2007

Executable Ontological Business Process Model

J. Barjis
Georgia Southern University, jbarjis@georgiasouthern.edu

EXECUTABLE ONTOLOGICAL BUSINESS PROCESS MODELS

Barjis, Joseph, Georgia Southern University, P.O. Box 8150, Statesboro, GA 30460, USA, JBarjis@GeorgiaSouthern.edu

Abstract

Often business processes modelling is confined to Flow Chart like representation and the models are not amenable to analysis (e.g., simulation) to study their behaviour. As a consequence, validation and verification of such models are merely conducted manually based on the intuition, experience and knowledge of analysts and processes managers. While for small models this may not represent a challenge, for complex processes it is very challenging to avoid inaccuracies – broken flows, missing loops, deadlocks. Execution (or simulation) of models using computer tools would surface most of errors and better enforce model correctness. In this paper we introduce a modelling method and technique aiming at developing executable ontological business process models. The proposed method is theoretically based on the concept of Business Transaction derived from the DEMO Methodology, and diagrammatically on the formal semantics of Petri net. The transaction concept is used for conceiving the organization's social reality, and the formalism of Petri nets as a modelling technique. For the coherence between these two as a whole, a few aspects were to be investigated.

Keywords: business process modelling, ontological modelling, model checking, business process simulation, modelling method, Petri net.

1 INTRODUCTION

Business process modelling is dominantly used to get insight into the operations of an organization, its collaborating actors and interaction with the environment. Modelling has proven to be a valuable tool for learning how processes dynamically evolve, especially when some changes are planned or new IT systems are introduced. As organizational environment evolves, analysts use modelling as a multi-purpose tool from understanding the operations of an existing organization, to redesigning business processes, studying the impacts of changes. In this regard, an interesting research question is how to build models that can be verified, validated, and checked in an automatic fashion using computer tools. In practice, mostly these are accomplished via simulation. But, in order to conduct computer simulation, one needs to build executable models, models based on formal semantics.

Current literature in the field manifests significant re-emergence of interest in the area of business process simulation. This area attracted researchers from divergent perspectives and backgrounds ranging from so called soft to technical and formal dominations (Gladwin & Tumay 1994, Hlupic & Robinson 1998, Harrison 2002, Paul & Serrano 2003, Vreede, Verbraeck & Eijck 2003, Rittgen 2005, Seila 2005). In order to conduct a rigorous analysis of business processes, both modelling and simulation should be applied in concert. Practice shows that with only modelling it is hard to achieve significant insight into a business system, especially into its dynamic behaviour and response to changes (Hlupic & Vreede 2005). On the other hand, despite the abundance of simulation tools, simulation alone may provide little help without profound conceptual modelling preceding it. It would be like "expedition without a map". Lessons endorsed by practice of modelling and simulation suggest, like expedition without a map, simulation without a profound concept (conceptual model) is possible, but it would be very hard, if not impossible, to achieve accurate and precise results.

Business systems are distinguished by their social nature, where human actors interact and collaborate to carry out tasks and fulfil the mission of an organization. As such, business processes are not merely a flow of jobs, tasks, or physical materials to be captured by Flowchart models, but a complex interactive system comprising of actors that communicate, negotiate, and make commitments for carrying out certain tasks. The social nature of business processes entails a fundamentally different perspective to perceive the reality of an organization and the role (responsibility and authority) of its members rather than the approaches used by conventional methods. One such new perspective was introduced in a framework referred to as the Language Action Perspective (Winograd & Flores, 1986). The LAP perspective and its philosophical stance inspired emergence of a number of modelling methodologies and techniques such as SAMPO (Lehtinen & Lyytinen 1986, Auramäki, Lethinen & Lyytinen 1988), Action Workflow model (Medina-Mora, Winograd, Flores & Flores 1992), and BAT (Goldkuhl 1996). Since the main focus in these methodologies is put on capturing communicative acts and just modelling business processes, their underlying modelling techniques do not result in models ready for simulation. Normally, in order to simulate these models, either an additional mapping schema is developed or the models are translated into other state-transitions like diagrams, as it is done in (Dietz & Barjis 1999, Dietz & Barjis 2000). In order to develop simulation ready business process models, this paper introduces a method and discusses the *business transaction concept* as a suitable framework for constructing models of an enterprise. In this paper, we further explore the original works resulted from the LAP (Language Action Perspective) Community by Dietz (1994, 2006).The proposed method deploys a modelling technique based on the formalism of Petri nets. This method is a slightly simplification and further application of CAP Net by Dietz (2006).

