IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITB12602

This paper appears in the book, *Emerging Trends and Challenges in Information Technology Management, Volume 1 and Volume 2* edited by Mehdi Khosrow-Pour © 2006, Idea Group Inc.

Using Petri Nets to Represent Cross-Departmental Business Processes

Joseph Barjis, Han Reichgelt, Paul Forté & Chris Small
Georgia Southern University, PO Box 8150, Statesboro, GA 30460, P 912-486-7756,
{jbarjis, han, pforte, csmall2}@georgiasouthern.edu

ABSTRACT

Business process modeling is an important tool in the design and creation of information systems that deliver business value. Business processes often require the exchange of information across departmental boundaries. However, this fact has received relatively little attention and, as a result, many of the tools that have been developed for business process modeling do not allow analysts to easily model such business transactions. This paper describes a case study conducted to model business processes that crossed departmental boundaries in the department of university housing of Georgia Southern University, and shows that such processes can be naturally represented in Petri nets.

INTRODUCTION

In a provocative article, Carr (2003) argues that, even as the global expenditure on information technology (IT) by businesses approaches 2 trillion dollar per year, IT no longer matters in that organizations can no longer rely on IT to give them a competitive advantage in terms of either increased productivity or enhanced quality of their products or services. While there has been a widespread debate of Carr's proposition, with many authors arguing forcefully against Carr's main thesis, there seems to be a consensus that IT on its own indeed does not matter, but that for an IT application to be successful, one must take into account the business processes that the application is expected to support (Smith and Finger, 2003).

Given the importance of business processes for the likely success of IT application, it is clearly important that the business analyst be able to clearly and accurately model business processes. An important element of business process modeling, and one that has received little attention, is the modeling of processes that cross departmental boundaries. This paper is an attempt at addressing this issue and describes a case study in which we use the notion of a business transaction, introduced by Dietz (1999), to conceptualize business processes and use Petri nets, introduced by Peterson (1980), to graphically model business transactions. Our approach is called TOP for Transaction-Based Petri net methodology.

The paper first introduces the concept of a business transaction and then describes Petri nets, before it presents the case study.

THE CONCEPT OF A BUSINESS TRANSACTION

One of the central concepts in TOP is that of a business transaction. The concept is due to Dietz (1999, www.demo.nl) and originated in an approach called DEMO (Design & Engineering Methodology for Organizations formerly known as Dynamic Essential Modeling of Organization.) The concept is based on the idea that an organization and its underlying business processes can be analyzed as a network of business transactions, which jointly constitute the essence of this organization. Following the *Language/Action Perspective* (Dietz, 1999; Goldkuhl *et al.*, 1998), business transactions are conceptualized as patterns of communication, which lead to certain actions that, in turn, create new facts, deliver results, and so accomplish the mission of an organization. Moreover, communication is not merely an exchange of information but also includes patterns of negotiation, coordination, agreement and commitment.

Each business transaction encompasses communications or interactions, and actions, as illustrated in figure 1. The *action* is the core of a business transaction and represents an activity that changes the state of the world or creates a new fact. An *interaction* is either the initiation of an action or the communication of the result of an action.

A business transaction consists of an interaction, typically a request from one actor, called the *initiator* to another, called the *executor*, followed by an action, typically the executor complying with the request by the initiator, followed by a second interaction, typically the communication of the result of the action by the executor to the initiator. A business transaction thus consists of three phases, namely the *order phase*, the *execution phase*, and the *result phase*, abbreviated as O, E and R respectively. A business transaction therefore consists of an OER cycle. As an illustration, consider the process of customer applying for a home mortgage to a loan officer in a bank. The order phase consists of the initial interaction which the customer (the initiator) communicates with the officer (the executor) to request a loan and submits an application. In the execution phase, the officer performs the action of processing the application and making a decision. The result phase consists of a second interaction in which the officer communicates his or her decision to the customer.

