Business Process Optimization Using Simulation

Joseph Barjis, Georgia Southern University, P.O. Box 8150, Statesboro, GA 30460, USA; E-mail: JBarjis@georgiasouthern.edu
Matt Hall, Georgia Southern University, P.O. Box 8150, Statesboro, GA 30460, USA; E-mail: MHall31@georgiasouthern.edu

ABSTRACT

In this paper we study a Pharmacy that recently went through a major IT initiative. With the acquisition of the new software, the Pharmacy had more flexibility to rearrange its flow of activities. We used several "what if" scenarios and applied a simulation tool to study the impact of each scenario on the performance of the pharmacy and search for a maximum optimization given the resources available. An issue that we investigated was that a simulation tool would be of less use if it is not preceded by a profound conceptual modeling. In turn, modeling alone is not sufficient to get insight into the dynamic behavior of a system. For the pharmacy business process modeling we used a method that is introduced and discussed in a separate paper.

Note: Due to space restriction, this paper represents application part of another paper "Business Processes Modeling as Social Systems", published in these proceedings, where the modeling method itself and its constructs are discussed. For the understanding of this paper, readers are strongly recommended to read the first paper where the method is described and discussed.

INTRODUCTION

For a thorough analysis and study of business processes, both modeling and simulation should play in concert. Only modeling may not reveal sufficient information about the processes (Hlupic & Vreede, 2005). For significant results and accuracy, business process optimization and modeling need simulation. On the other hand, only simulation provides little help if there is no profound conceptual modeling preceding it. It would be like "expedition without a map".

Continuous competition, increasing capabilities and features of emerging technologies and growing customer demands require organizations to keep current, i.e., adapt changes in order to gain market share, improve performance, increase customers and incorporate "best practices". In such an environment, process optimization is no longer a competitive advantage but a requirement of doing business (Rivera & Marovich, 2001). Obviously any change is risky and may invoke serious consequences for organizations. Early mitigation of these risks is undoubtedly a prerequisite of success and survival in risky changes with unforeseen variables. Here is where business process simulation plays a significant role in process optimization. Simulation is a safe and inexpensive way of studying the impact of changes and revealing hidden behaviors of a complex system.

Since changes would be a driving force for *the 21st century enterprises*, business process optimization and simulation is not a question of *"to be or not to be"*, but a navigational compass to set the right course for sailing into the storms of rapid change. According to some experts (e.g.: Paul & Serrano, 2003; Seila, 2005; Kleijnen, 2005; Hlupic & Vreede, 2005) the potential and full capacity of business process simulation still have yet to be revealed. Although one may argue that the diligent efforts of researchers have fairly advanced the research in this area, simulation as an effective tool should still be widely accepted and adapted by businesses.

Since this paper is an application part of another paper that contains detailed description of the method, here we only focus on the application – case study.

CASE STUDY: PHARMACY

The case study reported here is not intended to be exhaustive, it is a simplified version to demonstrate how the proposed method is capable of capturing the dynamic behaviour of business processes and serve an input for simulation. This case study was conducted at a time when a Pharmacy was planning to acquire and implement a new system and extend its business with e-commerce. This case study using modelling and simulation, was assumed to provide an insight into the business and help to understand the Pharmacy's operations and requirements for a new system.

Prescription Filling Process

Upon arrival at the Pharmacy a patient proceeds to the pharmacy counter and requests prescription refilling. If it is a new patient, the technician asks the patient to fill out a short information sheet, which includes information such as the patient's name, address, telephone number, allergies, and whether or not the patient has any type of insurance or medicine coverage. When the profile is created, the technician selects medicine according to the prescription.

The software automatically checks the current medicine for interactions. Then the user transmits a claim to the patient's insurance. If no insurance coverage, a cash price is assigned.

The computer generates a label and sends the information to the 'robot' for automatic filling. The medicine is then checked one final time visually by a pharmacist. Once verified, the prescription is bagged and then sent out to the cashier for pick-up by the patient. The entire process normally takes no more than 10-15 minutes. The end of this process is related to another process called inventory control. Inventory must be accurately maintained because QuickScrip uses an automated ordering system which examines the current quantities of medicine in stock and networks with the wholesaler company to ensure proper levels are maintained. Although the inventory control process and its interrelation with the prescription filling process were also studied in the case study, here we skip this part due to space restrictions.