The new result reported in this paper is a set of notations and modelling technique that allows building executable business process models based on the DEMO Transaction Concept. So far, a dozen of models were developed and executed (simulated), and as a sample, a screenshot of such a model will be presented at the end of the paper. In general, this paper is hoped to make the following contributions:

1) *Executable models* of an enterprise (business processes) based on the transaction concept derived from the DEMO Methodology. Our contribution is to make the resulting models executable to help analysts with model checking, validation and verification, and studying impacts of changes by testing different scenarios. It should be recognized that this is achieved through simplification of the models.

2) *Compact models* of complex processes. Often, in business process modelling analysts either is not interested in all details, or the process under study is too large to be depicted at detailed level, or the analysts may spotlight part of the process while leave other parts concealed. In these situations, compact modelling where certain activities are compressed into one well defined component would be of highest interest.

3) *The knowledge*, generated as a result, contributes to business process modelling, simulation, modelling methodology, application of modelling and simulation, and advancing the discipline of modelling and simulation in an organizational context.

To the best of our knowledge, this paper is among the first attempts to produce executable models based on the DEMO Transaction Concept without any mapping schema or translation. We have adopted Petri net's formal semantics and graphical notations from the very beginning, so, there wouldn't be a need for intermediate mapping or translation.

2 BACKGROUND STUDY

Both the Transaction Concept and Petri nets have been extensively investigated by different researchers. The transaction concept attracted researchers because of its innovative philosophical stance allowing perceive the reality of social systems, i.e., organizations. Application of Petri nets on the other hand attracted researchers for its formal semantics, logics and simple, but powerful grammar. However, true business process modelling techniques based on Petri net are still lacking. For an example of using Petri net for Business Process Modelling, the reader is referred to (Moldt & Valk 1998). More examples of research in this area can be found in a collection of papers in (Aalst, Desel & Oberweis 1998) or in a comprehensive report of research works at Eindhoven University, Netherlands, (Aalst & Hee, 2002). Most of the Petri net models proposed are dominantly process or workflow oriented rather than business process in the sense of socially interacting and communicating actors. In the framework that we apply Petri net, it is implied that the underlying system is of a social nature, an organization where social actors make requests, commitments, negotiations, and bring about new results. Thus, in the proposed method, the emphasis is placed on the social characteristics of business process that better fits service oriented organizations. In this regard it would be interesting for readers to mention that there are numerous studies on the suitability of certain BPM methods for one or another purpose (or perspective), e.g., for the purpose of business process documentation, business process analysis and design, IS/IT Application design, and so forth. Each of the methods fits well for a certain purpose or from a certain perspective. Bider (2005) based on an extensive analysis of existing methods, states: "There is no universal method of business process modeling suitable for all possible projects in this field". Analogously, we hope that our proposed method would serve as a complement to the variety of existing methods with its suitability for a certain perspective. In particular, the proposed method is well suitable for service oriented business systems with intensive interactions among participating actors, an organization and customers, and across organizational processes.

In contrast to prevailing process-oriented and object-oriented models, the introduced method models not only process flow but also takes into account the social character of the modelled enterprise such as interacting actors (or actor roles), and the nested structure of their activities. This paper further develops the works of (Dietz & Barjis, 2000, Dietz, 2006). Compared to all previous works, as it will be later demonstrated through examples, models developed in this paper are immediately executable for simulation, analysis, model checking, validation, and verification purposes. Any Petri net tools can be used for their analysis.

We believe that the proposed method and its associated notations better contribute towards the requirements that a satisfactory method must meet in order to better communicate the user requirements to the system designers, and in order to adequately visualize the underlying conceptual notions. Some of the features that we hope this method excels in are: rich graphical representation and yet lending to formal analysis; modelling not only actions flow, but also actors' interaction; dealing with the deep (nested) structure of business process; embedded compact modelling.

