Business Process Simulation: How to get it right?

W.M.P. van der Aalst, J. Nakatumba, A. Rozinat, and N. Russell

Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands.
w.m.p.v.d.aalst@tue.nl

Abstract. Although simulation is typically considered as relevant and highly applicable, in reality the use of simulation is limited. Many organizations have tried to use simulation to analyze their business processes at some stage. However, few are using simulation in a structured and effective manner. This may be caused by a lack of training and limitations of existing tools, but in this paper we will argue that there are also several additional and more fundamental problems. First of all, the focus is mainly on design while managers would also like to use simulation for operational decision making (solving the concrete problem at hand rather than some abstract future problem). Second, there is limited support for using existing artifacts such as historical data and workflow schemas. Third, the behavior of resources is modeled in a rather naive manner. This paper focuses on the latter problem. It proposes a new way of characterizing resource availability. The ideas are described and analyzed using CPN Tools. Experiments show that it is indeed possible to capture human behavior in business processes in a much better way. By incorporating better resource characterizations in contemporary tools, business process simulation can finally deliver on its outstanding promise.

1 Introduction

The correctness, effectiveness, and efficiency of the business processes supported by a Process-Aware Information System (PAIS) [13] are vital to the organization. If a PAIS is configured based on a process definition which contains errors, then the resulting process may lead to angry customers, back-log, damage claims, and loss of goodwill. Moreover, an inadequate design may also lead to processes which perform poorly, e.g., long response times, unbalanced utilization of resources, and low service levels. This is why it is important to *analyze* processes before they are put into production (to find design flaws), but also while they are running (for diagnosis and decision support). In this paper, we focus on the role of *simulation* when analyzing business processes. The goal is to identify *limitations* of existing approaches and to discuss possible solutions. In particular, we will focus on the *availability of resources*. It will be shown that organizations have naive views on the availability of their employees and that today's simulation

tools do not support the more refined views that are needed. The goal is to transform simulation from a "toy for managers and consultants" into a truly useful and versatile tool.

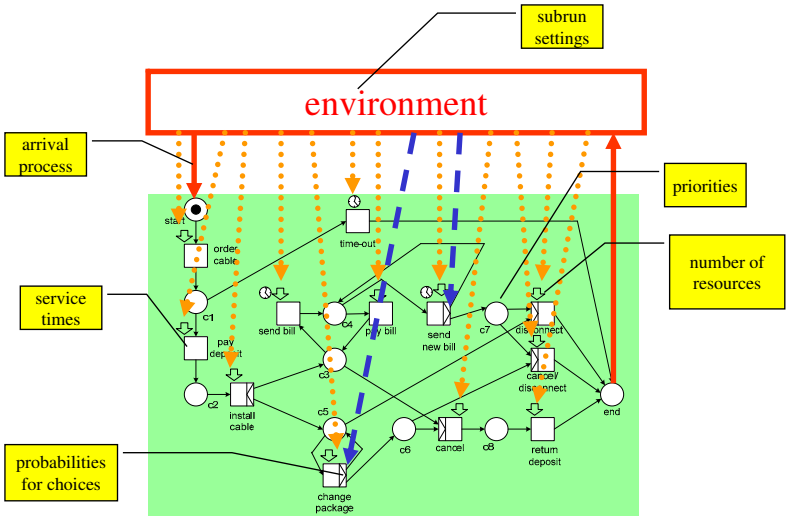

Fig. 1. Information required for a traditional simulation.

To introduce the concept of business process simulation, let us consider Figure 1. In the background a workflow specification is shown using the YAWL notation [4]. The process starts with task *order cable*. After this task is executed, task *pay deposit* is enabled. However, if payment does not follow within two weeks, task *time-out* is executed. The details of the process and the exact notation are not important. However, it is important to see that a workflow model defines the ordering of tasks, models (time) triggers, etc. The arrow above a task indicates that the task requires a resource of a particular type, e.g., using the role concept resources are linked to tasks. When there are choices, conditions are added to specify when to take a particular route, etc. The YAWL model in Figure 1 can be used to configure a PAIS (in this case the workflow management system YAWL) and thus enact the corresponding business process. However, the YAWL model is not sufficient for simulation. The ordering of activities and information about roles, conditions, triggers, etc. is useful for simulation purposes, but as Figure 1 shows, additional information is needed. First of all, an environment needs to be added. While in a PAIS the real environment interacts directly with the model, in a simulation tool the behaviorial characteristics of the environment need to be specified. For example, the arrival of new cases (i.e., process instances) needs to be specified (see box *arrival process* in Figure 1). Typically a Poisson arrival process is assumed and the analyst needs to indicate the average arrival rate. Sec-

ond, the service time, also called the process time, of tasks needs to be specified. For example, one can assume that the service time is described by a Beta distribution with a minimum, a maximum, an average, and a mode. Note that the simulation model needs to abstract from the actual implementation of the task and replace the detailed behavior by stochastic distributions. Similarly, choices, priorities, etc. are replaced by probability distributions. Finally, the workflow model needs to be complemented by information about resources (e.g., number of people having a particular role). In order to conduct experiments one also has to specify the number of subruns, the length of each subrun, etc. Based on all this information, simulation tools can provide information about, for example, expected flow times, service levels (e.g., percentage of cases handled within two weeks), and resource utilization.

