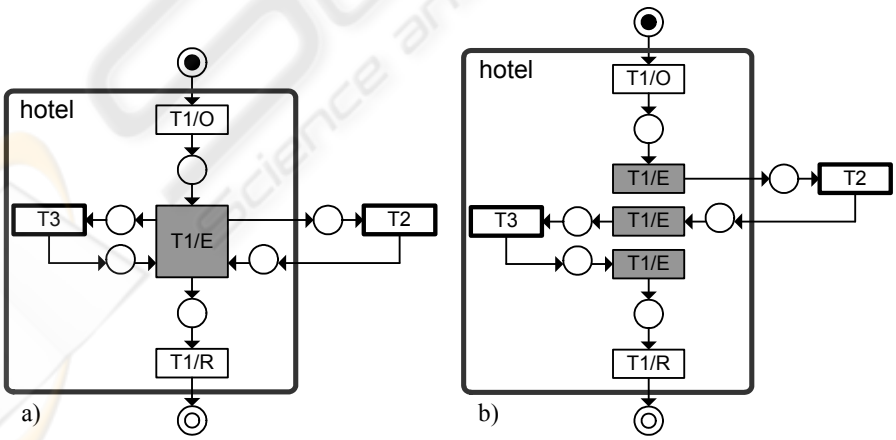


**Fig. 6.** a) Modified model of "Room Reservation"; b) and its low level Petri net model.

One final modification and then this introduction will be concluded. In the above modified model we stated that the cancellation of existing reservation should take place first before the receptionist can request for a payment. Let change it to a parallel process where both transactions (cancellation and payment) can be executed in parallel as depicted in Figure 7.

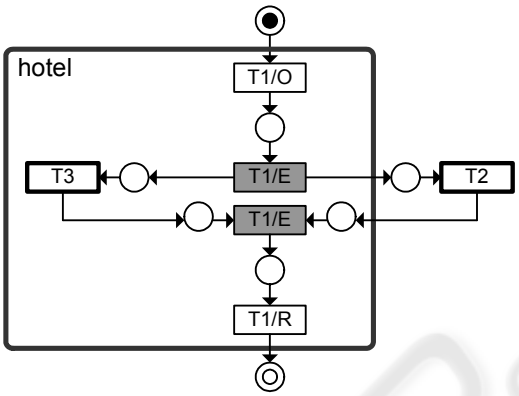


**Fig. 7.** Modified low level Petri net model of "Room Reservation".

Based on the explanatory example and discussed notions, the following extended definition of TOP can be formulated:

**Definition 1:** The *Transaction-Oriented Petri net* is a nested model that can be completely mapped (or represented) into standard place-transition Petri net. The TOP models both communicative action (interaction) and productive action (action). The main building blocks of TOP are business transactions. Each transaction is represented as a set of three transitions whereas each transition corresponds to one of the three transaction phases (OER). Two of these transitions represent an *interaction* and one of these three transitions represents an *action*. Overall, TOP can be classified as a hierarchical Petri net, but with some principal distinctions that are obvious from the example introduced.

In the following section we describe the steps used within the TOP Methodology, a roadmap describing all the phases and activities in each phase, for business process modeling.

## 4 The TOP Methodology

As we stated in the introduction, most current methodologies for business process modeling are deficient in that they either lack a clear theoretical basis, or an intuitive easy to use and understand notational framework. The Transaction-Oriented Petri Net-Based methodology TOP was designed specifically to provide both a clear theoretical foundation and an intuitive notational framework. In this section, we introduce the TOP methodology and illustrate its use through a simple case study.

The TOP methodology is a top-down design process that uses incremental refinement to arrive at a detailed model of a business process. The business process modeling process therefore starts with the identification of the major high-level business processes that an organization is engaged in, such as order processing, customer call processing, inventory control, admission to hospital, new member enrollment, insurance claim processing. Once the major business processes have been identified, they need to be described, based on internal documents used within the organization (e.g., policy manuals, training material, and so on) and/or interviews with for example the manager in charge of the specific business process.

The next step involves an analysis of each major business process as a network of business transactions, each involving two actors and each bringing about a change in the state of the system. In order to do so, one first identifies the basic transactions and the actors involved in them before constructing a model using transaction-based Petri nets. Given the hierarchical nature of transaction-based Petri nets, the construction of the model may itself take place in a top-down incremental refinement fashion in that the details of certain interactions or actions may initially be glossed over.

