Extending Palladio by Business Process Simulation Concepts

Robert Heinrich, Barbara Paech
Institute of Computer Science
University of Heidelberg, Germany
{heinrich, paech}@informatik.uni-heidelberg.de

Jörg Henss
Institute for Programme Structures and Data Organisation
Karlsruhe Institute of Technology, Germany
joerg.henss@kit.edu

***Abstract*—Business process design and enterprise information system (IT system) design are often not well aligned, which leads to problems at runtime caused by neglecting the mutual impact of business processes and IT systems. Simulation is a promising approach to support the alignment of business process design and IT system design by impact prediction. Currently, the Palladio approach does not include business process modeling and simulation functionality, which impairs the accuracy of simulation results. We propose an extension to the Palladio Component Model (PCM) and the simulation behavior in order to enable the modeling of business processes and the simulation of the mutual impact of business processes and IT systems using the existing Palladio tool chain.**

***Index Terms*—Business Process; Enterprise Information System; Simulation; Alignment; Performance.**

I. INTRODUCTION

Business processes and IT systems mutually impact their performance in non-trivial ways. Nonetheless, business process design and IT system design are often not well aligned. Missing alignment of business process design and IT system design results in problems at runtime such as large IT response times, large process execution times, overloaded IT systems or interrupted processes. Simulation is a powerful approach to predict the impact of a business process design on the performance of supporting IT systems and vice versa. Based on the predicted impact, business process design and IT system design can be adapted to enable alignment.

There are several business process simulation approaches and several IT system simulation approaches. In the business process domain, simulation is commonly used to predict business process performance and financial impact (e.g. [1]). In the IT domain, computer network simulation is widely spread for decades to estimate performance of network topologies. Moreover, there are approaches for software architecture simulation on component level (e.g. [2]) and service-oriented architecture simulation (e.g. [3]). Focused on one domain, these approaches are adequate to predict the performance of a business process or an IT system isolated of each other. However, in current simulation approaches, there is little integration between the business process domain and the IT domain.

Considering business processes along with IT systems in simulation can support several roles in the joint development of business processes and IT systems.

- Requirement engineers can check in design phase whether an IT requirement can be met by a proposed IT system design for a given business process design.
- System developers can compare design alternatives of IT systems invoked in a given process to each other without implementing prototypes.
- Hardware administrators can check the utilization of IT resources such as CPU or hard disk drive for a proposed IT system design and a given business process design.
- Business analysts can check in design phase whether a process requirement can be met by a proposed business process design and a given IT system design.
- Process designers can compare business process design alternatives to each other without executing a business process in practice. Thereby, the impact of the given IT system(s) on the process is considered.

The Palladio approach [2] currently does not provide business process modeling and simulation functionality. This impairs the accuracy of IT simulation results since mutual impact, e.g. on workload distribution, is not correctly considered. Betz et al. [4] sketch a framework to integrate the life-cycles of business processes and IT systems based on Palladio. This framework uses IT simulation and business process simulation isolated of each other. However, isolated simulations are not able to consider the mutual impact of business processes and IT systems addressed in this paper.

If we abstract from the different semantics of business process simulation and IT simulation, there are several analogies. Both kinds of simulations...

- ...can be built upon queuing networks (queuing theory [5])
- ...simulate the utilization of resources (human actor resources or IT resources). An actor resource is the representation of a human actor in the process. Actor resources as well as IT resources process jobs from their waiting queue in a certain processing rate. See [1] for an example of actor resources and their waiting queues in business process simulation.
- ...use a specification of a workflow of actions to be processed by the resources. In business process simulation besides system steps also actor steps are considered.
- ...use actions that can be composed hierarchically.

- ...use a specification of workload. In process simulation workload is often specified in the form of inter-arrival times which is comparable to the open workload in Palladio.
- ...acquire and release shared passive resources. In analogy to passive resources in Palladio, passive resources in a business process are devices or machines which are available in limited capacity and are required to perform the process but do not actively process it (cf. [1]).

Considering these analogies, Palladio seems to be an adequate foundation to be extended by business process simulation concepts to enable an integrated process and IT simulation for performance prediction purposes.

In this paper, we propose extensions of the PCM and the simulation behavior in order to enable the simulation of the mutual impact of business processes and IT systems. The paper is structured as follows: In Section II, we introduce definitions required for understanding the following sections. We present an example process and discuss the mutual impact of business processes and IT systems in Section III. The need for an integrated business process and IT simulation is presented in Section IV by pointing out open issues in Palladio. Requirements on an integrated simulation are listed in Section V. In Section VI, we show how to extend the PCM and the simulation behavior by business process concepts to enable an integrated simulation. An example of the behavior of the proposed extensions to the simulation is shown in Section VII. Section VIII concludes the paper and presents future work.

II. Definitions

A *business process* is a “set of one or more linked activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships“ [6]. Each activity within the process consists of a set of one or more linked steps. Steps are either completely performed by a human actor – called *actor step*s – or completely performed by an IT system – called *system step*s. In the PCM, a system step is represented by an EntryLevelSystemCall [2]. Yet, to keep things generally intelligible, we use the term system step in this paper.