Figure 1 presents a rather classical view on business process simulation. This is the type of simulation supported by hundreds, if not thousands, of commercial simulation packages. Some vendors provide a pure simulation tool (e.g., Arena, Extend, etc.) while others embed this in a workflow management system (e.g., FileNet, COSA, etc.) or a business process modeling tool (e.g., Protos, ARIS, etc.). All of these tools more or less use the information presented in Figure 1 to calculate various performance indicators. In this paper we will call this "traditional simulation". We will argue that this type of simulation is not very useful. Figure 2 shows the need to move beyond traditional simulation approaches.

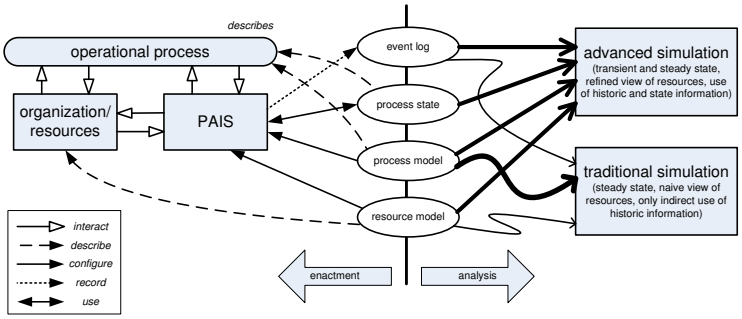

Fig. 2. Overview of the relationship between enactment and simulation and the different data sources.

The left-hand-side of Figure 2 shows the role of a PAIS (e.g., a workflow engine as well as also other types of process-oriented information systems) in supporting operational business processes. The PAIS supports, controls, and monitors operational processes. The resources within the organization perform tasks in such processes and therefore also interact with the PAIS. The PAIS can only do meaningful things if it has knowledge of the process, the resources within the organization and the current states of active cases. Moreover, a PAIS often records historical information for auditing and performance analysis. The

four ellipses in the middle of Figure 2 show these four types of data: (1) event log, (2) process state, (3) process model, and (4) resource model. The *event log* contains historical information about "When, How, and by Whom?" in the form of recorded events. The *process state* represents all information that is attached to cases, e.g., Customer order XYZ consists of 25 order lines and has been in the state "waiting for replenishment" since Monday. The *process model* describes the ordering of tasks, routing conditions, etc. (cf. the YAWL model in Figure 1). The *resource model* holds information about people, roles, departments, etc. Clearly, the process state, process model, and resource model are needed to enact the process using a PAIS. The event log merely records the process as it is actually enacted.

The right-hand-side of Figure 2 links the four types of data to simulation. For traditional simulation (i.e., in the sense of Figure 1) a process model is needed. This model can be derived from the model used by the PAIS. Moreover, information about resources, arrival processes, processing times, etc. is added. The arcs between the box *traditional simulation* and the three types of data (event log, process model, and resource model) are curved to illustrate that the relationship between the data used by the PAIS and the simulation tool is typically rather indirect. For example, the analyst cannot use the process model directly, but needs to transform it to another language or notation. The resource model used for simulation is typically very simple. Each activity has a single role and for each role there are a fixed number of resources available. Moreover, it is assumed that these resources are available on a full-time basis. The event logs are not used directly. At best, they are used to estimate the parameters for some of the probability distributions. Hence, traditional simulation can be characterized as having a weak link with the actual PAIS and historical data and a rather naive view of resources. Moreover, the current state is not used at all. As such, simulation focuses on steady-state behavior and cannot be used for operational decision making.

This paper advocates the use of more advanced notions of simulation. Key aspects of which include the establishment of a close coupling with the data used by the PAIS together with the extensive use of event log and process state information. Moreover, we will not only focus on steady-state behavior but also on transient behavior in order to also support operational decision making. This is illustrated by the box *advanced simulation* in Figure 2. The contribution of this paper is twofold:

- First of all, we provide a *critical analysis of current simulation approaches and tools* as summarized by Figure 2. We argue that there is too much focus on process design and that there should be more emphasis on operational decision making using transient analysis. We also advocate the use of existing artifacts such as workflow models, event logs, state information, etc. It is our belief that vital information remains unused in current approaches. In our analysis of current simulation approaches, we also address the problem that resources are modeled in a way that does not reflect the true behavior of

people. For example, the working speed may depend on the utilization of people and people may prefer to work in batches.
- Second, we provide a *detailed analysis of the effect of resource availability* in simulation studies. We show that many assumptions about resources do not hold, and using a concrete model, we prove that the effects of these assumptions can be dramatic. As a result, the simulation model may indicate that the average flow time is around one hour while in reality the average flow time is actually more than one month.