Identification of Business Transactions

The process of "Prescription Filling" starts when a patient presents a prescription to be filled. Thus, the first transaction (T1) is "prescription filling". Actually, this is a super transaction that nests many other transactions. This transaction is initiated by a "patient" and executed by the "pharmacist". The result of this transaction is a filled prescription. In this manner we identify all other transactions:

T1:	prescription filling
Initiator:	patient
Executor:	pharmacist
Result:	prescription is filled
T2:	creating profile
Initiator:	pharmacist
Executor:	patient
Result:	profile is created
T3:	checking medicine interaction
Initiator:	pharmacists (software agent)
Executor:	QuickScrip
Result:	interaction is checked

T4: claim processing
Initiator: pharmacist
Executor: insurance company
Result: claim is processed

T5: automatic dispensing
Initiator: pharmacist
Executor: robot
Result: medicine is dispensed into bottle

T6: paying for the medicine
Initiator: pharmacist
Executor: patient
Result: medicine is paid

Now, based on the above transaction, we build a detailed model as shown in Figure 1. By disclosing Transaction T1 (splitting its three phases), all other nested transactions are revealed. This figure shows all the transactions as an interrelated network. It also shows that once medicine is issued (T1/R), the inventory control process is activated. As the inventory control process is out of the scope, which itself is a network of transactions, we just illustrate it as a composite transaction (T#).

Within the scope of our model, only Transaction T1 is a composite transaction and, therefore, we decompose it. All other transactions (T2, T3, T4, T5 and T6) are simple transactions and, therefore, they are shown in a compact form to keep the model compact.

In Figure 1, the Pharmacy is considered as a composite actor delegating the role of a few other actors such as "pharmacist", "technician", "robot (A2)" and "software agent (A1)" for checking medicine interaction. In order to better understand the above figure, it should be read from left to right and from the top to down, just as the arrows indicate. It would be easier if the reader has a list of the transactions, previously identified, ready when reading the model: The patient requests prescription filling (T1/O) and with this request the execution by a pharmacist or technician starts (T1a/E). If it is a new patient, the technician requests them to fill in a form to create a new profile (T2). This is an optional transaction indicated with a small diamond-shape at the connection point. Then, within the pharmacy system (QuickScrip), a request is made to check the current medicine for any interaction (T3) (if an interaction is detected, the process terminates here). Through an online application, the claim for this medicine is transmitted to the patient's insurance company to define the price of the medicine (T4), if the patient is covered by an insurance plan. Then the robot is instructed to fill in the prescription (T5). At this point the patient is requested to make their portion of the payment or arrange for later billing (T6), and only then the medicine is issued to the patient and the process is completed (T1/R). Notice, the completion of this process triggers a transaction in the inventory control process (T#) making sure the issued medicine is subtracted from the inventory and checks if this medicine should be ordered for restocking.

Figure 1. The pharmacy detailed model (constructed with MS Visio software)

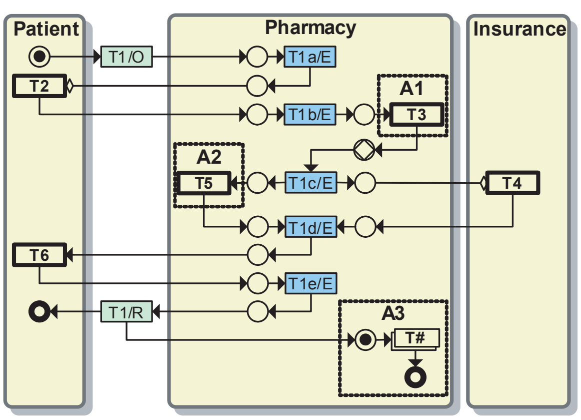
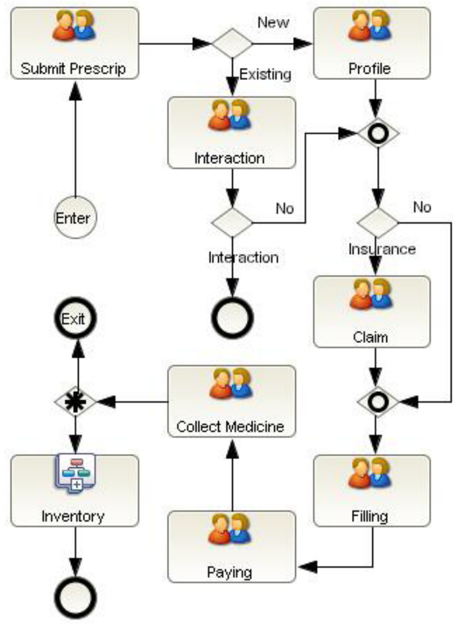