A human actor performs actor steps lined up in his/her *worklist*. A worklist is comparable to a waiting queue of an IT resource (cf. [2]) which lines up jobs to be processed by the IT resource. Business processes are typically specified on several levels of abstraction which are composed hierarchically. Processes consist of subprocesses, subprocesses consist of activities, and activities consist of actor steps and system steps. Hierarchical composition is required to keep track of large processes. In this paper, we focus on the level of steps as detailed actor steps and system steps are required to determine the mutual impact of business processes and IT systems in simulation. The mutual impact is discussed in detail in Section III. A *business process instance* is the “representation of a single enactment of a process“ [6]. An *IT system instance* is an executable representation of the IT system design. A business process design P and an IT system design S are *aligned*, if

- System steps of S are invoked in P, and

for all the process instances P' of P and all the system instances S' of S holds:

- S' meets the requirements of S when used in P', and
- P' meets the requirements of P when uses S'.

Workload is “the amount of work to be done“ [7]. Business process workload determines the amount of process instances that traverse the business process. Often workload is measured in process instances per time unit. Process instances traverse all the actor steps and system steps on a certain path of the process from the process start point to a process end point. *Response time* is the total time required by a process instance to traverse a system step. *Execution time* is the total time required by a process instance to traverse an actor step. The time required to traverse an activity within the process or an entire process is called execution time, too. *Distance* refers to the difference in time in which the process instances reach a certain point in the process. The distance between two process instances begin the execution of the process is called *inter-arrival time*. *Workload distribution* refers to the distance between process instances within the process. For example, three process instances can occur in a constant distance (30 seconds) to each other, or they can occur in *bursts*, e.g. all three process instances occur directly after the other or even at the same time. In all cases, the workload is three process instances in one minute. While traversing the process, the distance between process instances can vary. Bursts of process instances or gaps between the process instances can be created.

III. Mutual Impact of Business Processes and IT Systems

Business processes and involved IT systems mutually impact each other in several ways. In this paper, we focus on performance impact. In the following, we introduce an example and discuss the mutual impact of business processes and IT systems based on the example.

A. The Process of Order Picking

Suppose the process of order picking in the store of a manufacturer. Goods requested in an order are taken from the store and are packed to be transported by trucks. The process is illustrated in Figure 1. The example process is a simplified representation of a process from practice we are currently analyzing in a case study.

The process has a start event and an end event, represented by circles. Steps are represented by rectangles with rounded corners. “AS:” is used to mark actor steps. “IT:” is used to mark system steps. Arrows represent the control flow in the process. Lanes represent roles of human actors. For each role, several human actors are available, i.e. orders can be processed concurrently.

The shift leader first gathers the data from the order for picking. The IT system inserts the data into a database and

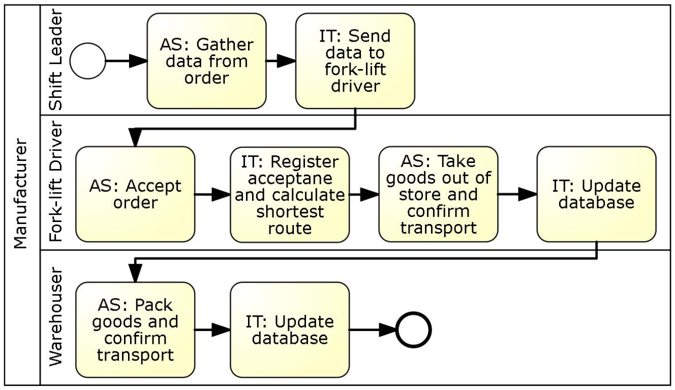

Fig. 1. Example Process

sends the order data from the database to a mobile client of the fork–lift driver. The fork–lift driver accepts the order, which is registered in the database by the IT system. The IT system then calculates the shortest route to the goods in the store of the manufacturer. After the route has been calculated, the fork–lift driver takes the goods out of the store and put them on a location where they are packed for transport. Then, s/he confirms the transport. The IT system updates the database and informs the warehouser. Then, the warehouser packs the goods for transport, put them on a location where they are picked up by a truck later, and confirms the transport. Finally, the IT system updates the database.

There are strict time constraints in the process, since requested orders have to be available for transport at the time the trucks arrive. Since delays are very expensive in the logistics business, it is a time-critical process.

B. Process Impact on IT System Performance

IT system performance is determined by the business process design as well as the business process workload. The business process design determines which system steps are invoked in the process, when a specific system step is invoked and which system steps are invoked concurrently. For example, suppose a business process that includes two system steps. IT system performance may differ depending on whether the steps are invoked sequentially or concurrently. As defined in Section II, business process workload determines the amount of process instances that traverse the process. Process instances traverse all the actor steps and system steps on a certain path of the process from the process start point to a process end point. Thus, business process workload determines how often an IT system is invoked. IT system performance may differ depending on whether the system is invoked once per second or 100 times per second.