The remainder of this paper is organized as follows. First, we provide an overview of the limitations of traditional simulation approaches. Then we look into the problem of describing resource availability. We develop a simple CPN model with which to do simulation experiments and use these results to show the effects of oversimplifying the availability of people. After providing concrete suggestions for improving the modeling of resources, we discuss related work and complementary approaches, and conclude the paper.

2 Pitfalls of Current Simulation Approaches

In the introduction, we used Figure 2 to summarize some of the limitations of contemporary simulation approaches. In this section, we describe these pitfalls in more detail.

2.1 Focus on Design Rather than Operational Decision Making

Simulation is widely used as a tool for analyzing business processes but it mostly focuses on examining rather abstract steady-state situations. Such analyses are helpful for the initial design of a business process but are less suitable for operational decision making and continuous improvement. To explain this we first elaborate on the difference between *transient analysis* and *steady-state analysis*.

The key idea of simulation is to execute a model repeatedly. The reason for doing the experiments repeatedly, is to not come up with just a single value (e.g., "the average response time is 10.36 minutes") but to provide confidence intervals (e.g., "the average response time is with 90 percent certainty between 10 and 11 minutes"). This is why there is not a single simulation run, but several *subruns*. Figure 3 shows two sets of four subruns. (Typically, dozens of subruns are used to calculate confidence intervals and, in the case of steady-state analysis, subruns can be obtained by partitioning one long run into smaller runs [19, 24].) In the four subruns depicted in Figure 3(a) the focus is on the initial part of the process, i.e., starting from the initial state the "near future" is explored. In the four subruns depicted in Figure 3(b) the initial part is discarded and only the later behavior is of interest. Note that for steady-state analysis the initial state is irrelevant. Typically, the simulation is started "empty" (i.e., without any cases in progress) and only when the system is filled with cases the measurements

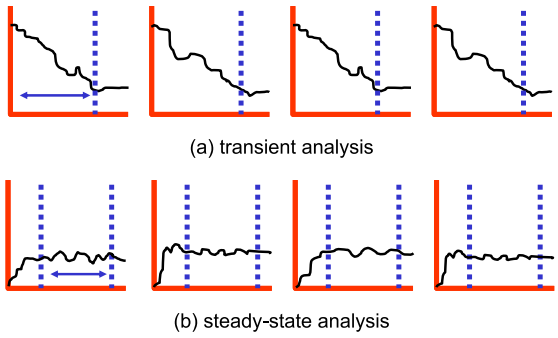

Fig. 3. For transient analysis the initial state is vital while for steady-state analysis the choice of initial state should have no effect on the simulation result.

start. Figure 3(a) clearly shows that for transient analysis the initial state is very important. If the simulation starts in a state with long queues of work, then in the near future flow times will be long and it may take some time to get rid of the backlog as shown in the diagram.

Despite the abundance of simulation tools, simulation is rarely used for operational decision making. One of the reasons is the inability of traditional tools to capture the real process (see above). However, another, perhaps more important, reason is that existing simulation tools aim at strategic or tactical decisions. Contemporary tools tend to support simulations that start in an arbitrary initial state (without any cases in the pipeline) and then simulate the process for a long period to make statements about the steady-state behavior. However, this steady-state behavior does not exist (the environment of the process changes continuously) and is thus considered irrelevant by the manager. Moreover, the really interesting questions are related to the near future. Therefore, it seems vital to also support transient analysis, often referred to as *short-term simulation* [22, 32, 26]. The 'fast-forward button' provided by short-term simulation is a useful option, however, it requires the use of the current state. Fortunately, when using a PAIS it is relatively easy to obtain the current state and load this into the simulation model.

2.2 Modeling from Scratch Rather Than Using Existing Artifacts

Another pitfall of current approaches is that existing artifacts (models, logs, data, etc.) are not used in a direct manner. If a PAIS is used, there are often models that are used to configure the system (e.g., workflow schemas). Today, these models are typically disconnected from the simulation models and created separately. Sometimes a business process modeling tool is used to make an initial process design. This design can be used for simulation purposes when using a tool like Protos or ARIS. When the designed process is implemented, another system is used and the connection between the implementation model and the

design model is lost. It may be that at a later stage, when the process needs to be analyzed, that a simulation model is built from scratch. This is a pity as the PAIS contains most of the information required. As a result the process is "reinvented" again and again, thus introducing errors and unnecessary work. The lack of reuse also applies to other sources of information. For example, the PAIS may provide detailed event logs. Therefore, there is no need to "invent" processing times, arrival times, and routing probabilities, etc. All of this information can be extracted from the logs. Note that all additional information shown in Figure 1 can be derived from event logs. In fact, in [25] it is demonstrated that complete simulation models can be extracted from event logs.

