Business Processes Modeling as Social Systems[1]

Joseph Barjis, Georgia Southern University, P.O. Box 8150, Statesboro, GA 30460, USA; E-mail: JBarjis@georgiasouthern.edu

ABSTRACT

Over the past few decades, a number of successful process modeling techniques have been developed. While these modeling techniques are instrumental in process modeling, they seem challenged to capture the deep structure of business process and its social character (commitment, coordination and negotiation). Most of the current modeling methods and techniques are characterized as flowchart-like techniques. Organizations are social systems, where its members communicate and negotiate to carry out different tasks and create results. On the other hand, many of the processes are nested within a super-process that can not be captured by flowchart-like techniques. Another issue of the existing modeling techniques is their semi-formality not lending to model execution (simulation) without further translations. This paper introduces a modeling method and technique that consider organizations and its business processes as a social system. The resulting models are based on the semantics of Petri net, and consequently, the models are fully executable. The proposed method is based on a rather non-traditional concept (the Language Action Perspective), and it adapts graphical notations of traditional modeling formalism (Petri net).

Keywords: social systems, business process modeling, modeling methodology

INTRODUCTION

Business processes have been studied for decades and even longer than that, but the real renaissance of research into business processes started with vanguards of re-engineering revolution such as Davenport, Short, Hammer, Champy (Davenport & Short, 1990; Hammer & Champy, 1993) whose works foregrounded business processes in the mainstream literature. Since then, numerous tools, techniques and methodologies have been developed to study the organizational phenomena and business processes.

A distinctive and important feature of an organization is its social nature –human actors interacting and collaborating to carry out tasks and fulfill the mission of the organization. As such, business process is not merely a sequence or flow of jobs, tasks, or physical materials, but a complex phenomenon involving actors communicating, negotiating, coordinating and agreeing upon certain tasks. As argued by advocates of alternative perspectives for business process study (Winograd & Flores, 1986, Stamper, 1988; Dietz, 1994), the social nature of business process entails a fundamentally different perspective to perceive the reality of an organization and the role (responsibility and authority) of its members. This understanding is difficult to achieve with conventional methods/techniques that are mostly characterized as flowchart-like techniques. One such new perspective was introduced in a framework referred to as the Language Action Perspective, or LAP for short, (Winograd & Flores, 1986). The LAP framework and its philosophical stance inspired emergence of several modeling methodologies and techniques such as SAMPO (Lehtinen & Lyytinen, 1986; Auramäki et al., 1988), Action Workflow model (Medina-Mora et al., 1992), DEMO (Dietz, 1994), BAT (Goldkuhl, 1996), to mention a few. However, since the main emphasis in these methodologies is placed on capturing communication acts and building business process models, their underlying modeling techniques do not lend to further execution or simulation (Rittgen, 2005), which makes it difficult to check the models for dynamic behavior and response to changes. In order to develop executable business process models based on a formalized semantics, this paper introduces a method and technique based on the *business transaction concept* derived from the LAP. The proposed method is further extension of CAP Net developed over the last decade by Dietz (2006), however the modeling technique (graphical notations) introduced is full adherent to the Petri net formal semantics.

Most of the previously introduced Petri net models are dominantly process or workflow oriented rather than business process as a social system. In contrast to prevailing process-oriented and object-oriented models, the introduced method allows not only model processes flow but also take into account the social character of the modeled enterprise such as interacting actors (or actor roles), and the nested structure of activities. This paper further develops the works of (Dietz & Barjis, 1999; Dietz & Barjis, 2000, Barjis & Reichgelt, 2006) on business process modeling deploying the transaction concept supported by Petri net.