3 BUSINESS TRANSACTIONS

In the proposed method, the main concept and building block is called a *Transaction* (or business transaction) adapted from the DEMO methodology (Dietz 1994, 2006). According to the DEMO methodology, transaction has a generic pattern that consists of three phases, delivers a new result and is carried out by two actors in a close collaboration. For example, applying for a new insurance policy is a transaction that involves two actor roles (a customer and the insurance company). The deliverable (result) of this transaction is a new policy. The *Transaction Concept* is based on the Language Action Perspective, one of the theories of Information Systems. Throughout this section, we will introduce different properties of a transaction along with the artifacts we have developed to diagrammatically represent the corresponding property that can be used for modeling business processes. The proposed artifacts are based on the Petri net formalism.

Transactions are patterns of interactions and actions, as illustrated in Figure 1a and distinguished by different colours. 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 a 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" button 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", "processing an e-form", "dispensing bills", and "moving an elevator to the corresponding floor".

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, 2006). 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 colours (the Execution phase is represented by a rectangle coloured in blue, or gray in greyscale printout). These three phases are a distinct feature that entails the discussed method as a *business process* modelling technique versus just a *process* modelling. The 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 modelling 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 undermining information about the relevant actors and their role.

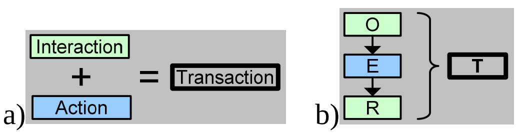


Figure 1. Transaction: a) pattern of action and interaction; b) sequence of three phases (detailed and compact)

In a structured language, a transaction is described according to Table 1, where a transaction is portrayed through the *activity pattern* it represents, its *initiator*, *executor*, and the *result* it delivers (or the new fact it creates). For illustration, policy issuance is described as a single transaction. Since real business processes are an arbitrary chain of transactions with the involvement of numerous actors, it is suggested to conveniently denote transactions by the letter "T" and accordingly number them (T1, T2, T#), and actors by the letter "A" and number them (A1, A2, A#).

Table 1. Transaction description in a structured language

Transaction:	**Name of the activity represented by a transaction**
Initiator	Name of the role that initiates the transaction, e.g., *customer*
Executor	Name of the role that executes the transaction, e.g., *supplier*
Result	The result or fact resulted from the interaction between the customer and supplier

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, representing the result 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 2a depicts a business transaction using the Petri net notations, where each of the three phases (OER) is represented as a transition (rectangle or box). In a compact notation, these three phases are compressed into a single transition, called Transaction (T). In the figure, the start and the end places are marked by different circles. These notations will show helpful when a complex process consisting of several sub-processes is modelled.

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 interactions 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.

From Information System perspective, a transaction diagram should also represent how the created result (data) 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 may be 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.

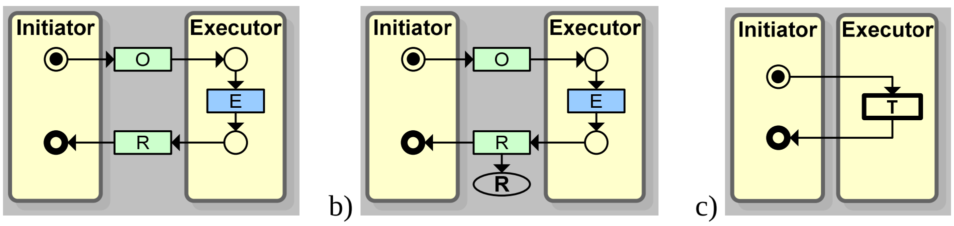
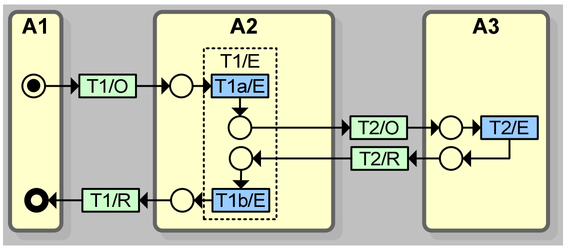
a) b) c)


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

Distinction is made between different types of transaction, simple (causal), composite, and optional transactions. Actors' interactions may be arbitrarily complex, nested, extensive and multilayered (hierarchical). A complex process typically consists of numerous transactions that are chained together and nested into each other. A *Simple* (causal) transaction does not involve (trigger or cause) other transactions during its execution (like in the above figure). It is carried out straightforwardly. In a *composite* transaction, 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 and checks the room availability, but in order to fulfil 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 transaction, 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 carried out between the hotel and a credit card company. Thus, the process rather involves three actors (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, which reveals the deep structure of business process usually ignored or omitted in conventional methods.

