Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

SIMULATION BY EXAMPLE FOR COMPLEX SYSTEMS

Amir Kalbasi
Diwakar Krishnamurthy

Electrical and Computer Engineering
University of Calgary
2500 University Drive NW
Calgary, AB T2N 1N4, CANADA

Jerry Rolia
Sharad Singhal

Analytics Lab
HP Labs
1501 Page Mill Road
Palo Alto, CA 94304, USA

ABSTRACT

Our goal is to support capacity management for systems such as hospitals, campuses, and cities, which utilize resources such as people, places, and things in complex ways. Simulation tools have traditionally been used for these sorts of studies, but they require expert model builders to create and maintain abstract business process models of the system under study. This can lead to a lack of representativeness and difficulty in adapting the model for additional or different study scenarios. This paper presents a new simulation approach, Simulation By Example, which overcomes these problems by guiding the simulation using traces, i.e., examples, of the behavior of the actual system but without requiring explicit business process models to be authored. Instead we rely on system instrumentation to capture the traces. We demonstrate the method in two case studies for healthcare systems as described in recent literature.

1 INTRODUCTION

Increasingly, complex processes and systems in environments such as hospitals, campuses and cities are being instrumented to provide continuously available, real-time digital information on their behavior (Chowdhury and Khosla 2007; Jha et al. 2009; Luo 2006; Janz, Pitts, and Otondo 2005; Isken et al. 2005). Typically, the goal of this monitoring is to improve the quality of service delivered while managing costs. Administrators are increasingly looking for capacity management tools to help them understand the behavior of their systems for process improvements that can improve efficiency and cost effectiveness through improved deployment and utilization of resources. We present a novel simulation method, Simulation By Example (SBE), which is well suited to exploit such new information to provide administrators with improved capacity management capabilities.

Most current capacity management systems begin by abstracting interactions in the system under examination into an abstract business process model, such as a flow chart, that describes the actual, best, or planned practice. Unfortunately, such abstract models are not always representative of the real system and, once constructed, they can be inflexible. Complex models may be needed to preserve representativeness. Such complexity may make it difficult for non-experts to understand the model's function, to verify its correctness both at initial deployment and as the system being examined evolves, and to create new what-if scenarios.

With SBE we take a different approach. We rely on instrumented digitized environments to provide example traces of the sequences of steps of process instances. In each step, an entity makes use of a set of resources that provide some service. For the systems we consider, resources may be people, places and things, i.e., anything that is used or is required in the system and that can be observed. We extract the resource possession properties for these resources directly from the traces provided by systems that monitor or coordinate the various resources in the environment being studied. Location sensing systems

based on technologies such as WiFi, Radio Frequency Identification (RFID) and sensor networks (Chong and Kumar 2003; Akyildiz et al. 2002) have been used in different industries such as healthcare to locate, contain, and monitor entity and resource usage (Isken et al. 2005; Amini et al. 2007). For example, a Taiwanese hospital in 2003 was tasked to implement a patient tracking and monitoring system to contain SARS patients (Wang et al. 2006). In another case study, the emergency department of a hospital in the USA utilized more than 20 RFID readers and around 100 RFIP tags to track the flow of patients, staff and equipment (Collins 2004). Along with such location sensing systems, information from other Information Technology (IT) systems, e.g., Electronic Health Records (EHR) (Gunter and Terry 2005) and scheduling systems, can be used to add details that may not be collected by the monitoring systems.

By using traces from sensors and IT systems directly, we avoid the problem of explicitly hand crafting a business process based simulation model. The traces provide the typical sequences for acquiring and releasing resources from which we directly derive a simulation execution. System administrators can choose to exclude certain services or resources to determine how their use or availability impacts the overall system behavior. We argue that changing the mapping between resources and entities or modifying exemplary service steps in a process instance is far easier than building an entire simulation model. As a result, users of SBE are likely to require less simulation expertise and can control the simulation execution in a manner which is more easily grasped than with traditional simulation environments.

The rest of this paper is structured as follows. Section 2 describes related work. The relationship between business processes and the monitoring traces we consider is explained in Section 3. Section 4 describes the SBE approach in detail. Two case studies that demonstrate the validity of the method and its flexibility are offered in Section 5. The paper concludes with a summary and next steps in Section 6.

2 RELATED WORK

Jansen-Vullers and Betjes provide a survey of business process simulation tools (Jansen-Vullers and Netjes 2006). The authors identify the following common approach for all tools surveyed:

"First the business process is mapped onto a process model, possibly supplemented with process documentation facilities. Then the sub-processes and steps are identified. The control flow definition is created by identifying the entities that flow through the system and describing the connectors that link the different parts of the process. Lastly, the resources are identified and assigned to the steps where they are necessary. The process model should be verified to ensure that the model does not contain errors."

An advantage of this approach is that the resulting process model is often a good educational tool. It helps to communicate the process to stakeholders. In addition, the approach can be easily generalized through the specification of system-specific distributions that govern customer arrivals and service. However, there are also several disadvantages with this approach.