As indicated in Figure 2, simulation could use all four types of data provided by the PAIS, i.e., not just the event log and process model but also the process state and resource model. The process state can be used to enable short-term simulation (as described before) and the resource model may be used to more accurately describe resources. In most simulation tools, only the number of resources per class is given. However, a PAIS holds detailed information about authorizations, delegations, working times, etc. By using this information directly, more realistic models can be constructed.

It is interesting to note that today's data mining and business intelligence tools are completely disconnected from simulation. These tools are merely used to measure performance indicators and to discover correlations and trends. Yet their objectives are similar, i.e., both simulation and data mining/business intelligence tools aim at improving operational business processes. Therefore, it seems good to combine things and exploit existing artifacts as much as possible.

2.3 Incorrect Modeling of Resources

Probably the biggest problem of current business simulation approaches is that human resources are modeled in a very naive manner. As a result, it is not uncommon that the simulated model predicts flow times of minutes or hours while in reality flow times are weeks or even months. Therefore, we list some of the main problems encountered when modeling resources in current simulation tools.

People are involved in multiple processes. In practice there are few people that only perform activities for a single process. Often people are involved in many different processes, e.g., a manager, doctor, or specialist may perform tasks in a wide range of processes. However, simulation often focuses on a single process. Suppose a manger is involved in 10 different processes and spends about 20 percent of his time on the process that we want to analyze. In most simulation tools it is impossible to model that a resource is only available 20 percent of the time. Hence, one needs to assume that the manager is there all the time and has a very low utilization. As a result the simulation results are too optimistic. In the more advanced simulation tools, one can indicate that resources are there at certain times in the week (e.g., only on Monday). This is also an incorrect

abstraction as the manger distributes his work over the various processes based on priorities and workload. Suppose that there are 5 managers all working 20 percent of their time on the process of interest. One could think that these 5 managers could be replaced by a single manager (5*20%=1*100%). However, from a simulation point of view this is an incorrect abstraction. There may be times that all 5 managers are available and there may be times that none of them are available.

People do not work at a constant speed. Another problem is that people work at different speeds based on their workload, i.e., it is not just the distribution of attention over various processes, but also their absolute working speed that determines their capacity for a particular process. There are various studies that suggest a relation between workload and performance of people. A well-known example is the so-called Yerkes-Dodson law [31]. The Yerkes-Dodson law models the relationship between arousal and performance as an inverse U-shaped curve. This implies that for a given individual and a given type of tasks, there exists an optimal arousal level. This is the level where the performance has its maximal value. Thus work pressure is productive, up to a certain point, beyond which performance collapses. Although this phenomenon can be easily observed in daily life, today's business process simulation tools do not support the modeling of workload dependent processing times.

People tend to work part-time and in batches. As indicated earlier, people may be involved in different processes. Moreover, they may work part-time (e.g., only in the morning). In addition to their limited availabilities, people have a tendency to work in batches (cf. Resource Pattern 38: Piled Execution [27]). In any operational process, the same task typically needs to be executed for many different cases (process instances). Often people prefer to let work-items related to the same task accumulate, and then process all of these in one batch. In most simulation tools a resource is either available or not, i.e., it is assumed that a resource is eagerly waiting for work and immediately reacts to any work-item that arrives. Clearly, this does not do justice to the way people work in reality. For example, consider how and when people reply to e-mails. Some people handle e-mails one-by-one when they arrive while others process their e-mail at fixed times in batch.

Process may change depending on context. Another problem is that most simulation tools assume a stable process and organization and that neither of them change over time. If the flow times become too long and work is accumulating, resources may decide to skip certain activities or additional resources may be mobilized. Depending on the context, processes may be configured differently and resources may be deployed differently. In [6] it is shown that such "second order dynamics" heavily influence performance.

The pitfalls mentioned above illustrate that simulation techniques and tools have

a very naive view of business processes. As a result, the simulation results may deviate dramatically from the real-life process that is modeled. One response could be to make more detailed models. We think that this is not the best solution. The simulation model should have the right level of detail and adding further detail does not always solve the problem. Therefore, we propose to use the data already present in a PAIS more effectively. Moreover, it is vital to characterize resources at a high abstraction level. Clearly, it is not wise to model a person as a full-time resource always available and eager to work, nor should we attempt to make a detailed model of human behavior. In the next section, we try to characterize resource availability using only a few parameters.

3 Resource Availability: How to Get it Right?

The previous section listed several pitfalls of contemporary simulation approaches. Some of these pitfalls have been addressed in other papers [6, 25, 26]. Here, we focus on the *accurate modeling of resource availability*. This can be used to capture various phenomena, e.g., people working in multiple processes or working part-time and the tendency of people to work in batches.