C. IT System Impact on Process Performance

IT system performance impacts the business process performance in two ways. Firstly, if the IT system is overloaded because too many actors invoke the system, it is no longer available for actors in the business process. Thus, the execution of the business process is impeded or even interrupted. For example, if the IT system cannot send the order data to the mobile client (e.g. as it is overloaded by too many actors), the goods may not be available for the truck in time. Secondly, the response time of system steps may impact the business process performance if its extent is comparable to the extent of execution time of actor steps within the process. Thus, IT system response time may significantly increase the execution time of the entire business process or single activities within the process. Response time and execution time is defined in Section II. For example, in our case study, the transmission of order data to the mobile client of the fork–lift driver can last up to 40 minutes and more which heavily impacts the process execution time as it increases accordingly.

D. Mutual Impact of Actor Steps and System Steps on Workload Distribution

IT system performance and business process performance are influenced by workload distribution within the process. See Section II for a definition of workload distribution. According to Mi et al. [8], workload distribution has "paramount importance for queueing prediction, both in terms of response time mean and tail" as it impacts performance significantly. Unequal distributions of workload ("burstiness" factor) often lead to increasing response times. As human actors process jobs on their worklist in a similar manner as IT resources, e.g. following the FIFO (First In First Out) principle, it is logically comprehensible that workload distribution impacts process performance the same way as it impacts IT performance.

Workload distribution in the process is influenced by actor steps as well as system steps assuming synchronous scheme of communication. In this paper, we discuss synchronous scheme of communication. Assuming asynchronous scheme of communication (i.e. actors do not wait for system steps to finish), workload distribution is only influenced by actor steps. If an actor is already busy when an actor step should be performed by this actor, the execution of the actor step has to wait until the actor is ready to perform the actor step. If an IT resource used in a system step is already busy when it is invoked by an actor request, the request has to wait until the resource is ready to process the request. Process instances may also have to wait for shared passive resources to be released. As mentioned above, there are shared passive resources in business processes as well as in IT systems. Waiting times hinder the flow of the process instances through the business process. For each step, waiting times may differ from process instance to process instance. Thus, the distances between the process instances in the process may vary during process execution. The example in Table I shows how the distance between process instances is decreased which may result in a burst.

Suppose, there are two actors A1 and A2 of the role fork–lift driver. The queue of A1 has a length of five time units at t_0. The queue of A2 has a length of six time units at t_0. This represents the processing time of the remaining work to be done by the actors. There are two process instances I1 and I2 which reach the actor step "AS: Accept order" at a distance

Time	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
Actor A1	5 (+3 from I1)	4 (+3)	3 (+3)	2 (+3)	1 (+3)	I1	I1	I1	–	–
Actor A2	6	5	4 (+3 from I2)	3 (+3)	2 (+3)	1 (+3)	I2	I2	I2	–

TABLE I
EXAMPLE: DECREASING THE DISTANCE

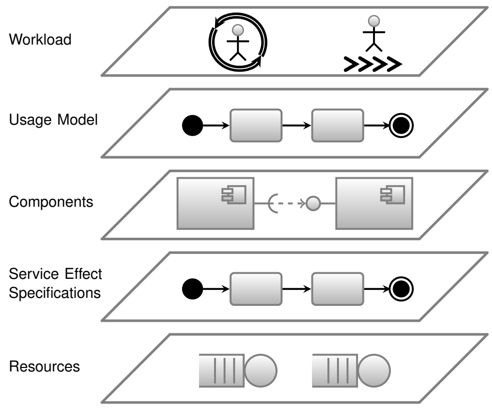

Fig. 2. Simulation Layers Used in the PCM

of two time units. Both instances put a demand of three time units on the actors. At the point in time t_0 the process instance I1 is allocated to the actor A1, because the queue of A1 is the shortest. At the point in time t_2 the process instance I2 is allocated to actor A2, because its waiting queue is the shortest, now (see Table). The processing of I1 is finished at the point in time t_7. The processing of I2 is finished at the point in time t_8. The distance between both process instances has decreased to one time unit. If A1 and A2 both had a waiting queue of length five time units at t_0, the processing of both process instances would have finished at the same time.

In this case, I1 and I2 reach the system step "IT: Calculate shortest route" at the same time which, according to Mi et al. [8], may results in a higher mean response time than if the process instances reach the system step at a distance of two time units.

IV. OPEN ISSUES IN PALLADIO

Considering the mutual impact of processes and IT systems described above, the following open issues currently appear when using Palladio and a business process simulation isolated from each other to simulate IT systems used in a business process.

O1: In the PCM usage model (cf. Figure 2) actor steps are considered as "black boxes" representing a delay. The execution time of actor steps (i.e. the extent of the delay) is not determined in simulation but has to be specified before simulation as an input. Thus, execution times represent assumptions made before simulation. The extent of the execution time of an actor step depends on the utilization of the corresponding actor which is unknown before simulation.