The process model guides simulation behavior but it may not be representative. During simulation, flow chart branching choices available within tools typically follow known distributions that in aggregate may not be representative of actual system behavior. Similarly, resource demands are typically characterized for steps or resources using additional distributions. The correlation between successive choices of branches and demands may affect system performance (Isken et al. 2005, Amini et al. 2007) but are not controlled directly in such models. This disadvantage can lead to a lack of representativeness, i.e., real system performance may differ from that reported by a simulation run. Additionally, a simulation of a process model focuses on the abstraction encoded in the process model. Any changes to business processes must be reconciled with the abstraction and its relation to resources and any monitoring data that characterizes demands.

Furthermore, to avoid overly complex flow charts simplifications are often made. There may well be many sequences of resource usage that are valid and even likely but that cannot be realized by a chosen model. For example, Isken et al. (2005) used sensor network data from two outpatient clinics in eastern USA to show that patients had highly unique needs in those systems. The authors logged 3737 patient visits,

where 65% of patients demonstrated a unique sequence of steps. Modeling thousands of unique scenarios is tedious and hence simplifications are made that could potentially affect simulation representativeness.

In contrast to traditional business process model driven simulation techniques, SBE does not require simplification and abstraction of collected data. Traces inferred from the data generated by real-time monitoring systems can be input to the simulator. SBE has some similarities with trace driven simulations where traces of request arrival instants and request service times are used to drive a simulation model directly (Zhou 1988, Arlitt and Williamson 1997). However, these methods typically operate at a level of resource abstraction that targets computing systems and do not consider the same issues as SBE.

Petri-nets and related techniques have often been used to describe process models, and can also be used to describe the example traces we consider. However, creating a single model that considers many example traces would be difficult. Integrating and executing examples considered by us using Petri-nets would require a simulation environment for integrated models that would likely be similar to our new approach.

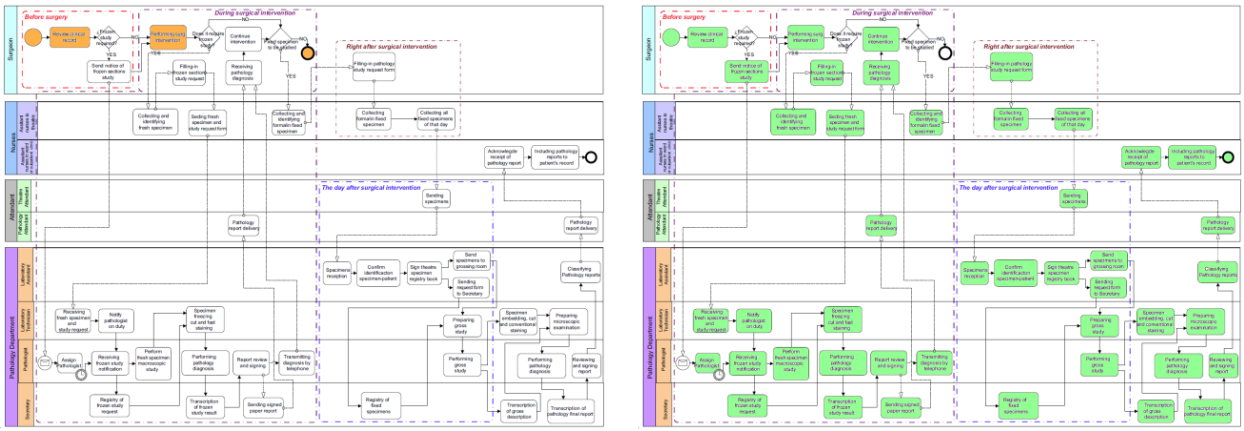
We note that SBE is different from black box techniques such as metamodeling (Cheng 1999). In metamodeling, machine learning models, e.g., regression models, that capture the input-output relationship of a simulated system are developed based on simulation runs. These models are then used as a fast mechanism to support optimization studies (Kleijnen 2009).

As is discussed in the next section, the traces we exploit can be considered as directed acyclic graphs (DAG) that describe partial orders of process steps. In the healthcare domain, several studies have been conducted (Wang et al. 2012; Kayis et al. 2012) where hospital staff manually collected such traces of steps describing instances of a patient surgery process from admission through discharge in support of a surgery scheduling system. As sensors, location based services, and more sophisticated IT systems become pervasive within hospitals and other environments we expect to obtain an abundance of similar information for a greater number of processes. The importance of partial orders for the monitoring and diagnosis of complex distributed event systems is well described by Fabre and Benveniste (2007). Partial order techniques are also used today for management in the telecom industry. Furthermore, techniques exist for finely controlling workload characteristics by manipulating partial order traces (Casale et al. 2012; Krishnamurthy, Rolia, and Majumdar 2006). We envision using such techniques to support what-if studies using SBE.