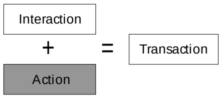
It is useful to distinguish between simple and complex transactions. Simple transactions consist of a single OER cycle and do not trigger or cause other transactions during their execution. Complex transactions, on the other hand, trigger other transactions, typically in the execution phase. Thus, a more detailed analysis of the above mortgage application example would indicate that the execution phase in which the bank officer processes the application triggers a number of additional business transactions, such as a credit check, which itself is a business transaction with an external authorized credit reporting agency.

Actors can be human actors, software agents or machines. For example, if the mortgage application is submitted online, a software agent, rather than a human loan officer, will collect the relevant data from the individual requesting the mortgage and process the application and make preliminary estimates, which it may then submit to a human loan officer for subsequent approval.

With this explanation in mind, we can define the notion of a business transaction as follows:

Definition: A business transaction is a generic pattern of activity carried out between two actors, the initiator and the executor. The activity is

Figure 1. The business transaction concept



carried out in three phases, namely the order phase, in which the initiator makes request to the executor, the execution phase, in which the executor performs some action, which changes the state of the world in some way, and the result phase in which the executor communicates the results of the execution phase to the initiator. The order and result phases are interactions and the execution phase is an action.

We can now redefine the mortgage and credit check example as follows:

Transaction T1:	Request mortgage
Initiator:	Customer
Executor:	Officer
Result:	Mortgage has been approved or declined

Transaction T2:	Check credit
Initiator:	Officer
Executor:	Credit agency
Result:	Credit report has been generated

T2 is triggered during T1, and T1 is therefore a complex transaction

The above example also illustrates that the initiation, execution, or completion of a business transaction may lead to the initiation and execution of new transactions. In this way transactions are chained into arbitrarily large structures, called *business processes* (Dietz, 1999). We use this observation to define the concept of a business process as follows:

Definition: A business process is a network of interrelated business transactions that delivers value (good or service) to a customer and that has a single starting point and a single end point. It starts with a request made by an actor and ends with a result being communicated to the same actor. A business process typically consists of a single overarching transaction that contains any number of nested transactions.

PETRI NETS

TOP uses Petri Nets to model business processes and business transactions. Petri nets are a graphical and mathematical modeling tool that is particularly well suited for discrete event simulation. There are many different types of Petri Net, including elementary Petri nets (Peterson, 1981), High-Level Petri Nets (Reisig and Rozenberg, 1998), Colored Petri Net (Jensen, 1997), Stochastic Petri Nets (Haas, 2002), Hierarchical Petri Nets, Timed Petri Nets, Predict /Transition Nets (Pr/T-net), Workflow Petri Net (Aalst and Hee, 2002), and so on. Space limitations do not permit us to discuss each in full detail, and we have to restrict ourselves to a brief introduction to Petri nets.

In its most basic form, Petri nets consists of *places* (or *states)*, *transitions* and *directed arcs*, as depicted in figure 2 Places can contain one or more tokens, indicating whether the place is active. Graphically, places are represented by circles (or ellipses), transitions by rectangles (or bars), and tokens by black dots (or numbers).

Each Petri net has one or more starting places and one or more end places. A starting place has no arc coming into it, while an end place has no arcs emanating from it. Any arc emanating from a place must lead to a transition, and arcs emanating from transitions must terminate in a place. Tokens are used in the simulation of events. An event is simulated by selecting an active place, i.e. a place with a token in it, removing it from that place, and inserting tokens in all places that can be directly reached from this place via a transition. Since a place may contain more than one token, the place from which the transitions originated is not necessarily no longer active afterwards.

Figure 2. Graphical notations of Petri nets

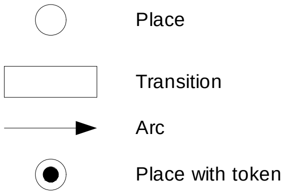
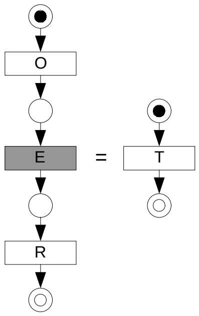


Figure 3. The transaction structure using Petri net diagram



Petri nets have proven especially useful for applications that require the modeling of concurrency and synchronization. Also, Petri nets allow for hierarchical representations, where a chain of place-transition-place triples is combined into a single place-transition-place triple. As we shall see below, this feature is particularly useful for modeling complex processes.