Palladio requires a specification of the usage of the IT system in the form of one or more usage scenarios. Using process simulation and IT simulation isolated of each other requires deriving an IT usage profile (i.e. a usage scenario) for IT simulation from the business process specification. The following open issues appear while deriving an IT usage profile:

O2: Variation of process arrival distribution during simulation cannot be easily mapped to a time-invariant IT workload specification as it is assumed in the PCM usage scenario. The execution of a business process typically lasts several hours or even days. Thus, there may be changes in the workload during simulation. For example, from 9:00 am to 11:30 am the process is triggered 100 times per minute, and from 12:30 pm to 5:00 pm, the process is triggered 50 times per minute. From 11:30 am to 12:30 pm, the process is not triggered at all, due to a lunch break. Figure 3 shows an example for time-variant workload on a resource and the resulting queue length assuming the resource has a processing rate of 53 requests per minute.

As shown in the example, inter-arrival time changes over time. The PCM assumes a time-invariant workload which can have a probabilistic distribution. Even so, changing arrival distributions cannot be mapped to a time-invariant workload without approximation. Approximation usually results in reduced accuracy of the predicted IT response times.

Moreover, deriving an IT usage profile includes the specification of actor steps and their execution time which is unknown at this stage (see O1).

O3: Although Palladio is already able to simulate system steps within the PCM usage model, systems steps cannot be included in the simulation of business process scenarios as the usage model currently does not provide business process model elements such as actor steps and existing simulators do not reflect business process simulation behavior such as simulating the utilization of actors.

O4: Workload distribution is only influenced by IT system steps. The impact of actor steps on workload distribution is neglected in current Palladio simulations. There may be some workarounds to manipulate workload distributions, e.g. by modeling delays as stochastic expressions. However, this is not an adequate way as it does not result from simulation but has to be specified before simulation. Thus, workload distributions are not correctly represented in current Palladio simulations. As a result, performance may not be predicted accurately (cf. [8]).

O5: Currently, the system steps contained in a PCM usage model are stochastically independent in terms of their parameters. Parametric dependencies are also possible for actor steps. Probabilistic parametric dependencies in the process impact on

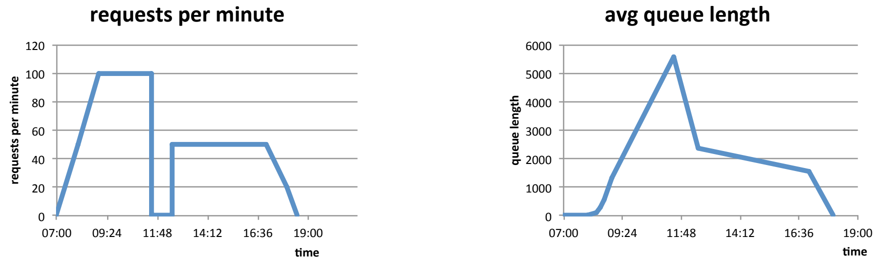

Fig. 3. Example: Time Dependent Workload on a Resource and Resulting Queue Length

workload as parameters such as the number of goods included in an order may increase the number of loop iterations required to handle the order. For example, when an order arrives at the IT system, it is first processed by the IT system optimizing the pickup sequence. Afterwards, the fork-lift driver has to pickup all the goods from the store. Therefore, not only the processing time of the order in the IT system is dependent on the number of goods included in the order, but also the actor step of picking up the goods. The more goods are ordered, the higher is the time required to process the steps.

V. Requirements on Palladio

In order to adequately represent the mutual impact of business processes and IT systems in simulation, Palladio has to fulfill the following requirements.

R1: In IT performance prediction the execution time of actor steps are determined by simulation instead of using assumptions. This requirement addresses O1.

R2: IT resources are demanded directly from the process simulation without deriving an IT usage profile. Thus, IT resource utilization directly results from the process model and the process workload without any approximation (see Section III-B). This requirement ensures that the process model and process workload specification correctly impacts on IT performance and thus addresses O2.

R3: In process performance prediction the response time of system steps are determined by simulation instead of using assumptions and system steps are considered as a factor of process performance. This requirement addresses O3 and is required to ensure that the impact of IT performance on process performance is adequately represented as described in Section III-C.

R4: In simulation the workload distribution within the process model is influenced by the utilization of actor waiting queues within actor steps and the utilization of IT resource waiting queues within system steps as described in Section III-D. This requirement addresses O4.

R5: Probabilistic parametric dependencies of actor steps and system steps within the process are considered in simulation. This requirement addresses O5.

VI. Extending Palladio by Business Process Simulation Concepts

In this section, we describe how to extend Palladio by process simulation concepts in order to meet the requirements introduced above. The proposed extensions include extensions to the PCM as well as extensions to the behavior of the simulation which are discussed in the following.

Figure 4 gives an overview of the proposed extensions on several layers. Extensions are colored blue in the figure. In the following sections we refer to certain layers of the figure to explain it in detail.