In summary, the research finding reported in this paper is hoped to make the following contributions:

1. *Executable models* of business systems based on the *Transaction Concept*. Previous models developed based on this concept are mostly focused on producing well defined and detailed models, so called, *atoms*, *molecules* and *matter* of organizations. Our contribution is to make the resulting models executable to help system designers with model checking and validation, making changes to the model and study the impacts of the changes prior to the intended system development.
2. *Compact models* of complex systems using the transaction concept. Often, in systems modeling designers are either not interested in all the details, or the system under study is too large to be depicted at detailed level, or the designers may spotlight a part of the system while leaving some other parts concealed. In these situations, compact modeling where certain activities are compressed into one well defined component would be of great advantage. Also, when using diagrams, models rapidly get too large to manage.
3. *New knowledge*, generated as a result, contributes to the concepts of Model-Driven System Development, business systems modeling, simulation, modeling methodology, application of modeling and simulation, and advancing the new perspectives of system design and development.

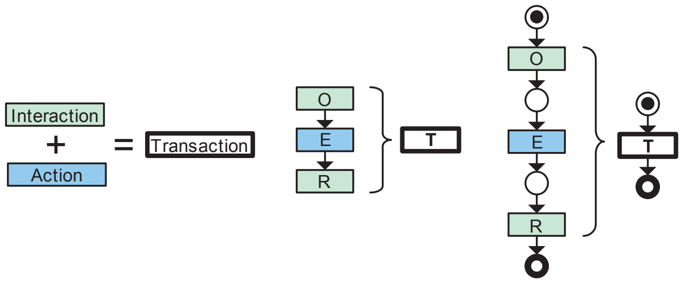
THE TRANSACTION CONCEPT

What follows is an illustrative introduction to the transaction concept using artifacts and constructs adapted by the authors. Readers, interested in more in-depth study about the transaction concept, are referred to the original works by Dietz (1994, 2006). We have adapted the Petri net notations and extended them as modeling constructs. Assuming that readers are familiar with the basic concepts of Petri nets that are widely used in systems analysis and design, we skip their introduction.

Transactions are patterns of interactions and actions, as illustrated in Figure 1a. In the figure, "action" and "interaction" are distinguished by different colors. An *action* is the core of a business transaction and represents an activity that brings about a new result, changing the state of the world. An *interaction* is communicative act involving two actors (actor roles) to coordinate or negotiate. An example of an interaction could be "requesting a new insurance policy", clicking "apply" or "submit" buttons on an electronic form, inserting a debit card into an ATM to withdraw cash, or pushing an elevator's summon button. Replying to the interacting actors and fulfilling their requests is an action, e.g., "issuing a new policy", "dispensing bills", "moving an elevator to the corresponding floor", etc.

Each business transaction is carried out in three distinct phases, the *Order phase*, the *Execution phase*, and the *Result phase*. These phases are abbreviated as O, E and R correspondingly (see Figure 1b), and constitute the OER paradigm (Dietz, 1994). The figure illustrates a business transaction in detailed OER form, and compact transaction form (T). Note that the order (O) and result (R) phases are interactions and the execution (E) phase is an action, therefore they are illustrated using different colors (the Execution phase is represented by a rectangle colored in blue (or gray in grayscale printout)). These three phases are a distinct feature

Figure 1. Transaction: a) pattern of action and interaction; b) sequence of three phases (detailed and compact); c) corresponding Petri net diagram



(a) (b) (c)

that entails the discussed method as a business process modeling technique versus just a process modeling. Also, these three phases not only allow for the boundary of an actor (or business unit) to be clearly defined, but also to depict interaction and action as a generic pattern involving (social) actors. Compared to UML, Flowchart, EPC and other conventional modeling methods, the transaction pattern clearly identifies the actors involved as it is discussed below. In other words, in conventional methods, a transaction would be reduced to only one execution phase omitting information about the relevant actors and their role.