One of the limitations in many modelling 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 process for the sake of compactness or for spotlighting a specific part of the process while concealing the other parts. The notion of nesting structure is especially helpful in inter-organizational process modelling 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.

a) 

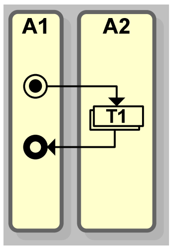
b) 

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

In this manner, complex processes with any number of transactions, actors and outcomes can be modelled and illustrated. However, for more complex processes one needs to use often 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 modelling are represented in Figure 4. In the first case, the nested transactions are initiated and executed in sequence, and in the second case, they are initiated and executed in parallel.

Another notion, a typical phenomenon in process modelling, 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 modelled by one place leading 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.

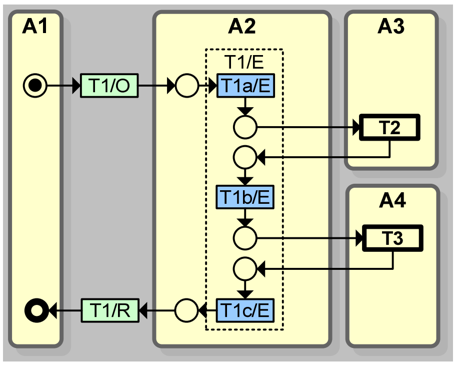
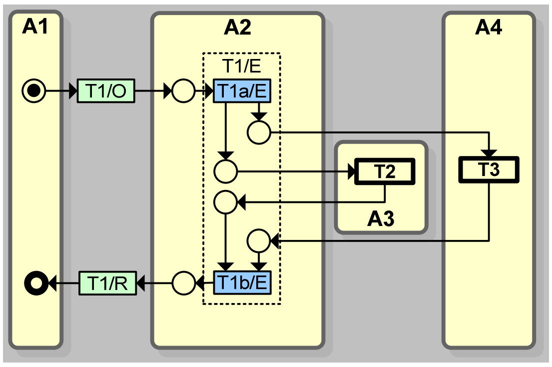
a)  b) 

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

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 modelled 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 modelled by one place leading 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. One additional construct, introduced for practical purpose, is a conditional link (dotted arrow), as depicted in Figure 5c. If a conditional link is needed to connect two places (circles), then it should be represented with an addition of a dummy transition. If it connects two transitions, then it should be added with a dummy place (small circle).

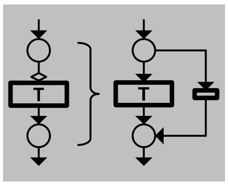
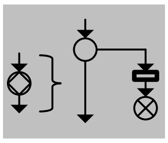
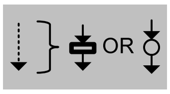
a)  b)  c) 

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

Through these few simplified constructs and mini-models, we aimed to introduce how the proposed method can capture typical situations in a business process, provide sound concept based on communication, and ultimately contribute towards more accurate Business Process Modelling and consequently more adequate EIS Design. Now that the basic ideas and constructs of the proposed method are introduced, in the following section we illustrate how this method can be applied to a simple real world business system.

4 CASE STUDY: FAMILY HEALTH CARE CENTER

The case study reported in this paper was conducted in a family health care centre of over 30 people staff members with a professional manager holding MBA degree. This case study was part of a larger project when the centre was undergoing planning of an Electronic Medical Record system for the first time. The study was conducted to analyze the centre current business process pertaining to healthcare delivery (patient examination process – PEP) to determine the IT components and functionalities needed to enable the centre operations. The presented description below is merely for illustration purpose, not the actual case study that comprises much larger and complicated document.