3 PROCESS MODELS, DAGS AND PARTIAL ORDERS OF STEPS

Business processes in systems such as hospitals can be very complex. Figure 1 (a) and (b) illustrate a process model for a portion of a surgical procedure (Rojo et al. 2008). The figures process model uses control flow mechanisms to express the precedence relationships for steps and swim lanes to illustrate the roles, i.e., types of work that define steps. Despite the apparent complexity, the diagram only illustrates aspects of the process that pertain to whether a tissue sample and pathology report are required as part of a surgery process. The figure omits many other steps that are part of a general surgery process. The figure thus illustrates both the art and complexity of preparing such process diagrams. Figure 1 (a) and (b) also show example sequences of steps, in beige and green, respectively, that correspond to two separate instances of the surgery process, i.e., when tissue samples are not taken and when they are taken, respectively. Each process instance is an example that highlights the steps that are used for an actual patient. Parallel subsequences of steps have fork and join style relationships. Other instances of the surgery process that use different branching choices may also be possible.

Our goal with SBE is the direct simulation of process instances as inferred from traces collected from real systems. These sequences capture: when specific resources are acquired and released; document successive demands placed on resources by steps; and inherently capture successive branching choices.



(a) Process instance example 1 (beige) (b) Process instance example 2 (green)

Figure 1: Surgical process model and process instances (Rojo et al. 2008).

4 SIMULATION BY EXAMPLE

Figure 2 shows key components of our approach. As discussed previously, monitoring systems capture traces of steps for process instances that are used as examples of behavior for the simulator. We draw relevant examples randomly to yield a representative workload for the system. Administrative staff may specify the simulation scenario by choosing a particular arrival schedule for patients and a mix of processes of interest for the system under study. The examples along with information pertaining to the quantity of resources available to process the examples steps are used to drive an SBE simulation.

For simulation, each example can be regarded as a DAG that is treated as a partial order data structure. Whereas the DAG specifies which resource was used for each step at a particular time, the partial order generalizes the usage by specifying a resource demand information for a particular resource role. The Simulation By Example engine uses the partial orders to simulate system behavior, matching each request for resources that satisfy particular roles with simulated resources that satisfy those roles. The partial orders are used to generate an initial discrete event list that is utilized and updated throughout the simulation. The administrator controls what-if experiments by varying the mix and order of examples, resource quantities, and the mapping of roles to resources. We next consider several challenges that arise when working with traces. We then present our simulation approach that overcomes the challenges.

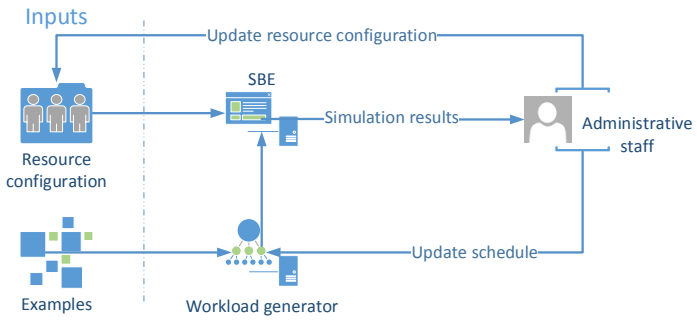


Figure 2: Simulation By Example.

We have found several interesting behaviors that must be considered when recording and simulating resource usage using SBE. These include: a method is needed to relate steps to resource acquisition and release; multiple resources acquired for a step may not all have the same service demands; not all resources

acquired for a step are released when the step ends; in a semantically correct example, an acquired resource may be released in any future step as long as it is only released once; some steps are asynchronous from the flow of the overall process instance; and, some steps, and/or later steps, may have an affinity to a particular resource. The purpose of the simulation is to match the needs of steps with available resources while observing partial order constraints.

Our SBE implementation is built of four basic abstractions: sessions, steps, resource groups and individual resources. A session is the simulators representation of a process instance. For example, the surgical process instance shown in Figure 1 (b) in the previous section may represent a complete session. A session is composed of individual steps. A new step is introduced at any point in a trace when a new resource is needed or is no longer needed to continue its work on behalf of the session. If multiple resources are needed or no longer needed at the same time, they are acquired or released as part of the same step. Steps may be dependent on the completion of previous steps (serial) or they may be allowed to continue independently of the completion of previous steps (parallel). Different types of step dependencies are described in Section 4.1. The SBE system determines which resources will be allocated to a given step by determining which resource groups are capable of supporting the step, and which individual resources are available within these groups. Capacity management studies can be performed by varying the numbers of resources within various groups and by managing the binding of steps to resources based on the steps requirements.

4.1 Simulation Execution

Figure 3 provides details of how the SBE abstractions are represented in the runtime environment. Sessions have a type, an arrival time, and one or more Steps. Steps have a name and one or more RoleInStep representing the Roles that are required to begin execution of the step. Each RoleInStep has a particular roleName that identifies the type of work. RoleInStep has several other attributes that describe blocking relationships. The aSync attribute indicates whether a RoleInStep must complete its resource demand before the session can advance to its next step. If all of a steps RoleInSteps have aSync as true then the session can advance to its next step even if all the resources needed to perform the step are not immediately available. This enables asynchronous processing. Steps have a followedBy relationship that allows multiple Steps to follow a Step to capture the partial order. A Step may only execute if the synchronous demands of its preceding Step(s) are complete. The holdOverSteps attribute indicates that a resource used for the RoleInStep is not released until later. Another RoleInStep must appear later in the session with a holdOverSteps value that indicates that a corresponding held resource should be released when it completes its resource demand. The combination of this feature with aSync enables fork and join behavior.