Figure 3 illustrates the use of a Petri net to model the order-execution-result cycle that we introduced in the previous section. Since the order and result phase are interactions and the execution phase an action, O and R are represented in a different color than E.

Figure 3 also illustrates the use of a hierarchical representation. On the left side of figure 3, a transaction is represented as a sequence of the three phases, while, for compactness, the right side of the figure compresses the three phases into a single transition called a transaction. The use of hierarchical representations allows one to hide details that are not necessary for a particular level of analysis. In the remainder of this paper, we will refer to a transaction whose details are hidden and which may contain any number of embedded transactions as a “super transaction”.

CASE EXAMPLE

To illustrate TOP, we conducted a case study in the Department of University Housing (referred as the “Housing Office”) of Georgia Southern University. The housing office serves about 4,000 campus residents. Although it performs various functions, the focus of this paper is on the Room Assignment Process, as this process crosses departmental boundaries in that it depends on inputs from processes that take place elsewhere within the institution.

The Room Assignment Business Process

A new student, accepted to Georgia Southern University, applies for housing by visiting the housing web site and logging into the Residential Management System (RMS). The login account is automatically created when the student has been entered into the campus student database, WINGS, a separate system maintained by the WINGS units located in the Registrar’s Office. WINGS acts as a gateway into the University’s information systems, and automatically creates a number of accounts, including an account on the University’s online course registration system, an email account, an account on the learning management system used by the institution (WebCT Vista) and an RMS login account.

After logging into the RMS, the student is prompted to create a profile and apply for a room. The application process prompts students to choose the term for which they are applying, and list their preferences for the halls/communities in which they do or do not wish to live. Since the housing office requires students to pay a $300 refundable security deposit, students who have completed their application are redirected to a billing page in which they either give credit card information or receive instructions for alternative payments. Once the housing office has received payment, the application is complete, and students may log in to RMS at any time and view their application and/or assignment status.

Cross-Departmental Process Modeling

The above description shows that there are three departments involved in the overall chain of business processes, namely the Registrar's Office, with responsibility for accepting and registering new students, the WINGS Office responsible for creating different type of student accounts, and the Housing Office itself. The flow of processes between the offices starts in the Registrar Office, proceeds to the WINGS Office and terminates in the Housing Office. This is depicted in figure 4.

The circles with back dots denote the starting point of each distinct process, while the circles with hole inside denote the end point. Rectangles with tick gray lines represent the process boundary line. The dotted arrows represent the interrelation of the processes, and indicate that the process at the head of the arrow can only start if the process at the tail of the arrow has been completed.

Identification of Business Transactions

We can now analyze each of the processes from a transactional perspective. Since the focus of this case study is on the business processes within the housing office, we do not analyze the processes in the other two offices in any great detail.

A process of registering a new student by the Registrar's Office is represented by one super transaction:

TR1	Registering a student
Initiator	Student
Executor	Registrar's office
Result	Student has been registered

The creation of housing account by the WINGS Office is also represented by a super transaction:

TW1	Creating an account
Initiator	Registrar's office
Executor	WINGS office
Result	A new account has been created

The room assignment process starts by the student creating a student profile and completing a form in which he or she indicates his or her preferences. However, before the form can be submitted, the system forces the student to complete another transaction, namely to make a one month deposit. So, the deposit processes is nested inside the room application process. In other words, the room application transaction invokes the deposit transaction, which must complete before the room application process can be completed. The following are these three business transactions:

Figure 4. Cross-departmental processes

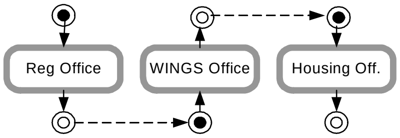
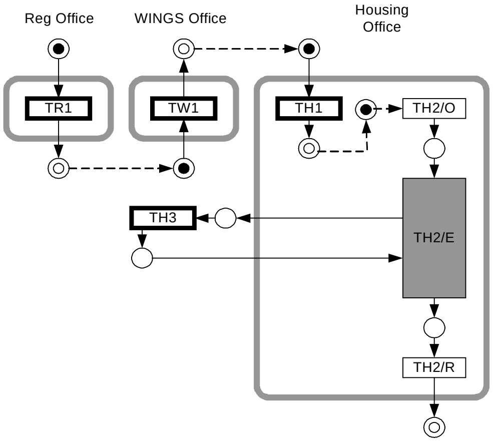


Figure 5. Detailed business processes



TH1	Creating a profile
Initiator	Housing office
Executor	Student
Result	A profile has been created
TH2	Applying for a room
Initiator	Student
Executor	Housing office
Result	An application is submitted
TH3	Making a deposit
Initiator	Housing office
Executor	Student
Result	A deposit has been made

We can now refine the analysis in figure 4, as shown in figure 5.

The figure shows that the deposit transaction is initiated from the room assignment process but executed outside its boundary. The result of this transaction communicated back to the initiator, and so completes the execution of TH2.

It should be noted that figure 5 is a simplified version of the room assignment business process. We have omitted quite few a transactions, including the generation of a master list of assignments and the inspection of the list for errors, as the details are not relevant to the primary purpose of the paper, which was to illustrate cross departmental process modeling where the result of one process completed on one department serves as an input or condition for a process completed in another department.

CONCLUSION

This paper introduced a small example to show how business transactions flow within and across departmental boundaries. It shows that even a simple process cannot be completely understood unless its relationships to other processes within the wider organization or business environment are elucidated. The example also illustrates the utility of the concept of a business transaction as an atomic business process and the power of Petri nets to model business processes.

In future work, the authors are planning to look at more complex processes that cross boundaries of departments and organizations. To support this effort, tools will be developed that can accommodate large models in a limited space by representing processes as super transactions but allowing the user to reveal their details as and when necessary.

REFERENCES

Aalst, W. van der; Hee, K. van (2002). Workflow Management: Models, Methods, and Systems, MIT Press, ISBN: 0-262-01189-1.

Carr, Nicholas G. (2003) IT Doesn't Matter. Harvard Business Review, May 2003, Vol. 81 Issue 5

DEMO: www.demo.nl

Dietz, J.L.G. (1999). Understanding and modeling business processes with DEMO. In the Proceedings of the Annual International Conference on Conceptual Modeling (ER'99), Paris, November.

Goldkuhl, G.; Lind, M.; Seigerroth, U. (1998). The Language Action Perspective on Communication Modelling. In the Proceedings of the Third International Workshop on LAP, Department of Informatics, Jönköping International Business School, Sweden.

Haas, P. (2002) Stochastic Petri Nets: Modeling, Stability, Simulation. Springer-Verlag, New York, ISBN: 0-387-95445-7.

Jensen, K. (1997) Colored Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag, 2nd ed. ISBN: 3-540-60943-1.

Jensen, K. (1997) Colored Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 2, Analysis Methods. Monographs in Theoretical Computer Science, Springer-Verlag, 2nd ed. ISBN: 3-540-58276-2.

Peterson, J. L. (1981) Petri Net Theory and the Modeling of Systems. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Reisig, W. G. Rozenberg (Eds.) (1998) Lectures on Petri Nets I: Basic Models.

Advances in Petri Nets, Lecture Notes in Computer Science, vol. 1491, Springer-Verlag, ISBN: 3-540-65306-6.

Smith, H.; Finger, P. (2003) IT Doesn't Matter Business Processes Do. Meghan-Kiffer Press, ISBN 0-929652-35-5.