In short, the TOP methodology consists of the following steps:

1 Identification of major business processes (this will lead to a "big picture" of an organization)
2 Definition of major business processes (this will lead to a series of detailed pictures)
3 For each major business process,
   - Identification of business transactions
   - Description of business transactions
   - Construction of Transaction-Oriented Petri net.

### 4.1 Application of the TOP Methodology

For better application of TOP for business process modeling, a real life example should be considered. However, due to page restrictions per paper, an example of real life system is studied and presented in a companion paper published in the second part of this book.

## 5 Conclusion and Future Research

This paper has presented the TOP methodology for business process modeling. The TOP methodology combines a sound theoretical grounding in the concept of a transaction as developed within the Language Action Perspective with a powerful, yet easy to use and read graphical notation in the form of Transaction Oriented Petri nets.

There are a number of avenues for future research that we intend to explore. First, we will continue to use the TOP methodology to build additional models and to ask others to do so too. As more models are constructed, we will get a better understanding of the strengths and weaknesses of the current notation. While we believe it is relatively easy to learn (a belief that is partly based on the fact that some of our students have been able to use the TOP methodology to build business process

models), it is useful to put this belief to an empirical test. Further testing of the methodology may also reveal weaknesses in our graphical notation. It may for example be necessary to explicitly include the initiator and executor of a transaction. Also, continuous development of models, certainly for organizations in a similar line of business, will allow us to build a library of business processes. This in turn can be used to identify best practices.

Second, until now our work has primarily concentrated on building static models. However, one of the attractions of Petri net is that as model, once constructed, can be used to simulate a business process as well, and we are currently working on software that allows us to make our models "active" and run simulations. Since our main graphical tool is based on the widely accepted Petri net notation, we expect to be able to use an existing simulation package.

Third, a simulation package opens up an array of possibilities for future research. For example, once we can "run" our models, and collect meaningful statistics about them, it becomes possible to determine whether possible process changes are likely to be beneficial to the organization. In other words, a TOP simulator will allow us to determine whether a proposed redesign of a set of business processes is likely to lead to improved organizational performance, thereby taking the guess work out of many business process redesign exercises.

A TOP simulator may also provide a powerful tool that one can use to increase the success of IT applications. It is well know that many IT applications fail not because of technical reasons but because of "soft" reasons, including the non-acceptance of the new application by end users or a mismatch between the processes that an organization currently has in place, and the processes that would have to be implemented if the IT application is to be successful. While the need to take processes into account when it comes to implementing IT applications is generally accepted, the continuing failure of many IT applications suggests that IT professionals do not have a good handle on modeling how the introduction of an IT application affects organizational processes. A TOP based simulator would see how a proposed IT application is likely to impact the organization and to determine whether this impact is likely to be positive or negative, and what measures have to be put in place to increase its chances of success. This paper then merely presents a first step towards a comprehensive set of tools and techniques that should help analyst in modeling, simulating, and analyzing business processes.

## References


1. Aalst, W. van der; Hee, K. van (2002). Workflow Management: Models, Methods, and Systems, MIT Press.
2. Austin, J. L. (1962). How to Do Things with Words, Cambridge, Mass.: Harvard Uni Press.
3. Carr, Nicholas G. (2003). IT Doesn't Matter. Harvard Business Review, May 2003, 81(5)
4. Dietz J.L.G.; Barjis J. (1999). Supporting the DEMO Methodology with a Business Oriented Petri Net. In the proceedings of the International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design, Heidelberg, Germany, June 14-15.
5. Dietz, J.L.G. (1999). Understanding and modelling business processes with DEMO. In the Proceedings of the Annual International Conference on Conceptual Modelling, Paris, November.

6. Dietz, J.L.G. (2002). The Atoms, Molecules and Matter of Organizations. In the Proceedings of the Seventh International Workshop on the LAP, Delft, Netherlands, ISBN: 90-9015981-9.
7. Duggan, E. & Reichgelt, H. (2006) Measuring Information Systems Delivery Quality. Hershey, PA: Idea Group
8. Searle, J. (1969). Speech Acts: An Essay in the Philosophy of Language, Cambridge, Eng.: Cambridge University Press.
9. Smith, H.; Finger, P. (2003a). Business Process Management (BPM): The Third Wave. Meghan-Kiffer Press.
10. Smith, H.; Finger, P. (2003b). IT Doesn't Matter Business Processes Do. Meghan-Kiffer Press
11. Winograd, T.; Flores, F. (1986). Understanding Computers and Cognition: A New Foundation for Design. Ablex, Norwood.