3.1 Approach

As already indicated in this paper, there are a number of issues that need to be considered when modeling resources. These issues deal with the way people actually carry out their work. The first issue is that people are not available to work all the time but for specific periods of time. In most cases, people are only part-time available (e.g., in the mornings, or only on the weekends). In this paper, this is described as the *availability* (denoted by a) of the resource, and it is the percentage of time over which a person is able to work. Secondly, when people are available to work, they divide up their work into portions which are called *chunks* and the size of a chunk is denoted by c. Chunk sizes may vary among different people, for example, a person that is available for 50% of his time may work whenever there is work and he did not exceed the 50% yet (i.e., small chunk size), or only in blocks of say half a day (i.e., large chunk size). Another case is that a person may save up work and then work for an extended period (large c) while another person prefers to regularly check for new work items and work on these for a shorter period of time (small c). The chunks of work to be done are distributed over particular *horizons* of length h. This is the time period over which constraints can be put in place.

Figure 4 shows the relationship between chunk size and horizon. The empty circles represent case arrivals, i.e., the points in time where a new work-item is offered. The filled circles represent case completions, i.e., the points in time where some work-item is completed. The chunks of work are divided over the horizon (see the double headed arcs labeled with c). The periods where the resource is actually working, is denoted by the horizontal bars. A resource can have three states:

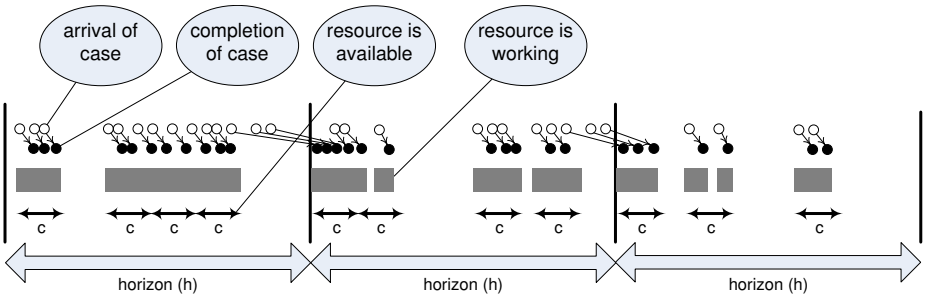

Fig. 4. Overview of the relation between horizon (h) and chunk size (c).

- *Inactive*, i.e., the resource is not allocated to the process because there is no work or because all available capacity has been used.
- *Ready*, i.e., the resource is allocated to the process but there is currently no work to be done.
- *Busy*, i.e., the resource is allocated to the process and is working on a case.

When a case arrives, and the resource is inactive and still has remaining chunks of time (given the current horizon), then a chunk of time is allocated and the resource starts working. If a case arrives and the resource is busy, the work is queued until the resource becomes available. Note that it may be the case that work cannot be completed in the current horizon and is postponed to the first chunk in the next period of length h, as illustrated in Figure 4. Furthermore, if a chunk has been started then it will be completed even though there might be no work left (in this case the resource is in the ready state).

The main parameters of the model are as follows.

- *Arrival rate* λ, i.e., the average number of cases arriving per time unit. We assume a Poisson arrival process, i.e., the time between two subsequent arrivals is sampled from a negative exponential distribution with mean $\frac{1}{\lambda}$. Note that $\lambda > 0$.
- *Service rate* μ, i.e., the average number of cases that can be handled per time unit. The processing time is also sampled from a negative exponential distribution. The mean processing time is $\frac{1}{\mu}$ and $\mu > 0$.
- *Utilization* $\rho = \frac{\lambda}{\mu}$ is the expected fraction of time that the resource will be busy.
- *Chunk size* c is the smallest duration a resource is allocated to a process. When a resource leaves the inactive state, i.e., becomes active (state ready or busy), it will do so for at least a period c. In fact, the active period is always a multiple of c.
- *Horizon* h is the length of the period considered ($h > 0$).
- *Availability* a is the fraction of time that the resource is available for the process ($0 < a \leq 1$), i.e., the resource is inactive at least $1 - a$ percent of the time.

Not all combinations of these parameters makes sense, as is illustrated by the following requirements.

- $\rho = \frac{\lambda}{\mu} \leq a$, i.e., the utilization should be smaller than the availability.
- $c \leq h$, i.e., the chunk size cannot be larger than the horizon.
- $(a * h)\ mod\ c = 0$, i.e., the maximum time a resource can be active each period should be a multiple of c, otherwise it would never be possible to actually use all of fraction a.

We use an example to explain the last requirement. Suppose that the horizon is 8 hours, the availability is 0.5, and the chunk size is 3 hours. In this case, $a * h = 4$ hours and $c = 3$ hours. Now it is obvious that per period only one chunk can be allocated. Hence, the effective availability is not 4 hours but just 3 hours (i.e., effectively $a = \frac{3}{8}$). Therefore, we require that $a * h$ is a multiple of c.

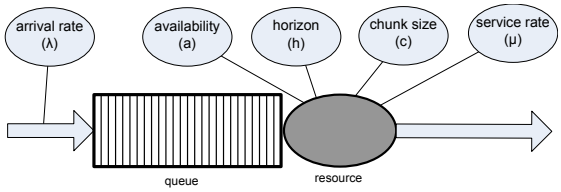

Fig. 5. Cases arrive with intensity λ and are placed in a queue and a resource with parameters a, c, h, and μ handles each case.