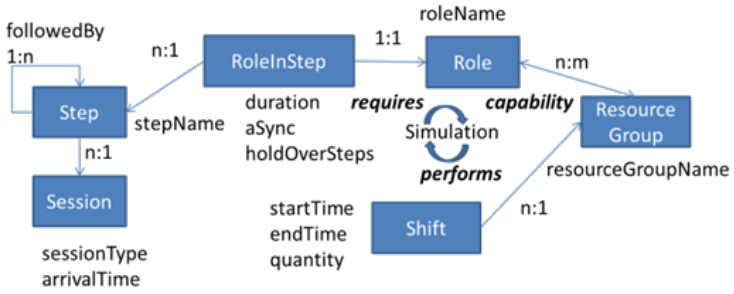


Figure 3: Simulation By Example data structures and simulation process.

If several Steps are enabled for execution at the same time a decision policy determines which has priority e.g., first come first served or execute the highest priority session. This is necessary because some resources may be needed by more than one of the steps but can only serve one step at a time. Note that roleNames bind the session step requirements for resources with available resources.

Finally, Resource groups identify one or more resources that support the same Roles. ResourceGroups have Shifts, which describe the time varying availability of resources in ResourceGroup. Specifically, they specify the start and end times for each shift, the Roles with roleName that can be performed during the shift, and the quantity of resource available for the shift.

4.2 Configuration Driven Input Processor

Figure 4 illustrates how a configuration driven input processor allows an administrator to manage the binding of Role with resources in ResourceGroup. We refer to such relationships as role affinities. The shift input file for SBE identifies the hierarchical organization of resource groups in a system, their role affinities, and the time varying availability of resource quantities. The session input file identifies the affinity requirements of each individual session, e.g., whether a patient that requires a resource supporting a particular role more than once requires the identical resource instance repeatedly. The case study of Section 5.2 shows how these are used and how relaxing affinity requirements can be used to improve overall performance.

Figure 4: Configuration file input.

4.3 Controlling Simulation Model Abstraction

End users can control simulation model abstraction by controlling whether a set of Session types, Roles, Resources, or ResourceGroups are used in the simulation or by controlling the affinity of work to resources.

Excluding certain entities changes the scope of the simulation model for an end-user of the simulation. Session types can be included or excluded from the session input file. Roles and ResourceGroups can be excluded from a shift input file. Resource demands for excluded entities can be efficiently preserved in the simulation as temporal delays.

Simple pattern matching and graph grammar based tools (Rozenberg and Ehrig 1997) can support the organization and manipulation of sessions. Resource demands and even process flows can be modified such that the resulting modified sessions can be inspected and verified. Thus, end users can control scope, affinity, and even process flows without requiring a simulation expert to create and or modify corresponding process models and map them onto traditional simulator constructs.

5 CASE STUDIES

5.1 Comparing SBE with a Traditional Simulator

The first case study considers a real RFID instrumented trauma center considered by Amini et al. (2007). The purpose of this case study is to validate and compare the simulation results of SBE with simulation results of Arena (Kelton, Sadowski, and Swets 2009), which is a general purpose simulation package. Furthermore, it demonstrates practical shortcomings of traditional simulation techniques.

In this trauma center, patients are treated in rounds of treatments. RFID tags were worn by patients to trace their advancement through medical care. 23 RFID readers were used to determine patients locations. Patient treatment process starts at Critical Care Assessment (CCA), where patients are evaluated. After evaluation, patients are either released from the center or they are moved to the X-ray and/or Computer Tomography/Magnetic Resonance Imaging (CT/MRI). After imaging, patients are moved back to the CCA and evaluated again. This process can be repeated up to 4 times. Each repetition is termed as a round.

A simulation model of the trauma center is shown in Figure 5. Due to the complexity introduced by rounds of treatment each with their own unique properties, the process model is very hard to reproduce clearly in this paper. We show a scaled down version in Figure 5 to illustrate the models complexity. The model is very similar to that used by Amini et al. (2007). The models complexity arises due to patients experiencing different service time distributions for the same resource during treatment. Specifically, the service time distribution of each round depends on how many rounds a patient has to go through as well as the current round of treatment. In total, Amini et al. identified more than 30 service time distributions. To account for the different service time distributions in the simulation model, they had to flatten the rounds rather than using a looping mechanism. Also, they had to include a complex decision making logic to ensure patients receive a correct service time at each step. As shown in the figure, to realistically model this system and take into account the round-specific service times distributions, the model had to be significantly complicated. Such a system can benefit from SBE since one can avoid complexities associated with developing the simulation model.

Due to the unavailability of the original traces from the system, synthetic workloads that represent real traces were generated using the published distributions (Amini et al. 2007). Similar logic that is used in the Arena model was used to generate different types of patients. SBE was executed with this trace while Arena was driven by the model shown in Figure 5. Both simulators were run for 1000 patients. The simulation runs were replicated 10 times.