A. Extension of the Palladio Component Model

a) Process model: The PCM usage model is extended by the model element ActorStep in order to model steps performed by a human actor. Figure 5 shows the extension. ActorSteps are visualized as rectangles with rounded corners and a stickman icon in Figure 4. For each ActorStep the processing time has to be specified which is described in detail in the following. Moreover, delays commonly used in business process specifications (cf. [1]) can be documented and included in simulation. Idle time is the delay between the possible start and the actual start of the execution of the actor step. Resting time is the delay between the completion of an actor step and the start of the following step respectively the start of a transport. Transport time is the delay required to transport objects. Moreover, the model element Activity is added to enable the modeling of sub processes. The extension of the PCM usage model to a process model contribute to R1 as the modeling of actor steps is required to include them in simulation. It also contributes to R3 as now a process model is available for process performance prediction which also includes system steps. Moreover, the extension contributes to R2 as system steps are now triggered directly from the process model without deriving a usage profile.

b) Simulated resources: The PCM is extended by the organization environment model which is the counterpart of the hardware environment model. Figure 6 shows the organization environment model. It represents the organizational context of the process and contains basically three types of elements – ActorResources, Roles and DeviceResources. An ActorResource represents a human actor. ActorResources are visualized as circles around a stickman icon in Figure 4.

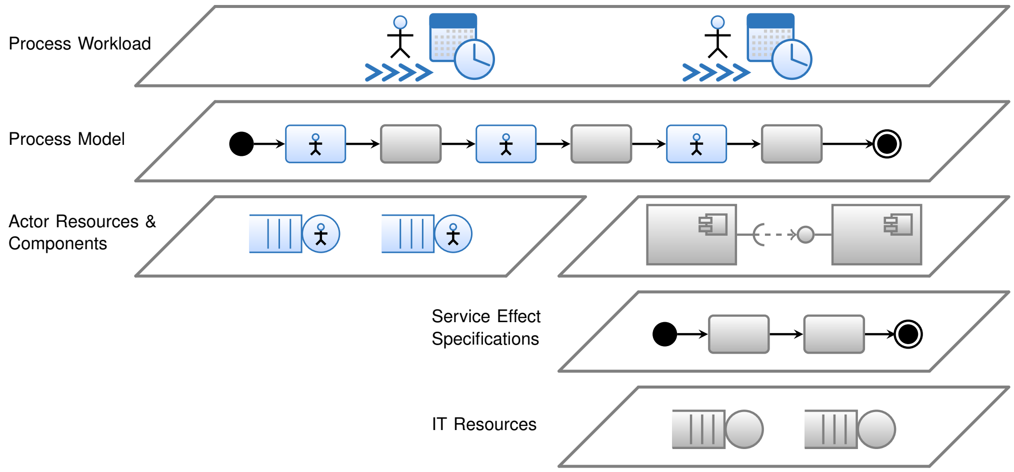

Fig. 4. Extended Simulation Layers

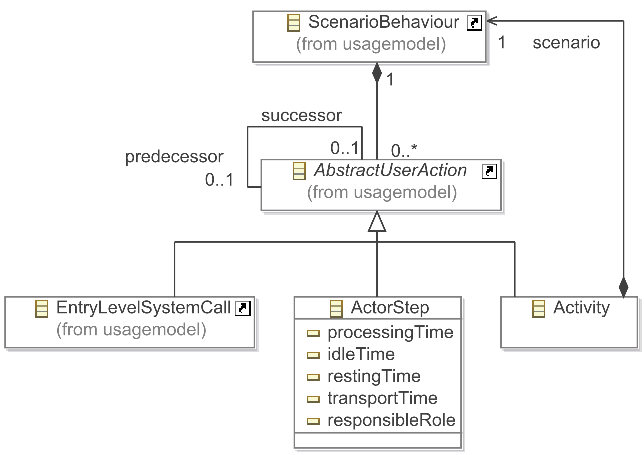

Fig. 5. Extension of the PCM Usage Model

In analogy to the ProcessingResource, which represent IT resources in Palladio [2], each ActorResource has a waiting queue in which the actor steps (i.e. the jobs) to be done by the ActorResource are lined up. In contrast to ProcessingResources, ActorResources are not continuously available, e.g. human actors have to take a lunch break or to sleep at night. Thus, the availability of each ActorResource has to be specified in one or more WorkingPeriods. Each WorkingPeriod is specified by a start time and an end time. Each ActorResource can have one or more Role(s) which group several ActorResources that have the same properties.

A DeviceResource is a device or machine which is required to perform an actor step of the process but does not actively process the step. Thus, it is called a passive resource. DeviceResources are shared by process instances and are available in a limited capacity. A process instance can acquire a DeviceResource required to perform one or more actor step(s) and release again when the actor step(s) are finished.

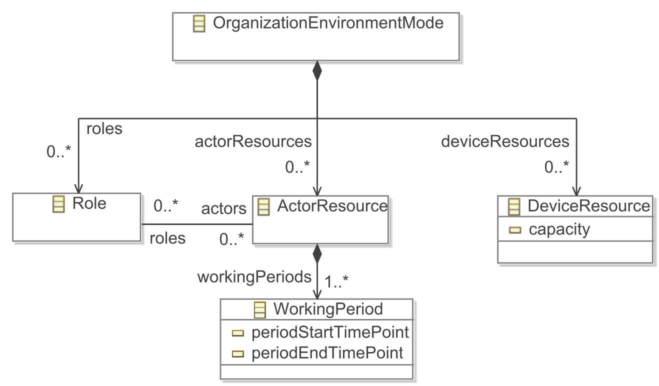

Fig. 6. Organization Environment Model

The extension of the simulated resources contributes to R1.