Figure 5 summarizes the parameters used in our basic model. Cases arrive with a particular arrival rate λ and are then placed in a queue. A resource, described by four main parameters (availability a, horizon h, chunk size c and service rate μ), is then made available to work on the case as shown in Figure 5. A resource will work on the first case in the queue. If the case is not completed within a particular chunk, then it is sent back to the beginning of the queue to wait for the next chunk to be allocated.

3.2 Modeling in terms of CPN Tools

We analyzed the effects of the various resource characteristics using a simulation model. Colored Petri Nets (CPNs) [17, 18] were used as a modeling language. CPNs allow for the modeling of complex processes. Using CPN Tools [18] such models can be analyzed in various ways, i.e., simulation, state-space analysis, etc. Our CPN model is a hierarchical model that is divided into 3 pages which are: the *generator* page (which creates cases for which a task needs to be performed), the *activation* page (which models the availability of resources), and the *main* page (which models the actual execution of tasks). This CPN model is used to clearly

study the behavior of a single resource, but it can easily be extended to more realistic situations with different resources (see Section 3.4). In the following we briefly describe each of the 3 pages of the CPN model[1].

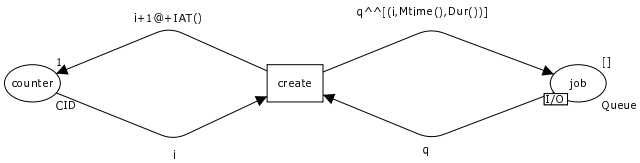

Fig. 6. The *generator* page. The time between two subsequent case arrivals is given by the function $IAT()$, the creation time of cases is recorded by the current model time function $Mtime()$, and the duration of the task is given by the function $Dur()$.

Figure 6 shows the *generator* page of the CPN model. Cases arrive through this page and are put in a queue. We assume the arrival process to be Poisson with parameter λ (i.e., negative exponential inter-arrival times with mean $\frac{1}{\lambda}$). The cases themselves are represented by tokens with a value which is a product of three parameters: *caseid*, *arrival time*, and *duration* (the processing time is sampled from a negative exponential distribution with mean $\frac{1}{\mu}$).

The modeling of the availability of resources is done in the activation page of the CPN model shown in Figure 7. We consider the variables introduced earlier in this section, i.e., h, a, and c. The token in place *resource info* holds details about a resource with values related to its availability a, chunk size c, and horizon h. It is important at this point to determine the amount of work that a person can do. This is obtained by multiplying availability by horizon. The availability can be distributed over the period h in chunks of size c. Not more than $(a * h)$ *div* c chunks can be allocated and allocation is eager, i.e., as long that there is work to be done and available capacity, the resource is active. When transition *activate* fires, then a resource with the parameters r and $Mtime() + c$ becomes available to work. The resource will have a delay attached to it which is equivalent to the current time plus the chunk size, i.e., the resource will be active for c time units and this period ends at time $Mtime() + c$.

The actual processing of the cases is carried out in the main page shown in Figure 8. This page uses the *generator* and *activation* pages described above. Cases come in from the generator page through the place *job* and a resource is made available from the activation page through the place *ready*. The token in place *busy* indicates the actual processing of a case by the resource. The length of the processing of a case is restricted by the task duration (already sampled during case creation) and the remaining chunk size. If cases leave place *busy* but

[1] The interested reader can look up the declarations that would initialize this model with $\lambda = \frac{1}{100}$, $\mu = \frac{1}{15}$, and one resource "r1" characterized by $h = 1000$, $a = 0.2$, and $c = 200$ in Appendix A.

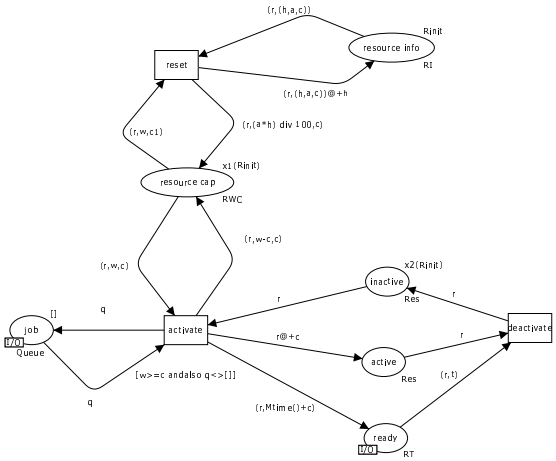

Fig. 7. The *activation* page. Transitions *activate* and *deactivate* control the actual availability of resources.