In order to be examined, a patient needs to make an appointment beforehand. Upon arrival on the appointed day, the patient signs in on the "Check In" sheet at the front desk and waits in the waiting *room to be called by the corresponding physician. As part of examination by physician, according to the procedures, the nurse should see the patient first and conduct general checkups (blood pressure, EKG, basic lab work) and records chief complaint(s), and reason(s) for the visit. After completing this preliminary exam, the doctor examines the patient. After completing the examination, the patient goes to the side-desk to check out, to make payment for the service.*

In most cases, a patient's visit to the doctor represents a routine reason (high blood pressure, diabetes, and/or infections). The FHCC is capable of providing most of the services and treatments a patient may need, including if any basic lab works are needed to be conducted during the physician's examination. Basic lab works can be immediately conducted in the centre. However, in rare cases, patients may need further examination by external healthcare providers (specialist) or advanced diagnostic equipment such as a CAT scan, available elsewhere. In this case, the FHCC schedules an appointment with the external healthcare provider. Some procedures such as a CAT scan may require the insurance company's pre-approval in which case the FHCC first requests pre-approval and then makes the appointment arrangement. Usually, this takes a day or two, and a nurse will make the arrangements. Finally, either the FHCC or the external healthcare provider itself informs the patient about the new appointment.

4.1 Identification of Business Transactions

The above description provides a context for elicitation of business transactions. Making an appointment is the first business transaction in the "patient examination process". By making an appointment, a new fact (result) is created, i.e., a new appointment record is entered into the system. A patient is the initiator of this transaction and the receptionist is the executor. This transaction is a part of the patient examination process because, without an appointment, one cannot be examined. In a similar manner, we identify all other business transactions, as depicted in Table 2:

Table 2. Transactions of the PEP

T1:	**Making an appointment**
Initiator	Patient
Executor	FHCC (Receptionist)
Result	A new appointment is made
T2:	**Conducting examination**
Initiator	Patient
Executor	FHCC (Physician)
Result	The patient is examined
T3:	**Conducting general checkup**
Initiator	FHCC (Physician)
Executor	FHCC (Nurse)
Result	The patient general checkup is completed
T4:	**Paying for the service**
Initiator	FHCC (Business office)
Executor	Patient
Result	The service is paid
T5:	**Examining lab works**
Initiator	FHCC (Physician)
Executor	Technician
Result	Lab works are done
T6:	**Making an external appointment**
Initiator	FHCC (Nurse)
Executor	External specialist
Result	An external appointment is made
T7:	**Arranging insurance pre-approval**
Initiator	FHCC (Nurse)
Executor	Insurance company
Result	Pre-approval is granted

In order to develop a detailed model of the PEP based on Table 2, we first need to draw how many actors are involved in the process. For each actor we draw a stripe-like rectangle. As Figure 6 illustrates, we draw one such rectangle for the Patient role, one for FHCC (a complex actor), one for the Insurance Company and one for the external Specialist. Also, as depicted, we highlight three sub-processes that are parts of the patient examination: Making an appointment (Appointment process); requesting the insurance pre-approval, should the patient need to see an external specialist (Approval process); and handling the billing and payment (Payment process). It is possible that some of the sub-processes are more complicated than showed here. For example, the payment process may involve a number of transactions if one studies all the procedures, rules, and arrangements. But to keep the model of reasonable size, we do not focus on those details. Each of the three sub-processes and the examination process itself has a start point and end point (notice the different circles).

As seen from the figure, three of the transactions are optional (T5, T6 and T7) and, therefore, to each of these transactions an alternative dummy transition is added that will skip the corresponding transaction if the execution condition does not hold. This is done to correspond to the formal semantics of Petri net. The usual flow of the Examination process includes only T1 (making appointment), T2 (conducting examination), T3 (conducting general check-ups).