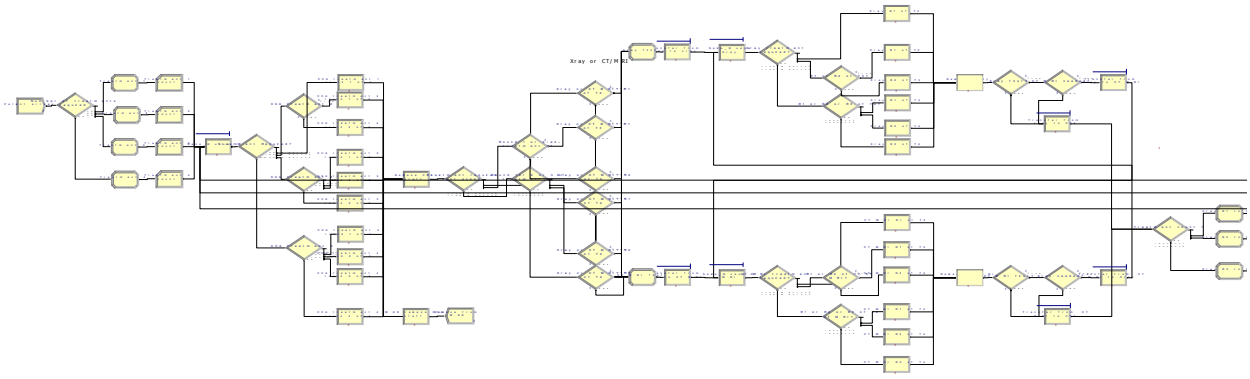


Figure 5: Simulation model of the trauma center.

Table 1 shows the mean and variance of the utilization of the resources as well as the patient busy time, wait time, and response time. Two-tailed t-tests at a significance level $\alpha = 0.05$ were performed on the results to verify whether the results from both simulators match. Our null hypothesis is that the SBE and Arena values for each of the metrics in Table 1 are statistically the same. Since all the p-values are greater than $\alpha = 0.05$, the t-tests show that there is no significant difference between key performance measures reported by both simulation tools.

Table 1: Mean, variance, and t-test of simulation results.

		CCA Staff Utilization	CT/MRI Staff Utilization	Nurse Utilization	X-ray Staff Utilization	Patient Busy Time	Patient Wait Time	Patient Response Time
Arena	Mean	64.64%	89.83%	53.42%	79.42%	69.19	8.18	77.37 min.
	Variance	0.13	6.27	0.31	1.40	0.43	3.38	6.04
SBE	Mean	64.43%	90.03%	53.70%	79.87%	69.20	8.61	77.81 min.
	Variance	0.10	0.58	0.97	0.52	0.15	1.13	1.65
t-test (p-value)		**0.20**	**0.82**	**0.45**	**0.33**	**0.97**	**0.53**	**0.62**

The study shows that SBE is able to closely match the results from a traditional simulator. Furthermore, it also shows that one can avoid the business process modeling step if such information is encoded implicitly within traces.

5.2 Ambulatory Care Unit Case Study

Another case study was performed to demonstrate the flexibility of the SBE method for what-if analyses. The simulator is applied to study the care process for an Ambulatory Care Unit (ACU) for cancer treatment (Santibáñez et al. 2009).

Our case study considers the impact of patient-oncologist affinity on system performance. Affinity is an important characteristic of health systems. With patient-oncologist affinity, when a patient sees a doctor they may want to see a specific doctor. If that doctor is busy, then they have to wait. Similarly, a doctor may be available but only meets with his or her own patients. With advances in EMR systems it may be possible to receive comparable service from any available doctor with the same specialty. This scenario explores the impact of affinity on the total time needed to process patients and average patient waiting time.

The clinic normally operates from 8:00 am to 4:30 pm; however, if at 4:30 pm there are patients remaining to be serviced, the working hours are extended until all patients are serviced. Patients have scheduled appointments with oncologists that are booked in advance. However, they may arrive sooner or later than their scheduled appointment times. If a patient arrives earlier, they are processed as soon as appropriate resources become available.

Figure 6 shows the steps of an example ACU session for a patient. The example is in a format that is input to the simulator. Note that the input has a resourceDepartmentName that is useful for organizing resources for this study. While this is not a simulator concept, the configuration driven input processor creates role names that are a concatenation of the input roleName, resourceGroupName, and resourceDepartmentName. The corresponding shift file also specifies role names that are a combination of such strings. As we shall see, the use of names in this manner helps to manage affinity in a straightforward manner.

step	stepName	aSync	holdOverSteps	duration	roleName	resourceGroupName	resourceDepartmentName
0	Check-in	N	N	00:04:00	Receptionist	Receptionist_Check-in	Check-in
1	Waiting	N	N	00:00:00	WaitingRoom	WaitingRoom_Waiting_Area	Waiting_Area
2	Preparation	N	N	00:05:00	Nurse	Nurse_Preparation_Medical	Preparation_Medical
2	Preparation	N	Y	00:05:00	ExaminationRoom	ExaminationRoom_Preparation_Medical	Preparation_Medical
3	Examination	N	Y	00:14:00	Oncologist	Oncologist_Examination_Medical_3	Examination_Medical
4	Room_Cleanup	Y	N	00:03:00	Nurse	Nurse_Room_Cleanup_Medical	Room_Cleanup_Medical
4	Room_Cleanup	Y	R	00:03:00	ExaminationRoom	ExaminationRoom_Preparation_Medical	Preparation_Medical
5	Report_Preparation	N	R	00:07:00	Oncologist	Oncologist_Examination_Medical_3	Examination_Medical
6	Discharge_Preparation	N	N	00:07:00	Clerk	Clerk_Discharge_Medical	Discharge_Medical
7	Discharge	N	N	00:04:00	Nurse	Nurse_Discharge_Medical	Discharge_Medical

Figure 6: Example ACU input session.