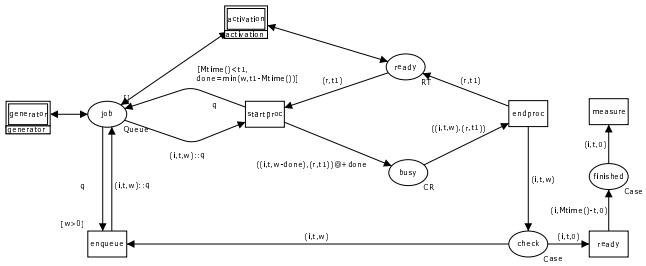

Fig. 8. The *main* page. Place *busy* shows the interaction between a resource and a case while *ready* shows that a resource is available to process a case.

are still incomplete, because the resource is no longer available for a time period sufficient to complete the case, then these cases are put back on the queue. When the processing of the case is completed the resource is made available to work again, but this is only possible if there is still time left in the current chunk. Otherwise, the resource will be deactivated and is no longer available until the next chunk is allocated. The deactivation is controlled by the *activation* page shown in Figure 7.

The CPN model just described specifies the resource behaviors considered. As indicated before, we assume a very basic setting and the model can easily be extended. However, our goal is to show that parameters such as availability a, chunk size c, and horizon h really matter. Most business simulation tools do not provide such parameters and assume $a = 1$ (always available), $c \to 0$ (resources are only active if they are actually busy working on a case), and $h \to \infty$ (infinite horizon). The next subsection shows that this may result in unrealistic simulations with huge deviations from reality.

3.3 Experiments

Using the CPN model, experiments were carried out to investigate the relationship between the flow time of cases and the main parameters related to resource availability. Monitors were used to extract numerical data during the simulation. The monitor concept of CPN Tools allows for the measurement of various performance indicators without changing or influencing the model [10]. All experimental results reported here are based on a simulation with 10 subruns, each subrun having 10,000 cases. For each performance indicator measured, we calculated the so-called 90% confidence interval. These are shown in the graphs but are typically too narrow to be observed (which is good as it confirms the validity of the trends observed).

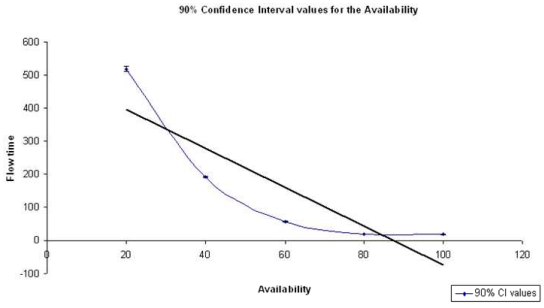

Fig. 9. Graph showing availability against flow time ($\lambda = \frac{1}{100}$, $\mu = \frac{1}{15}$, $\rho = 0.15$, $c = 200$, and $h = 1000$). The flow time reduces as the availability increases.

As discussed already, the availability a of a resource is the percentage of time over which a person is able to work. In the CPN model, different availability values were investigated while keeping the chunk size and horizon constant. The results from the experiment are shown in Figure 9. The graph was plotted to show the values of the averages with a 90% confidence interval and in the caption all fixed parameter values are shown. The idea behind this experiment was to determine whether one's availability has any effect on the flow time. The result is obvious: the more people are available, the more work they can do and the shorter the flow time is. However, one should realize that in most simulation tools it is not possible to set the parameter a, i.e., a 100% availability ($a = 1$) is assumed. Figure 9 shows that this may lead to severe discrepancies.

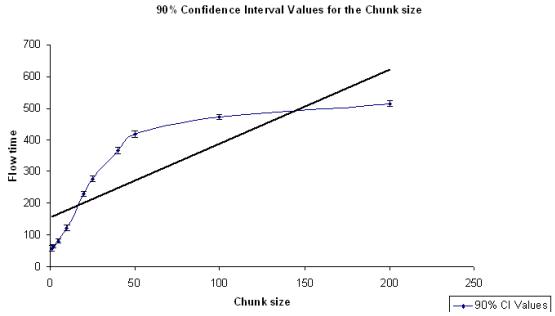

Fig. 10. Graph showing chunk size against flow time ($\lambda = \frac{1}{100}$, $\mu = \frac{1}{15}$, $\rho = 0.15$, $a = 0.2$, and $h = 1000$). The flow time increases as the chunk size increases.

While the effect of reduced availability may be obvious, the effect of the chunk size c on the flow time may be more surprising. People can divide up their work into chunks of varying sizes. When availability is distributed over chunks, the bigger the chunk, the larger the flow times of cases. This is because work is more likely to accumulate. The results obtained from the experiments carried out with different chunk sizes (while keeping all other parameters constant) are shown in Figure 10. The graph shows the values of the average flow times and the 90% confidence intervals. Our findings indeed confirm that flow time increases as the chunk size increases. The reason is that the larger the chunk size, the longer the periods between chunks become. Figure 10 shows an important insight that people making simulation models often do not realize.