Now, we try to introduce the further notions of the transaction concept along with the Petri net notations we adapted. In general, Petri net structure consists of places (graphically illustrated by circles and representing outcome of an activity or process), transitions (graphically illustrated by rectangles and representing an activity or process) and directed arcs (graphically illustrated by arrows and representing flow sequence). Figure 1c depicts a business transaction using the Petri net notations. Again, the figure illustrates a business transaction in detailed OER form, and compact transaction form (T). In the figure, the start and the end places are marked by different circles. These notations will make better sense when a complex process consisting of several related processes is studied and modeled.

Another notion of the transaction concept is the role of actors involved in a transaction. Each business transaction is carried out by exactly two actors (or actor roles), see Figure 2a. 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. Since the Order (O) and Result (R) phases are interaction between the two actors, their corresponding transitions are positioned between the two actors. The Execution (E) phase is an activity solely carried out by the executor and, therefore, its corresponding transition is positioned within the confines (boundaries) of the executor. In case of multiple actors, they will be conveniently denoted by the letter A and numbered (A1, A2, A#).

A transaction diagram should also represent how the created result (outcome) is recorded. Since each transaction brings about a new result, the Result phase of a transaction is linked to an oval-shaped element representing the new result created (see Figure 2b). For simplicity sake, the depiction of the oval representing a transaction result maybe omitted in the models studied later. If a business transaction is a simple one (not nesting further transactions), it is better to compress its three phases into a compact notation, see Figure 2c. In this case, the transaction is placed within the boundary of the executing actor, while the initiation and ending points are placed within the boundary of the initiating actor.

A distinction is made between simple and composite transactions. Actors' interactions may be arbitrarily complex, nested, extensive and multilayered (hierarchical). A complex collaboration typically consists 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 (like the above figure). In *composite* transactions, on the other hand, one or more phases will trigger further, nested, transactions. For instance, think if actor A1 contacts actor A2 to reserve a hotel room (we denote this request as Transaction 1, or T1). Actor A2 receives the request, checks the room availability, but in order to complete the request, it has to request actor A1 for a payment guarantee (we denote this second request as Transaction 2, or T2). For actor A2 to complete the reservation task, first the payment transaction should be completed. This process is represented in Figure 3a in the form of a nested transaction. Notice that the Execution phase of T1 now has several sub-phases or interactions, where each of the sub-phases is distinguished with a letter of the alphabet attached to the transaction number (e.g., T1a/E denotes "first sub-phase of the Execution phase of Transaction T1"). The process illustrated in the figure starts with the receiving of a reservation request and checking the room availability, then it waits for the payment transaction to get completed, only then the Execution phase gets completed, let say, by conveying a confirmation number to the first actor.

A close look at the reservation process reveals that in fact, the payment transaction, T2, is rather an interaction between the hotel and a credit card company. Thus, the process rather involves three actors (or actor roles): A1 (customer or guest), A2 (hotel receptionist) and A3 (credit card company). The interaction process between the three actors forms a nested transaction structure.

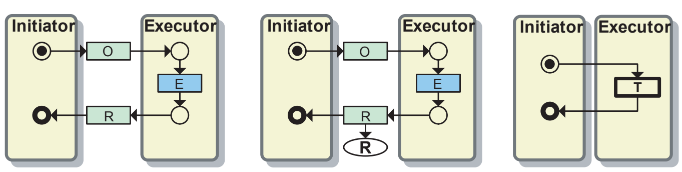
One of the limitations in many modeling techniques is coping with complex real-life systems. Usually models of real systems turn too large using diagrammatic representation. In dealing with this issue, we introduce the "composite" (or nesting) notation graphically represented as a multiple (layered) rectangle. For instance, the model illustrated can be reduced to one composite transaction as shown in 3b. This can be applied to any part of a complex system for the sake of compactness or for spotlighting a specific part of the system while concealing the other parts. The notion of nesting structure is especially helpful in inter-organizational process modeling in which a whole process within an organization or business unit can be reduced to a single composite transaction, thus, keeping the model more manageable.

It should be noted that at any point (phase) an actor may quit the process or decline to proceed or a process is terminated due to internal or external circumstances.