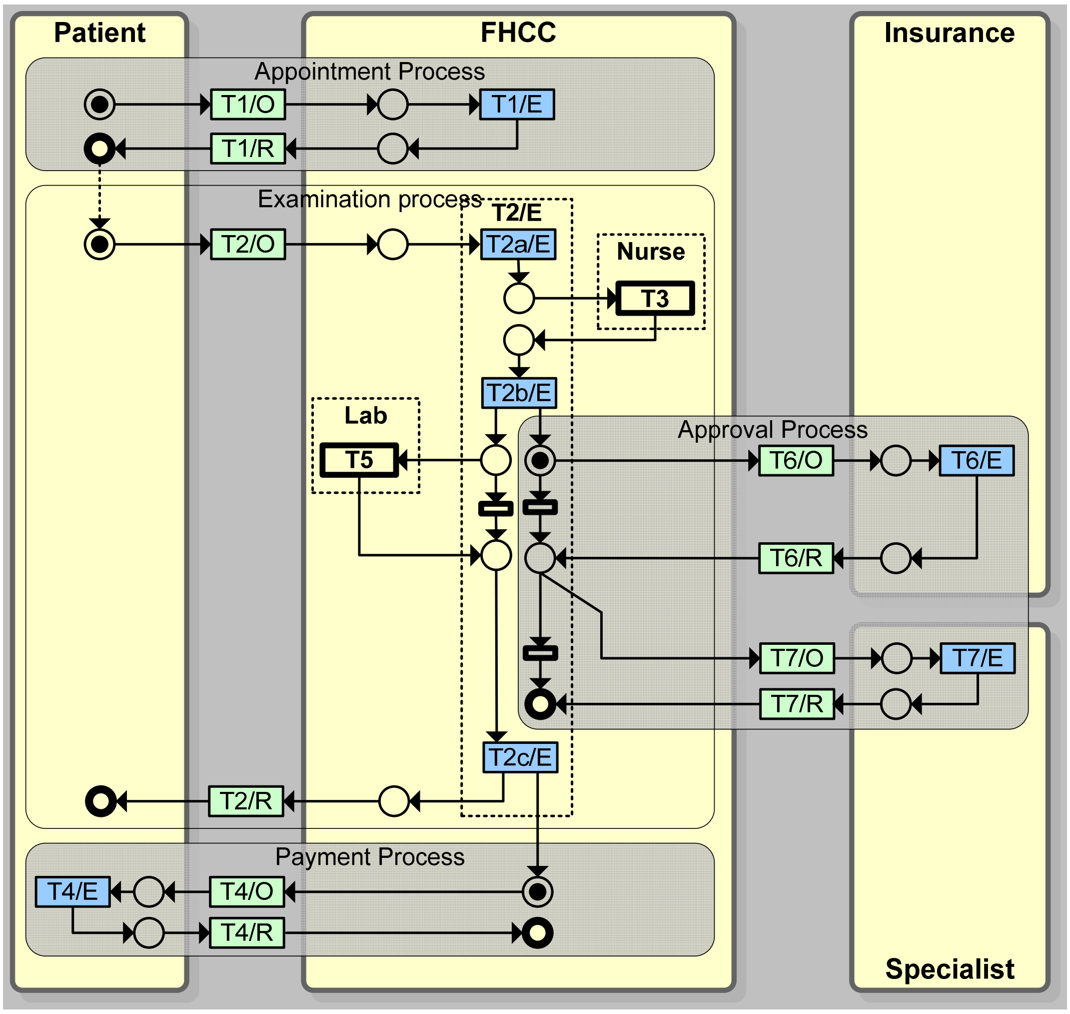


Figure 6. The patient examination detailed model (constructed with MS Visio software)

4.2 Checking the PEP Model

The model constructed and presented in Figure 6 is drawn using MS Visio software using a designed stencil. In order to execute (simulate) this model, any Petri net tool can be used. We used HPSim tool for its simplicity (http://www.winpesim.de/, checked on March 21, 2007). Since this tool does not import MS Visio diagrams, we constructed the model again using HPSim graphical editor. A screenshot of the model with *simulation-in-progress* is shown in Figure 7. Since the screenshot is taken during the simulation run, it also shows moving tokens (animating patients) and enabled transitions (in different colour). The diagram is identical to the one constructed with MS Visio.

We did not aim for extensive analysis of the model. Our purpose in this paper was to simply demonstrate how executable models of an enterprise can be developed using the DEMO Transaction Concept and Petri net notations. We wanted to introduce a model that can be automatically checked for consistency, deadlock, broken links or other validation purposes. It is a starting point for many possible research directions and applications. For more complex investigations, analysts can use other Petri net tools such as CPN Tool widely used within the Petri net community. For more friendly demonstrations for non-technical users, the Arena™ animated simulation tool can be used where the proposed model may serve as a logical model added with the animation components provided by the Arena environment. Since Arena is not a Petri net tool, it requires some transformations to build an animated model.