The session in Figure 6 has 8 steps. In step 0, a patient arrives at the clinic and requires a receptionist to check in. For this example the check in step takes 4 minutes. It requires a resource that supports the role Receptionist from the resource group Receptionist_Check-in in department Check-in. In the next step, step 1, the patient waits in the waiting room for a minimum of 0 minutes until called by a nurse. For step 2 to begin, both a nurse and an examination room need to be available. When a nurse and an examination room are available, the nurse takes the patient to the examination room and prepares the patient for an oncologists visit, e.g., collects the patients current weight. In this example, this takes 5 minutes. Since the examination room is needed for multiple steps of this session, the holdOverSteps attribute for the corresponding RoleInStep of the examination room is enabled in this step. The nurse then leaves the patient in the examination room to wait for the oncologist. In the next step, step 3, the oncologist visits the patient for 14 minutes. The oncologist is also heldOverSteps because he goes on to prepare a medical report in step 5 before being released and made available to help another patient. After step 3 completes, the session advances to step 4. Step 4 causes work that is asynchronous to the process. It requires a nurse who cleans

the examination room. Cleaning the room takes 3 minutes for this example. After the room is cleaned the examination room becomes available for use by another patient and the nurse becomes available for another task. Because both roleInStep are asynchronous, the session advances from step 3 to step 5 in zero time. In step 5, the oncologist takes 7 minutes to prepare the report in the electronic records system. Steps 6 and 7 require a clerk to prepare a discharge report for the patient and a nurse to present the report to the patient and arrange an appointment for a follow up visit if necessary, respectively.

Alternative sessions could also be considered. For example, it may be that an oncologist is also needed at the time of discharge, or that the same oncologist that prepared the report must be held until after discharge. Mixes of such sessions can easily be included in a simulation. This is in contrast to more traditional approaches that would try to confound such related examples onto a more abstract business process model.

5.3 Generating Sessions and Controlling Affinity

The generation of ACU example sessions is straightforward for our case study. Generating many instances of the single session type simply required different session arrival times and different demand values for each of the steps. Table 2 shows the mean demands for resources for the steps. We were motivated by distributions from Santibáñez et al. (2009) and Wang et al. (2012) when generating demands.

Table 2: Mean service times for each RoleInStep.

Resource	**Step**	**Mean service time (min.)**
Receptionist	0	5
Nurse Prep	2	6
Oncologist Examination	3	16 (bimodal: 8, 24)
Nurse Clean	4	4
Oncologist Report	5	8
Clerk	6	8
Nurse Discharge	7	5

Step 3 of Figure 6 shows that the patient requires a specific oncologist as indicated by the resourceGroupName Oncologist_Examination_Medical_3. This refers to oncologist resource group 3 and there is only one oncologist per oncologist group as specified in the shifts file. Here the roleName, resourceGroupName, and resourceDepartmentName are fully specified. To relax the affinity constraint for patient-oncologist a percentage of the sessions are generated with an empty string for the oncologist resourceGroupName. For example, in step 3 of Figure 7, the resourceGroupName is left blank. This suggests that any oncologist in the Examination_Medical ResourceDepartment that has role OncologistExamination_Medical – as concatenated by the input processor, as an additional role in the shifts file can consult with this patient.

step	stepName	aSync	holdOverSteps	duration	roleName	resourceGroupName	resourceDepartmentName
3	Examination	N	Y	00:14:00	Oncologist		Examination_Medical

Figure 7: Example of step with affinity to any oncologist.

To determine the impact of affinity on system performance synthetic workloads were generated for 11 working days. Each working day consisted of 210 sessions and had patient-oncologist affinity ranging from 0%, i.e., a patient can see any available oncologist, up to 100%, i.e., each patient sees a specific oncologist, in 10% increments. The workload was designed to fully utilize oncologists since they are the most expensive resource. Patients are scheduled to arrive on average every 24 minutes per oncologist. The arrival times were Normally distributed with the average patient arriving 13 minutes before their scheduled appointment time. The standard deviation for inter-arrival times is 15 minutes so some patients do arrive late. Examination times follow a bimodal distribution with half the appointments taking 8 minutes on

average and the remaining half taking 24 minutes on average. In each case random values are chosen from the Normal distribution with the standard deviation equal to the mean. The bimodal distribution reflects the fact that some appointments are significantly shorter than others. The effects of variable service times, competition for resources, and randomized arrival times affect the resulting oncologist utilization. In general, the oncologists are approximately 90-93% utilized instead of the planned 100% busy and the total patient processing time goes beyond the desired 4:30 pm.