In this manner, any complex process with any number of actors and outcomes can be modeled and illustrated. However, for more complex processes one needs to use the compact notation of a transaction in order to keep the model better managed and controlled. The compact notation is useful for those transactions that are simple (not nesting further transactions). If a compact notation is used, by a convention, the whole transaction is positioned within the confines of the executing actor. Two instances of such a compact modeling are represented in Figure 4a and Figure 4b. In the first case, the two nested transactions are initiated and executed in sequence, and in the second case, the two nested transactions are initiated and executed in parallel.

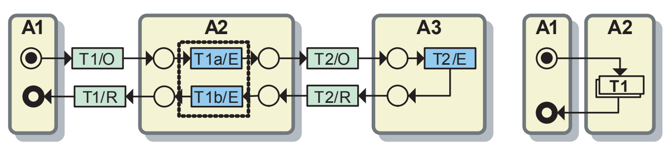
Another notion, a typical phenomenon in process modeling, is of probability of some activities – optional transactions that may take place depending on some conditions. To indicate that a transaction is an optional one, a small decision symbol (diamond shape) is attached to its initiation (connection) point as illustrated in Figure 5a. In order to transform this optional transaction construct into standard Petri net semantics, a traditional XOR-split that could be modeled by one place that

Figure 2. A process diagram of a business transaction: (a) detailed; (b) with the result; (c) compact



(a) (b) (c)

Figure 3. Nested transactions with three actors: (a) detailed; (b) compact



(a) (b)

Figure 4. A model with two nested transactions: (a) in sequence; (b) in parallel

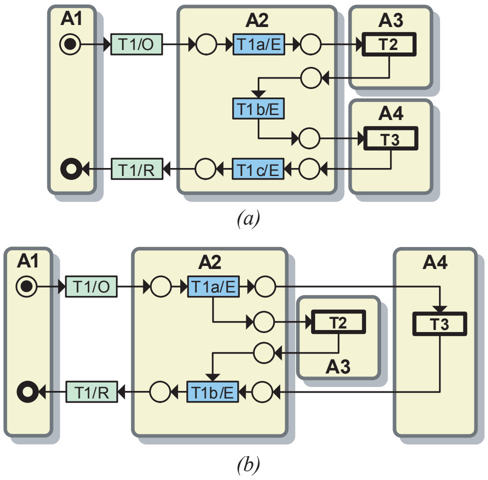
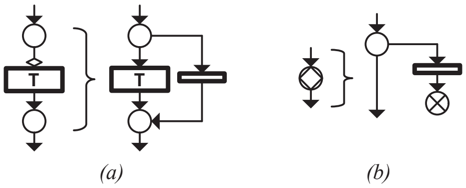


Figure 5. Standard Petri net representation of: (a) an optional transaction; (b) a decision state



leads to two transitions is used. It requires addition of a skip (or dummy) transition as demonstrated in the figure (notice the tiny rectangle with no labels). A dummy transition is meant that it has zero duration and utilizes no resources.

Finally, there are situations that a process may halt and result in a termination. For example, if there is no room available, then the payment transaction is not initiated at all. This situation is modeled through a place identified as "decision state" graphically represented via a circle with the decision symbol (diamond shape) within it, see Figure 5b. As it is seen, for the transformation of a decision state into standard Petri net semantics, a traditional XOR-split that could be modeled by one place that leads to *proceed* or *stop* is used. Depending on the value of the state, the process either proceeds or terminates as indicated by a place filled with a cross.