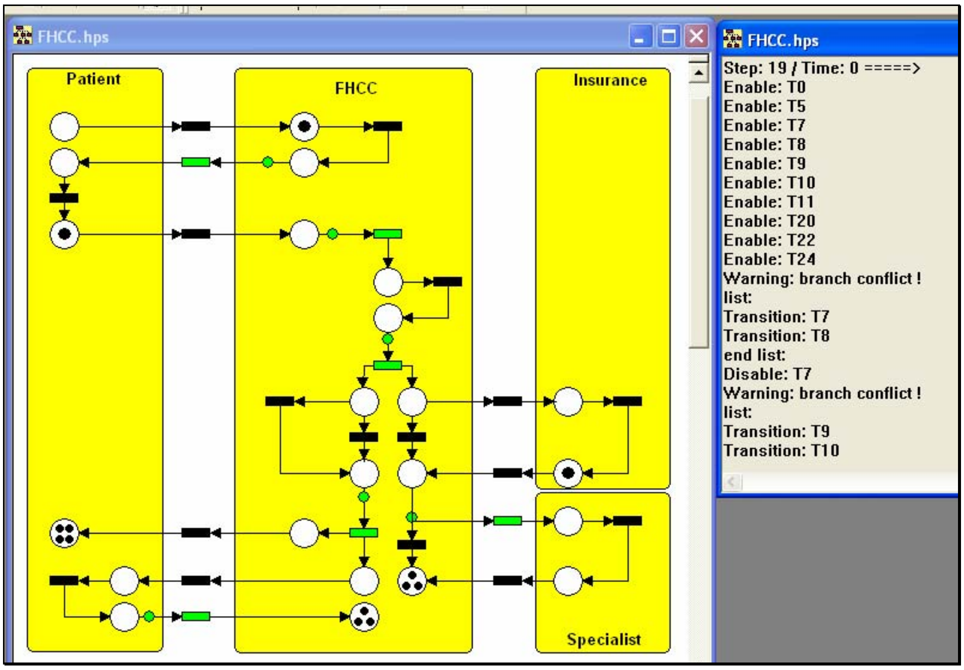


Figure 7. Screenshot of simulation in-progress (HPSim Tool)

5 EVALUATION OF THE PROPOSED METHOD

A new method and artefacts require evaluation to examine their soundness and rigorousness. Hevner, March, Park & Ram (2004) suggest that graphical representation should be very simple, intuitive and easily understandable, at the same time, the accuracy and adequacy of such a representation should not be compromised. Furthermore, Hevner et al. (2004) suggest that methods deploying artefacts should be evaluated using *observational* (e.g., case study) and *experimental* (e.g., simulation) methods. Observational (case study) approach reveals the applicability potential of a method and its artefacts in a given environment and category that are targeted by the method. For example, case studies on different size organizations, different levels of complexity, different levels of abstraction, but all within the same category, e.g., service oriented organizations (insurance companies, healthcare/hospitals, hotels). Experimental (simulation) approach reveals if models can be checked, analyzed and verified. This not only allows models to be checked for consistency, but also eliminates syntactic errors, illustrates dynamic behaviour of models, and lends to formal analysis.

In light of these recommendations, the proposed method has been tested on both observational and experimental bases. A dozen case studies have been conducted using the proposed method and discussed among peer experts in peer-refereed publications. Some of them purposefully were conducted with the involvement of non-experts to not only evaluate the method, but also its complexity and mastering by only lightly trained analysts and system designers. Since the resulting models are based on formal semantics of Petri net, each of the models has been straightforwardly simulated, using Petri net tools, to check the models correctness. All the models were communicated to and discussed with the users and researchers, including within the AMCIS and ICEIS community. Actually, an earlier version of this paper is published by ICEIS. The feedback obtained from modellers, designers, users, and researchers greatly helped to polish and improve the method and eliminate flawed constructs. Although, it is not claimed that this method is now the best and therefore we urge researchers to use and analyze it, but the feedback received confirms that the method has a promising potential within the community. We would appreciate to see some researchers use the method and give it a critical analysis.