For the patient-oncologist affinity experiment, there are 10 oncologists each in his or her own group. All other resources are over provisioned so that they do not affect processing time or patient waiting time. Figure 8 shows the impact of patient-oncologist affinity on the total time it takes for the hospital to process 210 patients and upon the patient wait times per session. The figure shows the mean results of 30 replications of each experiment. If every patient must see one specific oncologist then it takes nearly 60 minutes longer to process patients than if there is no affinity. This is a significant cost for a hospital as staff and facilities must be fully operational. Furthermore, patients have average waiting times that are 14 minutes longer with 100 percent affinity than with zero affinity. If half the patients could see the first available oncologist then the total time needed to process patients could be reduced by nearly 40 minutes and patients would have an average waiting time about 6 minutes shorter than with 100% affinity. 95% confidence intervals for total processing times were within ± 5 and ± 10 minutes, with the larger values for the larger processing times. Similarly, mean patient waiting times had 95% confidence intervals within ± 1 and ± 3 minutes.

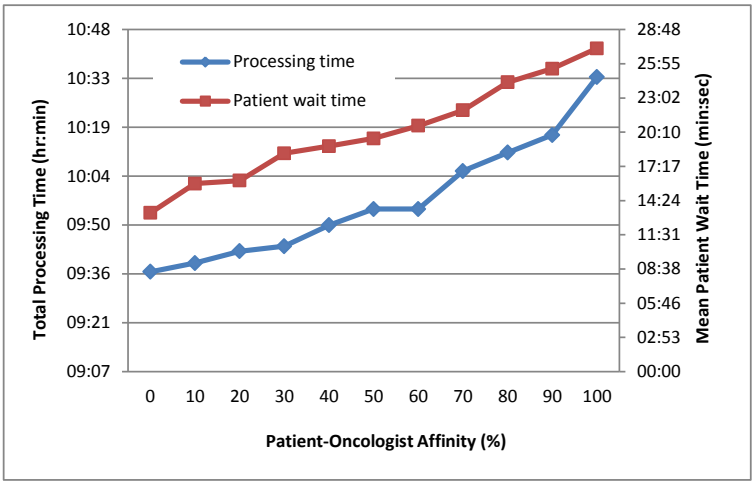


Figure 8: Patient-oncologist affinity vs. total patient processing time and patient wait time.

6 SUMMARY AND CONCLUSIONS

This paper introduced a new simulation by example method for supporting capacity planning exercises in systems such as hospitals, campuses, and grids. The method enables modeling exercises without a traditional business process model building step. It enables the study of simulation scenarios where changes to process instances or sessions are important but may be hard to capture in process models. This is the main contribution of the approach and is important for the complex systems we aim to support. We have also shown through a case study that the method is flexible for use by non-simulation experts for common what-if scenarios such as relaxing affinity requirements.

The method takes as input traces that are examples of process executions for the system under study. Advances in location aware sensors, workflow management systems, and scheduling systems promise to yield an abundance of such information as the systems become smarter and more digital. The traces from

such system correspond to an implicit monitoring model that we exploit. This frees us from the need for a business process model. Since the method is trace driven, it is more likely to offer predictions that are representative of actual systems. For situations where traces do not exist, as in our case study, traditional business process modeling frameworks can be used to generate synthetic traces. In this way SBE is also complementary to existing frameworks.

We note that traces may contain some outliers. We can use techniques such as clustering (Han, Kamber, and Pei 2006) to identify outlier examples and allow administrators to decide whether such behavior is valid or should be excluded from the simulation. Our future work includes testing SBE using real traces from healthcare environments. In addition, we are planning to incorporate techniques that can automate the currently manual schedule and resource configuration optimization process shown in Figure 2.

REFERENCES

Akyildiz, I. F., W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. "A Survey on Sensor Networks". *IEEE Communications Magazine* 40 (8): 102–114.

Amini, M., R. F. Otondo, B. D. Janz, and M. G. Pitts. 2007. "Simulation Modeling and Analysis: A Collateral Application and Exposition of RFID Technology". *Production and Operations Management* 16 (5): 586–598.

Arlitt, M. F., and C. L. Williamson. 1997. "Trace-Driven Simulation of Document Caching Strategies for Internet Web Servers". *SIMULATION* 68 (1): 23–33.

Casale, G., A. Kalbasi, D. Krishnamurthy, and J. Rolia. 2012. "BURN: Enabling Workload Burstiness in Customized Service Benchmarks". *IEEE Transactions on Software Engineering* 38 (4): 778–793.

Cheng, R. C. H. 1999. "Regression Metamodeling in Simulation Using Bayesian Methods". In *Proceedings of the 1999 Winter Simulation Conference*, edited by P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 330–335. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.: ACM.

Chong, C.-Y., and S. P. Kumar. 2003. "Sensor networks: evolution, opportunities, and challenges". *Proceedings of the IEEE* 91 (8): 1247–1256.

Chowdhury, B., and R. Khosla. 2007. "RFID-based hospital real-time patient management system". In *Proceedings of the 6th IEEE International Conference on Computer and Information Science*, 363–368. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Collins, J. 2004. "Hospital Gets Ultra-Wideband RFID". *RFID Journal*.