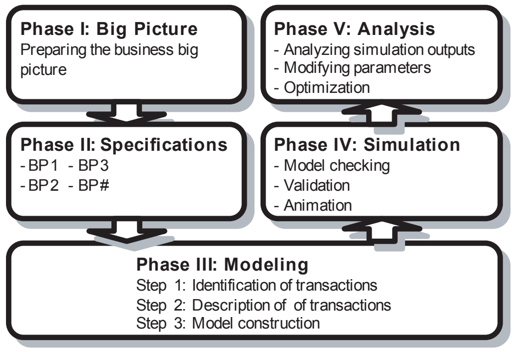
Through these few simplified constructs and mini-models, we aimed to introduce how the proposed method can capture typical situations in business processes, provide sound concept based on communication, and ultimately contribute towards more accurate Business Process Modeling and consequently more adequate IS Design, since the models can be executed several times before it is finalized.

Now that the basic ideas and constructs are introduced, we discuss the underlying framework (guidelines) for deploying the proposed method.

APPLICATION GUIDELINES

Based on practice and application experiments, the following framework (guidelines) was developed. This framework is diagrammatically illustrated in Figure 6, in which both the process flow (block arrows) and feedback loop (circled block arrow) between the phases are depicted. As seen, this is an iterative process where after each simulation and output analysis, the model is refined, some parameters are modified and the experiment is repeated. It may be also required to return to earlier phases (phase I or phase II) for missing pieces of information, if the analysis reveals any flaws or doubts. This is especially important when changes occur for the system under consideration, modifications must be made to the model, and the change impact has to be studied. The entire process consists of the following major phases:

Figure 6. An application framework (guidelines)



Phase I – *Big Picture:* during this phase major processes are identified. Identification of the major processes actually portrays the *"big picture"* of an organization. Also during this phase, scope estimation is conducted – a major process, or focal point, is defined where the main focus will be directed. The perspective taken in this phase considers an organization as a network of business processes (BP). Methods used in this phase are mainly the review of the corporate documents and interview with the business manager if such documentation is lacking or the collected information is vague.

Phase II – *Detailed Picture:* During this phase, each major process of interest is described to fill in the details of the "big picture" previously identified, boundaries of organizational units are defined, and actors and their roles are identified. As a result, an analyst may describe a series of interrelated business processes (BP1, BP2, etc.). Methods used in this phase are mainly based on interviews, observations and review of the documented procedures. However, description can be more articulated so the events, their timelines, and involved actors can be easily distinguished.

Phase III – *Modeling:* For each specific major process of interest:

- **Step 1:** Identification of business transactions using the *transaction concept*.
- **Step 2:** Description of business transactions (actors involved and results created) using the *transaction concept*.
- **Step 3:** Construction of an interaction (process) model using the developed constructs.

Phase IV – *Simulation (Animation, Validation):* In this phase, first the model is checked for absence of semantic flaws and deadlocks. The model is animated for better communication to non-technical users, especially the process manager. Taking the process manager's feedback and input the model is now validated for accuracy and adequacy. Once the model is validated, its behavior is studied through the simulation runs using a discrete-event simulation tools.

Phase V – *Analysis & Improvement:* Finally, the simulation outputs are analyzed for modifications, optimizations, improvements, and comparison of different scenarios. As an objective, at this stage analysts may suggest improvements in the form of redesigning processes, redistributing resources, designing better systems.

This just described framework, and the above discussed modeling method and technique are applied to numerous real world business system. One such a case-study is reported in a separate paper published in these proceedings. Readers interested in the application of the method are referred to the application paper entitled "Business Process Optimization Using Simulation".

CONCLUSION

This paper outlined a modeling method and technique based on the transaction concept and Petri net formalism. This paper studied that the transaction concept derived from the LAP can adequately capture and represent business processes as social systems. The core concept of the transaction concept is its perceiving of an organization and its business processes as social processes involving interacting actors.

The graphical notations adapted for the transaction concept are based on the formal semantics of Petri nets. This allows modelers to build models that can be directly simulated on computer using Petri net based tools. Simulation of the models benefits in many ways: check the models for deadlocks or flaws; study the model dynamic behavior; analyze and compare different set of model parameters.