c) Modeling of demand on ActorResource: Actor steps specify resource demands in terms of their processing time as shown in Figure 5. Processing time is the time a human actor spends actively processing an actor step. The processing time is specified as a number of abstract time units. In contrast to resource demands on IT resources, work of human actors does not have a measuring unit such as CPU cycles or byte. So we decided to use the time required to perform a step as measuring unit. In contrast to system steps there is only one resource demand per actor step possible. Moreover, there is only one type of resource demand – demand on ActorResources. In contrast to ProcessingResources in Palladio, it is not directly specified which ActorResource performs a certain actor step but in each actor step it is specified which role is required to perform the step. The ActorResource that performs the step is selected dynamically in simulation. From the ActorResources that are available at the current point in simulation time, the ActorResource that has the shortest waiting queue in terms of the sum of processing time of the actor steps in the queue will be selected to perform the actor step. Dynamic selection of actors in simulation is common in process simulation (cf. [1]).

This extension contributes to R1.

d) Modeling of process workload: Besides the simulated resources and resource demands also the workload to be processed by the resources has to be specified for simulation. In Figure 4, workload is represented on the upper layer. In PCM, to each usage scenario within the usage model a workload driver is associated. Currently, there are two types of workload drivers in the PCM – closed workload and open workload. In a closed workload a population of *n* users execute the scenario concurrently [2]. In an open workload users enter the scenario at a specific arrival distribution [2]. Both workload drivers do not support changing process arrival distributions as described in Section IV (O2). We derived a new ProcessWorkload driver from the OpenWorkload driver as visualized in Figure 7. The workload of the process results from the arrival distributions of the ProcessTriggerPeriods associated to the ProcessWorkload driver. A ProcessTriggerPeriod specifies an interval of simulation time in which process instances start the execution of the process in simulation. Outside of the interval there is no start of process instances possible. Each ProcessTriggerPeriod consists of a start time point, an end time point and an inter-arrival time specification. Time designation in the ProcessTriggerPeriod follows the common date and time format which is more readable for human modelers than number of abstract time units usually used in Palladio. However, as Palladio only supports abstract time units, time designations are converted into abstract time units before simulation.

The workload on the IT system is a consequence of the process workload as system steps are invoked by the process instances. The introduction of the ProcessWorkload driver contributes to R2.

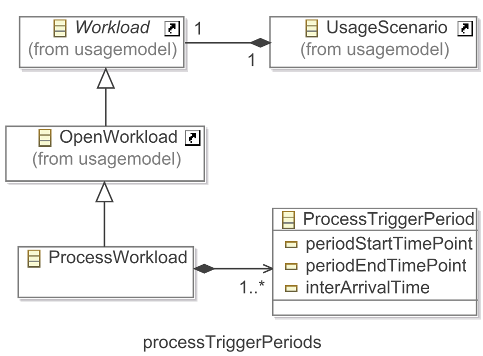

Fig. 7. Process Workload Driver

B. Extension of the Behavior of the Simulation

e) Simulation of process instance arrival: The simulation continuously generates process instances that traverse the process model. The start point of the first instance is the start time of the first ProcessTriggerPeriod of the process. Then, the distance to the start point of the next instance is generated randomly, based on the arrival distribution of the current ProcessTriggerPeriod allocated to the process, and added to the last start point. The next instance starts traversing the process model at that point in time. Instance start points will be generated and instances start traversing the process model until the generated start point of a process instance exceeds the end time of the last ProcessTriggerPeriod. This extension contributes to R2 as the simulation is now able to consider time-variant process workload.

f) Simulation of execution time: If a process instance reaches an actor step within the usage model, the actor step is put as a job into the waiting queue of an ActorResource allocated via his/her role to the actor step. The specific ActorResource is selected based on the length of its waiting queue and whether it is currently available (i.e. the current simulation time lies within a WorkingPeriod). Each ActorResource processes the actor steps in its waiting queue, e.g. following the FIFO principle. For actor steps, the processing time is already specified as resource demand. The waiting time is determined in simulation by waiting in the queue of the ActorResource. The resulting execution time of an actor step is the sum of its processing time and its waiting time at the corresponding ActorResource. This extension determines the execution time of actor steps in simulation and thus contributes to R1. The behavior of DeviceResource in simulation is comparable to the behavior of passive IT resources already contained in Palladio. They can be acquired and released again. Thus, the simulation of DeviceResource is not described in detail.

g) Simulation of workload distribution: In order to fulfill R4, simulation has to consider the mutual impact on workload distribution as described Section III-D. This is represented by the waiting queues of ActorResources and ProcessingResources in simulation. If a job is put in a waiting queue as the resource is busy at the time the job arrives at the resource, the flow of the process instance is hindered. For both kinds of steps, waiting times may vary from instance to instance. As a result, the distance between instances in the process model may change. Thus, workload distribution is manipulated by actor steps as well as system steps in the simulation.

h) Simulation of parametric dependencies: Currently, probabilistic parametric dependencies are limited to basic definitions in the usage model layer, i.e. it is only possible to define variable characterizations for input parameters. This is a good staring point to fully include parametric dependencies of actor steps as well as system steps to fulfill requirement R5. We can built upon the parametric dependency implementation already contained in EventSim [9] for simulating dependencies of Service Effect Specifications (SEFFs).