Fabre, E., and A. Benveniste. 2007. "Partial Order Techniques for Distributed Discrete Event Systems: why you can't avoid using them". Technical Report 5916, INRIA.

Gunter, T. D., and N. P. Terry. 2005. "The Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions". *Journal of Medical Internet Research* 7 (1): Article e3.

Han, J., M. Kamber, and J. Pei. 2006. *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Isken, M. W., V. Sugumaran, T. J. Ward, D. Minds, and W. Ferris. 2005. "Collection and Preparation of Sensor Network Data to Support Modeling and Analysis of Outpatient Clinics". *Health Care Management Science* 8 (2): 87–99.

Jansen-Vullers, M., and M. Netjes. 2006. "Business Process Simulation-A Tool Survey". In *Proceedings of the Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark*.

Janz, B. D., M. G. Pitts, and R. F. Otondo. 2005. "Information Systems and Health Care-II: Back to the Future with RFID: Lessons Learned - Some Old, Some New". *Communications of the Association for Information Systems* 15 (1): Article 7.

Jha, A. K., C. M. DesRoches, E. G. Campbell, K. Donelan, S. R. Rao, T. G. Ferris, A. Shields, S. Rosenbaum, and D. Blumenthal. 2009. "Use of Electronic Health Records in US Hospitals". *New England Journal of Medicine* 360 (16): 1628–1638.

Kayis, E., H. Wang, M. Patel, T. Gonzalez, S. Jain, C. Longhurst, R. Ramamurthi, C. Santos, S. Singhal, J. Suermondt, and K. Sylvester. 2012. "Improved Prediction of Surgery Duration Using Operational and Temporal Factors". In *Proceedings of the 2012 Annual Symposium of the American Medical Informatics Association*, 456–462. Chicago, IL, USA.

Kelton, W. D., R. Sadowski, and N. Swets. 2009. *Simulation with Arena*. 5th ed. McGraw-Hill.

Kleijnen, J. P. C. 2009. "Kriging Metamodeling in Simulation: A Review". *European Journal of Operational Research* 192 (3): 707–716.

Krishnamurthy, D., J. A. Rolia, and S. Majumdar. 2006. "A Synthetic Workload Generation Technique for Stress Testing Session-Based Systems". *IEEE Transactions on Software Engineering* 32 (11): 868–882.

Luo, J. 2006. "Electronic Medical Records". *Primary Psychiatry* 13 (2): 20–23.

Rojo, M. G., E. Rolón, L. Calahorra, F. O. García, R. P. Sánchez, F. Ruiz, N. Ballester, M. Armenteros, T. Rodríguez, and R. M. Espartero. 2008. "Implementation of the Business Process Modelling Notation (BPMN) in the Modelling of Anatomic Pathology Processes". *Diagnostic Pathology* 3 (Suppl 1): Article S22.

Rozenberg, G., and H. Ehrig. 1997. *Handbook of Graph Grammars and Computing by Graph Transformation*, Volume 1. World Scientific Singapore.

Santibáñez, P., V. S. Chow, J. French, M. L. Puterman, and S. Tyldesley. 2009. "Reducing Patient Wait Times and Improving Resource Utilization at British Columbia Cancer Agency's Ambulatory Care Unit Through Simulation". *Health Care Management Science* 12 (4): 392–407.

Wang, H., E. Kayis, M. Patel, C. Santos, T. Gonzalez, S. Jain, S. Singhal, R. Ramamurthi, J. Suermondt, and K. Sylvester. 2012. "An Integrated Next-Day Operating Room Scheduling System". In *Proceedings of the 2012 Annual Symposium of the American Medical Informatics Association*. Chicago, IL, USA.

Wang, S.-W., W.-H. Chen, C.-S. Ong, L. Liu, and Y.-W. Chuang. 2006. "RFID Applications in Hospitals: A Case Study on a Demonstration RFID Project in a Taiwan Hospital". In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, Volume 8, Article 184.

Zhou, S. 1988. "A Trace-Driven Simulation Study of Dynamic Load Balancing". *IEEE Transactions on Software Engineering* 14 (9): 1327–1341.

AUTHOR BIOGRAPHIES

AMIR KALBASI is a PhD candidate and research assistant at the University of Calgary, Canada. He received his MSc degree from the University of Calgary. His research interests are system performance modeling and simulation, system management and optimization, and software performance.

JERRY ROLIA is a principal scientist in the Analytics lab of HP Labs. Prior to joining HP Labs, Jerry was an Associate Professor at Carleton University in Ottawa, Canada. His interests include analytic systems for big data, software and system performance, in-memory computing, Internet of Things, and systems simulation.

DIWAKAR KRISHNAMURTHY is an Associate Professor at the University of Calgary. His research interests are focused on the performance evaluation of software systems. He is currently involved in research projects related to cloud computing, virtualization technologies, and healthcare simulation.

SHARAD SINGHAL is Director of the Analytics Workload Group at HP Labs. Prior to joining HP, he worked at Bell Labs and Bellcore (now part of Ericsson). At HP Labs, he has led teams that have developed techniques for monitoring and managing service level agreements; methods for controlling service quality in multi-tier applications, resource allocation and assignment algorithms; as well as architectures for management of large-scale data centers and cloud computing.