Figure 2. Screenshot of the "pharmacy" simulation model



SIMULATING THE PHARMACY MODEL

We developed a simulation model of the Pharmacy based on the detailed model presented in the previous section. For the simulation purpose we use Savvion Process Modeler to show the usefulness of the developed models as an input to be used by simulation packages other than Petri net tools. This way, it is demonstrated that the Petri net models are easily portable to different environments and can be adequately represented by other models, although some minor adaptations are needed. By adequately we admit one essential compromise – the interaction part of the models are largely omitted and the model is reduced to merely execution phases, where the actual actions take place. One of the typical adaptations required, concerns composite transactions. A composite transaction should have both start and finish parts, e.g., the "prescription filling (T1)" is divided into submitting a prescription for filling (Submit prescription) and collecting the medicine (Collect medicine) parts, while all other transactions can be represented as a single activity, as shown in Figure 2. This figure depicts an animated simulation model of the Pharmacy using Savvion Process Modeler.

Savvion Process Modeler has its own graphical editor for constructing models using a set of artifacts. The figure below is a screenshot of the Pharmacy model in the Savvion Process Modeler graphical editor.

The model contains 6 transactions each representing an atomic process. In the Savvion simulation model above, these transactions are represented as worksteps (grey rectangles: e.g., Submit & Collect, Profile, Interaction, Claim, Filling and Paying). Each workstep corresponds to one transaction (more precisely, the execution phase of a transaction), however, the first transaction, the composite one, is divided into two parts (Submit, Collect). In addition, the simulation model illustrates how the last transaction 'Paying' is linked to the 'inventory control' process. This process is represented through a *subprocess* element without revelation of its inner structure (transactions), which by itself is a model.

For accurate analysis and comparison, Savvion Business Modeler provides analysts with features such as generating a report on the simulation results in either HTML or Excel format. The following few lines and the corresponding Table 1 are excerpts from a complete simulation report (2-4 pages). These excerpts are about the bottlenecks in the process and warning about activities that never occurred:

***Filling*:** A bottleneck was detected for Filling (avg. queue length was 0.15, max. queue length was 5.0).

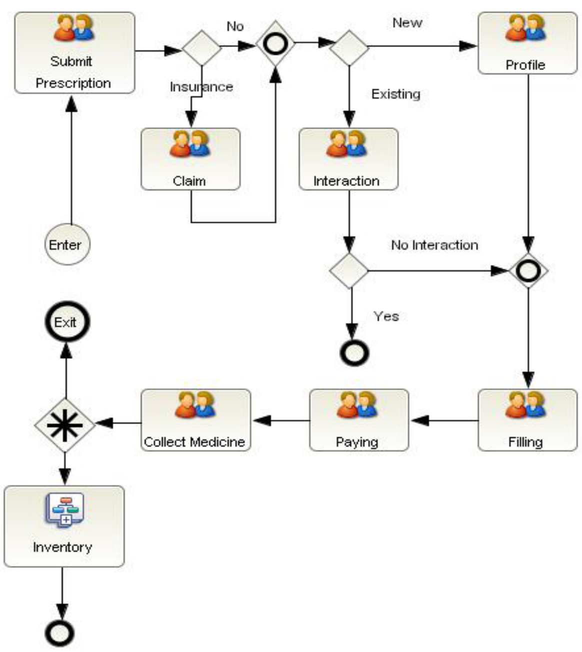
Table 1. A part of the simulation report