R3 is addressed as a consequence of the realization of the other requirements as system steps are already contained in the PCM usage model and are included in the simulation. As the PCM and the simulation behavior are extended by process simulation concepts, system steps are included in the process simulation.

C. Implementation-related Considerations

Extending the simulation behavior requires to extent the implementation of an existing simulator as well. Currently, there are two specialized software architecture simulators available for the PCM, SimuCom [2] and EventSim [9]. Furthermore, there exist transformations for translating PCM instances to the Layered Queueing Networks (LQNs) [10] and the Queueing Petri Nets (QPNs) [11] formalisms. The transformations and corresponding formalisms were successfully used in [12] and [13] to simulate complex Palladio models.

We decided to build upon the new event-driven simulator EventSim as the traversal strategy concept implemented in EventSim allows for a simpler extension of the simulation behavior as we expect it for SimuCom. Some of the existing traversal strategies can be reused or easily adapted for the new metamodel elements. Furthermore, using EventSim we can built upon the parametric dependency implementation.

Moreover, especially for high degrees of parallelism and concurrency, EventSim can perform simulations faster than SimuCom (cf. [9]). This is important, as process models usually have many process instances working concurrently. In addition, simulation speed is especially important in the business process context where simulated time is typically much longer than in the IT context. Simulated time in the business process context often spans months or even an entire year.

We decided against the LQN and QPN simulation approaches as these would require the development of additional model transformations to support the newly introduced elements and behavior.

D. Simulation-related Considerations

The different granularities of events in terms of their duration may limit the feasibility of the simulation. In cases where many events happen during a short time frame (e.g. a second) simulating a year may take a long time. Nevertheless, in order to get statistically significant results with workloads that vary over the day, many days have to be simulated. Also, if the actor steps may last several minutes, one needs to simulate longer. Thus, the combination of short running demands (milliseconds for IT events) on the one hand and long running demands (minutes for actor steps) as well as different time intervals (e.g. working time or breaks) on the other hand may cause large simulation times, as fine-grained simulation (which takes long per simulated minute) is required but also a long simulated time frame (e.g. a year) is needed. Especially the time and number of replications required for reaching a given confidence, when using a transient analysis, are a limiting factor. Thus, smart simulation strategies will be useful to circumvent these problems.

In our research, we focus on IT response times that may impact the business process performance as the extent of response time is comparable to the execution time of actor steps. Thus, we do not necessarily need to consider cases where the IT simulation has a large number of fine-grained events per second. Rough estimates of these events seem to be sufficient in the business process context. In the future, we want to analyze the feasibility of the integrated simulation based on several examples. If necessary, we want to investigate strategies to consider fine-grained IT events in a feasible way while simulating a long time frame. One possible strategy is to perform isolated fine-grained IT simulations prior to the integrated simulation for a set of representative classes of workload and workload distribution. During simulation the response time is then determined by looking up results from an equivalent class. Furthermore, an iterative refinement approach can be used to simulate critical time spans in more detail or add missing classes.

VII. Example of the Simulation Behavior

In this section, we continue the example introduced in Section III-A and sketch how the integrated simulation works.

Suppose there are two ProcessTriggerPeriods in the order picking process per working day. It is common that ProcessTriggerPeriods repeat for example per day or per week. From 8:00 am to 1:00 pm orders arrive and process instances start the execution of the process in a certain distribution. From 2:00 pm to 6:00 pm orders arrive and process instances start the execution in another distribution.

The simulation starts at 0:00 am of the first simulation day. The waiting queues of the ActorResources and the waiting queues of ProcessingResources are empty until the first process instance starts the execution of the process at 8:00 am of the first simulation day. At 8:00 am the first actor step "AS: gather data from order" is put in the waiting queue of an ActorResource that own the role shift leader. As at that point in simulation the waiting queues of all the ActorResources have the same length – they are all empty – the first ActorResource on a list of ActorResources that own the role shift leader is selected. For each ActorResource several working periods are defined. For example, from 8:00 am to 12:30 pm and from 1:00 pm to 6:00 pm. Between both periods there is a lunch break. In the course of simulation, actor steps line up in the waiting queues of ActorResources and internal actions of system steps line up in the waiting queues of ProcessingResources. Each ActorResource processes the actor steps in its waiting queue following the FIFO principle as long as the current simulation time is located within one of the ActorResource's WorkingPeriods. If the current simulation time exceeds a WorkingPeriod of the ActorResource, e.g. the current simulation time exceeds 12:30 pm, the ActorResource interrupts the processing of the actor steps until the current simulation time is again located in a WorkingPeriod. In the time between WorkingPeriods the process instances stuck in the waiting queue of the ActorResource and cannot reach another waiting queue, e.g. of a ProcessingResource. Being stuck in a waiting queue increases the waiting time for the corresponding instances. As ProcessingResources keep on processing jobs from their waiting queue during lunch break, waiting queues of ProcessingResources empty during lunch break and refill again in the afternoon as then another WorkingPeriod starts.