However, a few things are not fully investigated. First, we have no evidence how the proposed technique will be understood in comparison with conventional techniques (e.g., UML Activity Diagram). Second, how the resulting models can be mapped into well-known simulation tools (e.g., Arena, Extend). Finally, it is not tested how complex business systems can be dealt with using the proposed method.

As a conclusion, this paper is intended to provide theoretical and practical value for business process analysts, modeling and simulation experts, information system designers, and practitioners of modeling and simulation in general.

REFERENCES

Auramäki E, Lehtinen, E., Lyytinen, K. (1988). A Speech Act Based Office Modeling Approach. *ACM Transactions on Office Information Systems*, Vol. 6, No 2, pp. 126-152

Barjis J., & Dietz J.L.G. (1999). Business Process Modeling and Analysis Using GERT Networks. *In the proceedings of the First International Conference on Enterprise Information Systems* (ICEIS'99), Setubal, Portugal, March 27-30, volume II, pp. 748-756.

Barjis, J., & Dietz, J.L.G., Groenewegen, L.P.J. (2000). Organizational Simulation: Concept & Application. Proceedings of the 2000 Summer Computer Simulation Conference, July 16 - 20, 2000, Vancouver, B.C., Canada, pp. 836-841, ISBN: 1-56555-208-3

Barjis, J., & Reichgelt, H. (2006). A Petri Net Based Methodology for Business Process Modeling and Simulation. In the proceedings of the Fourth International Workshop on "Modeling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS'06)", Paphos, Cyprus, May 23-27, 2006. ISBN: 972-8865-49-X

Davenport, T.H., & Short, J.E. (1990). *The New Industrial Engineering: Information Technology and Business Process Redesign*. Sloan Management Review, Summer 1990, Vol. 31 Issue 4, p11, 17p

Dehnert, J., & Aalst, W.M.P. van der (2004). Bridging the Gap Between Business Models and Workflow Specifications. *International Journal of Cooperative Information Systems*, 13(3):289-332.

Dietz J L G (1994) Business Modeling for Business Redesign. *Proceedings of the 27th Hawaii International Conference on System Sciences*, IEEE Computer Society Press. Los Alamitos, pp. 723-732.

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* (EMMSAD'99), Heidelberg, Germany, June 14-15, 1999

Dietz, J.L.G., & Barjis, J. (2000). Petri Net expressions of DEMO Process Models as a rigid foundation for Requirements Engineering. In the proceedings of the 2nd International Conference on Enterprise Information Systems (ICEIS'00), Stafford, UK, July 4-7, pp. 269-274. ISBN: 972-98050-1-6.

Dietz, J.L.G. (2006). *Enterprise Ontology –Theory and Methodology*. Springer

Goldkuhl, G. (1996). Generic business frameworks and action modeling. *In the Proceedings of Conference Language/Action Perspective* '96, Springer Verlag.

Hammer, M., & Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Business

Lehtinen, E., Lyytinen, K. (1986). Action Based Model of Information System. *Information Systems*, Vol. 11, No. 4, pp. 299-317

Medina-Mora R., Winograd T., Flores R., & Flores F. (1992). The Action Workflow Approach to Workflow Management Technology. *In the Proceedings of the Conference on Computer-Supported Cooperative Work*, CSCW'92, ACM Press, New York

Rittgen, P. (2005). Language Action-Based Simulation of Business Processes. *The 10th Anniversary International Working Conference on the Language-Action Perspective on Communication Modeling*. Kiruna, Lapland, Sweden, June 19 – 20.

Stamper, R.K. (1988). MEASUR. University of Twente. Enschede, The Netherlands.

Winograd, T.; Flores, F. (1986). Understanding Computers and Cognition: A New Foundation for Design. Ablex, Norwood.

ENDNOTE

[1] Due to the paper length restriction, this paper is accompanied by another paper "Business Process Optimization Using Simulation", published in these proceedings, where a case-study is reported using the proposed method.