Pharmacy					
Scenario	*Original*				
Instances	*60 instances*				
Activity	Performer	Occurs	Waiting Time	Time to Complete	Total Time
Claim	Insurance	57	1:58:00	4:45:00	6:43:00
Collect Medicine	Patient	57	2:10:00	0:57:00	3:07:00
Filling	Robot	57	0:00:00	4:45:00	4:45:00
Interaction	Technician	57	0:00:00	0:57:00	0:57:00
Paying	Patient	57	1:23:00	4:45:00	6:08:00
Profile	Patient	3	0:05:00	0:15:00	0:20:00
Submit Prescription	Patient	60	1:57:00	1:00:00	2:57:00
Inventory	Generic	57	0:00:00	0:00:00	0:00:00

***Profile*:** Profile was never activated. Try increasing the number of instances.
***Submit*:** A bottleneck was detected for Submit (avg. queue length was 0.029, max. queue length was 1.0).

A complete report of the simulation outcome includes 2-4 pages in HTML format detailing average times, queues, busy and idle times for each performer, and other related information about the model behavior. All this is generated in a series of tables one of which is illustrated above (we skip the rest due to space restriction). The simulation package also has a kind of expert system that provides recommendations of how to improve the process or which components need to be redesigned for better performance.

Figure 3. Screenshot of the revised "pharmacy" simulation model



"What if" Scenario A

To improve the model, reduce bottlenecks and decrease wait time we put the model through several "what if" scenarios two of them reported in this paper. In our first such scenario (see Figure 3) the issue of the bottleneck during the filing of insurance claims was addressed. Our first step was to assess the current sequence of the "Claim Process" and optimize its position in the flow. The claim process being in itself a complex process involving the insurance company may take longer than some other actions and resulted in bottlenecks at its beginning and also was thought to be responsible for other process in the model. Our evaluation led us to believe that placing the claim at the beginning of the model instead of in the middle could help to reduce the bottleneck in that location and by doing so on a subsequent simulation run the model was made more efficient. Included below are excerpts from the results of the simulation run.

Collect: A bottleneck was detected for Collect Medicine (avg. queue length was 0.032, max. queue length was 1.0)
Profile: A bottleneck was detected for Profile (avg. queue length was 0.0090, max. queue length was 1.0)
Submit: A bottleneck was detected for Submit (avg. queue length was 0.071, max. queue length was 1.0)

"What if" Scenario B

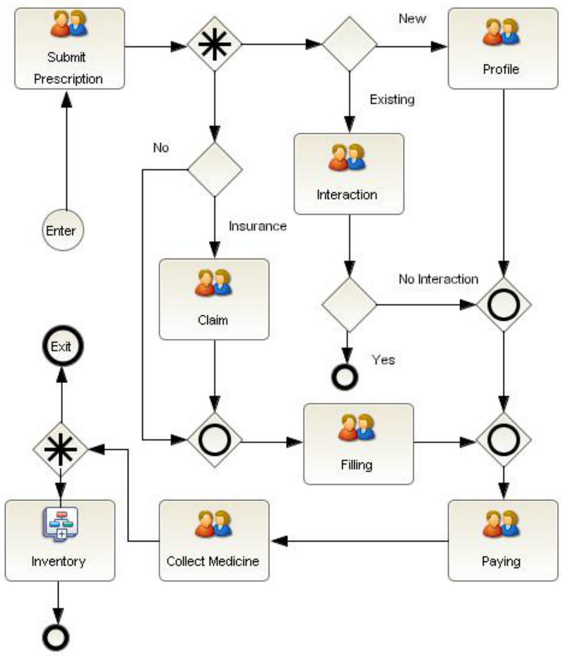
To further reduce wait time and the overall length of the process we decided to incorporate mass parallelization (see Figure 4). Utilizing this feature allows for the claim to begin processing at the start of the simulation and to have the other performers continue on with the interaction check and information profile for new customers. This model proved to be an improvement for overall processing time compared with the original and the previous scenario.