In the example process, the warehouser requires a fork-lift to put the goods on a location where they are picked up by a truck. Fork–lifts are shared DeviceResources which are available in a limited capacity. If all the fork–lifts are acquired at the moment, the warehouser has to wait until a fork-lift is released. Thus, in simulation the flow of process instances stuck. Waiting time is caused. The next system step "IT: Update database" is not reached by the process instance before a DeviceResource is released and acquired by the process instance. Waiting queues, e.g. of ProcessingResources demanded by the system step "IT: Update database", empty in meantime.

If a process instance is waiting for a passive IT resource, the flow of process instances stuck, too. Waiting time is caused which can impact the process performance, if its is long enough. Waiting queues, e.g. of ActorResources, empty in meantime.

As shown in the example, the integrated simulation of business processes and IT systems as proposed in this paper considers the process impact on waiting queues of ProcessingResources as well as the IT impact on waiting queues ActorResources. The other impact discussed in the paper is considered, too. Thus, we expect the integrated simulation to adequately represent the mutual performance impact of business processes and IT systems that occur in reality which results in increased performance prediction accuracy.

VIII. Conclusion and Future Work

In this paper, we discussed the mutual impact of business processes and involved IT systems in terms of performance. We pointed out the need for an integrated simulation of business process and IT and argued that Palladio is an adequate foundation to realize an integrated simulation. We also showed some open issues in Palladio to support business processes. We presented extensions of the PCM and the simulation behavior in order to enable the simulation of the mutual impact of business processes and IT systems.

Currently, we are implementing the extensions proposed in this paper in the new Palladio software architecture simulator EventSim. In a case study, we are currently investigating the mutual impact of business processes and IT systems in practice. We plan to use the extended tool support in the case study to perform what-if analysis on a process and IT system from practice.

In the future, we want to evaluate the prediction accuracy of the integrated simulation. We also plan to investigate the feasibility of a combined simulation of fine-grained IT events and long simulated time frames. The usability of the new features has to be improved and further evaluated. For example, a graphical representation of a calendar is useful for human modelers to specify ProcessTriggerPeriods quickly in the ProcessWorkload driver. In addition to the FIFO principle, further scheduling policies for actor resources are possible and will be explored in the future. Furthermore, a translation to the LQN and QPN formalisms could be developed to allow for more lightweight simulation.

IX. Acknowledgement

The authors want to thank Philipp Merkle for valuable comments and support related to the implementation in EventSim.

References

[1] S. Junginger, H. Kühn, F. Bartl, and J. Herbst, "Evaluation of financial service organizations with adonis simulation agents," in *Proceedings of the 10th European Simulation Symposium (ESS 98)*, 1998, pp. 582–588.

[2] S. Becker, H. Koziolek, and R. Reussner, "The Palladio component model for model-driven performance prediction," *Journal of Systems and Software*, vol. 82, pp. 3–22, 2009.

[3] "Jboss community, savara." [Online]. Available: http://www.jboss.org/savara

[4] S. Betz, E. Burger, A. Eckert, A. Oberweis, R. Reussner, and R. Trunko, *An approach for integrated lifecycle management for business processes and business software*. IGI Global, 2012.

[5] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. New York, NY, USA: Wiley-Interscience, 1998.

[6] W. M. C. Specification, *Workflow Management Coalition, Terminology & Glossary (Document No. WFMC-TC-1011)*. Workflow Management Coalition Specification, Feb. 1999.

[7] "Oxford dictionaries online." [Online]. Available: http://oxforddictionaries.com/

[8] N. Mi, G. Casale, L. Cherkasova, and E. Smirni, "Burstiness in multi-tier applications: symptoms, causes, and new models," in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, ser. Middleware '08. New York, NY, USA: Springer-Verlag New York, Inc., 2008, pp. 265–286.

[9] P. Merkle and J. Henss, "EventSim – an event-driven Palladio software architecture simulator," in *Palladio Days 2011 Proceedings*, ser. Karlsruhe Reports in Informatics ; 2011,32, S. Becker, J. Happe, and R. Reussner, Eds. Karlsruhe: KIT, Fakultät für Informatik, 2011, pp. 15–22.

[10] J. Rolia and K. Sevcik, "The method of layers," *Software Engineering, IEEE Transactions on*, vol. 21, no. 8, pp. 689–700, aug 1995.

[11] F. Bause, "Queueing Petri Nets˜- A formalism for the combined qualitative and quantitative analysis of systems," in *Proceedings of the 5th International Workshop on Petri Nets and Performance Models, Toulouse, France, October 19-22*, 1993.

[12] H. Koziolek and R. Reussner, "A model transformation from the palladio component model to layered queueing networks," in *Performance Evaluation: Metrics, Models and Benchmarks*, ser. Lecture Notes in Computer Science, S. Kounev, I. Gorton, and K. Sachs, Eds. Springer Berlin / Heidelberg, 2008, vol. 5119, pp. 58–78.

[13] P. Meier, S. Kounev, and H. Koziolek, "Automated Transformation of Palladio Component Models to Queueing Petri Nets," in *In 19th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2011), Singapore, July 25–27, 2011*, July 2011.