6 CONCLUSION

This paper has outlined a method that covers most phases in business process modelling and simulation cycle. Starting from a plain description, identification of business transactions, identification of relevant actors and finally to constructing a model of business process and simulation of the model. In order to illustrate these steps, a simplified version of a case study was discussed.

This paper studied that the transaction concept can be used as a profound concept in studying business processes of dominantly social character – service industry, where strong human interaction has place.

For constructing executable models, the paper introduced a set of graphical notations based on the formal semantics of Petri net. Although the graphical notations used in the model allow intuitive understanding of the underlying processes, it still may be challenging for non-technical users to follow the model. For example, users familiar with Flowchart diagrams may find it very technical that the rectangles are not described by the activities they represent rather than just labelled as T1, T2 or T1/O, T1/E, etc. However, Flowchart diagrams can get extremely sizable even if they represent a moderate size process. The advantage of labelling this way is that models can be kept very compact. Fairly a complex model can be represented on a single sheet of paper, especially if the compact notations are used.

Another limitation of the proposed method is its simplification of the rich social reality. In fact, it would be very challenging to capture all the reality of a social system in a formal model. As the reader may find in CAP Net (Dietz 2006), a transactional process is more sophisticated than just a serendipitous sequence of three phases. Indeed, a transactional process is a complex process of requesting, offering, negotiating, counter offering, declining, and committing. But as modelling in whole, this method is about a delicate compromise between simplicity, expressivity, reality, and formality to try to produce a model that is rich in expression and yet allowing analysis, simulation.

References

Aalst, W. van der, Desel, J., & Oberweis, A. (ed.) (1998). Business Process Management: Models, Techniques and Empirical Studies. Springer-Verlag Berlin Heidelberg New York

Aalst, W. van der; Hee, K. van (2002). Workflow Management: Models, Methods, and Systems, MIT Press.

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

Bider I. (2005). Choosing Approach to Business Process Modeling - Practical Perspective. Journal of Conceptual Modeling, issue 34, Publisher InConcept

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.

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, 4-7 July 2000, pp. 269-274

Dietz, J.L.G. (2006). *Enterprise Ontology –Theory and Methodology*. Springer, ISBN: 3-540-29169-5

Gladwin, B., & Tumay, K. (1994). Modelling Business Processes with Simulation Tools. *In the Proceedings of the 1994 Winter Simulation Conference*, Lake Buena Vista, FL, December 11-14.

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

Harrison, J.R. (2002). Organisational Processes. *Simulation Modelling Practice and Theory*, 10: 249-251

Hevner, A.R., March, S.T., Park, J., & Ram, A. (March, 2004). Design Science in Information Systems Research. *MIS Quarterly*, vol. 28 No. 1, pp. 75-105.

Hlupic, V., & Robinson, S. (1998). Business Process Modelling and Analysis Using Discrete-Event Simulation. *In the Proceedings of the 1998 Winter Simulation Conference*.

Hlupic, V.; de Vreede, G-J. (2005). Business process modelling using discrete-event simulation: current opportunities and future challenges. *International Journal of Simulation and Process Modelling*, 2005 - Vol. 1, No.1/2 pp. 72 – 81

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

Moldt, D., & Valk, R. (1998). Object Oriented Petri Nets on Business Process Modeling. In Aalst, W. van der, Desel, J., & Oberweis, A. (ed.) *Business Process Management: Models, Techniques and Empirical Studies*. Springer-Verlag Berlin Heidelberg New York

Paul, R.J., & Seranno, A. (2003). Simulation for Business Processes and Information Systems Design. *Proceedings of the 2003 Winter Simulation Conference*. New Orleans, Louisiana, December 7-10

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

Seila, A.F. (2005). The case for a standard model description for process simulation. *International Journal of Simulation and Process Modelling*, 2005 - Vol. 1, No.1/2 pp. 26 – 34

Vreede, G.J. de; Verbraeck, A.; Eijck. D.T.T. van (2003) "An Arena Simulation Library for Business Process Modelling". *Simulation*, Vol 79, No. 1, pp. 35-42.

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