Filling: A bottleneck was detected for Filling (avg. queue length was 0.0010, max. queue length was 1.0)
Paying: A bottleneck was detected for Paying (avg. queue length was 0.047, max. queue length was 1.0)
Submit: A bottleneck was detected for Submit (avg. queue length was 0.022, max. queue length was 1.0)

Comparison of "What if" Scenarios

As we revised the models it was easy to gather data from the software that allowed us to gauge the effectiveness of each model. Our efforts in creating and revising the models were in order to determine a successful way to reduce wait time for customers to streamline the business process. Each "what if" scenario was investigated with the same set of variables regarding the number of customers arriving as well as their frequency. The Original scenario (model) was compared to each of the two "what if" scenarios to see which model was the most effective (see Table 3). The Original model had an overall simulation run time of 15 Hours

Figure 4. Screenshot of a revised "pharmacy" simulation model incorporating mass parallelization



and 21 minutes. There were six Bottlenecks detected the sum of which created an average waiting time of 0.3697 hours for each customer.

The second model in which the insurance claim process was initiated immediately upon the submission of a prescription had a total run time of 15 Hours. In this model three Bottlenecks were detected the sum of which created an average waiting time of 0.106 hours for each customer. This scenario reduced both overall runtime and the waiting time for customers.

The third scenario incorporated mass parallelization into the model. The overall simulation run time was 14 Hours and 20 minutes. There were four bottlenecks detected, their averages totaling 0.274 hours. This model was found to be effective by reducing the overall time a full hour compared to the Original scenario and also reducing the wait time when compared to the Original scenario.

While both of the scenarios proved effective and each had their own strengths it is always up to the business to decide which is more valuable; reducing customer wait time or overall work time of the establishment. Regardless of the organizational choice, business process modeling is an exceptionally useful and unambiguous tool for assisting with managerial decisions.

CONCLUSION

In this paper we discussed application of simulation as a tool for business process optimization. We studied business processes in a pharmacy that has recently gone through an IT initiative.

This relatively non-complex example revealed a number of valuable conclusions:

- Only modelling is not sufficient to get insight into intricate business interactions.
- It is hard to see the impacts of changes unless the models are executed.
- Experimenting with simulations may prevent s from expensive trial-and-error designs.
- As for the users, simulation (animated) models are more easily communicated to users than static models.

REFERENCES

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

Kleijnen, J.P.C. (2005). Supply chain simulation tools and techniques: a survey. *International Journal of Simulation and Process Modelling*, 2005 - Vol. 1, No.1/2 pp. 82 - 89

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

Rivera, A., & Marovich, J. (2001). Use of six sigma to optimize cordis sales administration and order and revenue management process. Proceedings of the 2001 Winter Simulation Conference, pp. 1252-1258.

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

Table 2. Table depicting the results of each simulation

Simulation	Average Wait Time	Total time to Completion
Original Process	0.3697	15 Hours 21 Minutes
What if Scenario A	**0.106**	15 Hours
What if Scenario B	0.274	**14 Hours 20 Minutes**

Related Content

A Work System Front End for Object-Oriented Analysis and Design
Steven Alterand Narasimha Bolloju (2016). *International Journal of Information Technologies and Systems Approach (pp. 1-18).*
www.irma-international.org/article/a-work-system-front-end-for-object-oriented-analysis-and-design/144304

Analyzing the Use of Information Systems in Logistics Industry
Shaligram Pokharel (2009). *Information Systems Research Methods, Epistemology, and Applications (pp. 225-246).*
www.irma-international.org/chapter/analyzing-use-information-systems-logistics/23478

Intelligent Knowledge Systems
T.R. Gopalakrishnan Nair (2015). *Encyclopedia of Information Science and Technology, Third Edition (pp. 4591-4599).*
www.irma-international.org/chapter/intelligent-knowledge-systems/112901

Change Management: The Need for a Systems Approach
Harry Kogetsidis (2013). *International Journal of Information Technologies and Systems Approach (pp. 1-12).*
www.irma-international.org/article/change-management/78903

8-Bit Quantizer for Chaotic Generator With Reduced Hardware Complexity
Zamarrudand Muhammed Izharuddin (2018). *International Journal of Rough Sets and Data Analysis (pp. 55-70).*
www.irma-international.org/article/8-bit-quantizer-for-chaotic-generator-with-reduced-hardware-complexity/206877