When a horizon is large, then the distribution of chunks is more flexible. If $a * h = c$, then only one chunk per period is possible. This chunk will typically start in the beginning and if a is small, then for a large part of h no resource is available. If $a * h$ is much larger than c, then more chunks are possible and these can be more evenly distributed over the period h. Note that the effect of making the horizon longer is similar to making the chunk size smaller. Figure 11 shows

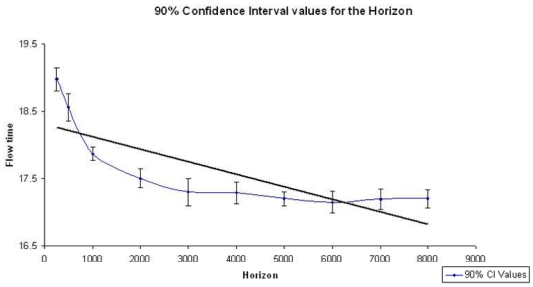

Fig. 11. Graph showing the horizon against the flow times ($\lambda = \frac{1}{100}$, $\mu = \frac{1}{15}$, $\rho = 0.15$, $c = 200$, and $a = 0.8$). The flow time decreases as the horizon increases.

the relation between flow time and horizon observed and clearly shows that shortening the horizon may lead to longer flow times. However, if the horizon is sufficiently large (in this case more than 3000), it does not seem to matter anymore.

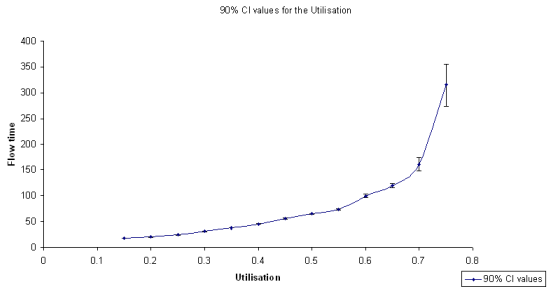

Fig. 12. Graph showing utilization against flow time ($\mu = \frac{1}{15}$, $c = 200$, $a = 0.8$, and $h = 1000$). The flow time increases as utilization increases.

Finally, it is important to measure the effect of utilization on the flow times of cases. With a higher utilization, the flow times obviously increase as shown in Figure 12. Typically, flow times dramatically increase when ρ get close to 1. However, with limited availability, the flow time dramatically increases when ρ gets close to a. Figure 12 shows the average flow times with 90% confidence intervals. Note that ρ results from dividing λ by μ. In this graph we keep μ constant and vary λ to get different utilization values. As expected, the confidence intervals get wider as ρ approaches a.

3.4 Example

This section describes a model that deals with the handling of claims in an insurance company (taken from [3]). The insurance company processes claims that result from accidents with cars where the customers of the insurance company are involved. Figure 13 shows the workflow modeled in terms of a Petri net using the YAWL notation [4]. A claim reported by a customer is registered by an employee of department Car Damages (CD). After registration, the insurance claim is classified by a claim handler of department CD. Based on this classification, either the claim is processed or a letter is sent to the customer explaining why the claim cannot be handled (50% is processed and 50% is not handled). If the claim can be handled, then two tasks are carried out which are *check_insurance* and *phone_garage*. These tasks are executed in parallel and are handled by employees in department CD. After executing these tasks, the claim handler makes a decision which has two possible outcomes: OK (positive) and NOK (negative). If the outcome is OK, then the claim is paid and a letter is sent to the customer. (Half of the decisions lead to a payment and the other half not.) Otherwise, just a letter is sent to the customer.

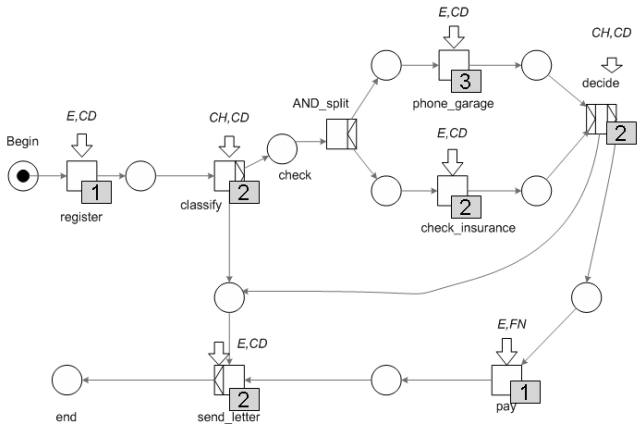

Fig. 13. Workflow model of the Insurance Company.

Each of the tasks shown in Figure 13 corresponds to an instance of the CPN model explained in Section 3.2. Each task shown in the workflow model has a number attached to it and this corresponds to the number of people (potentially) available to carry out that task. For example, there is one person able to execute task *register* and there are two persons able to execute task *classify*. The workflow model was implemented in CPN Tools and Figure 14 shows the main page of the CPN model.

Initially, a base scenario was chosen with suitable values for the chunk size, horizon, availability and utilization of the resources. Based on these values, ex-

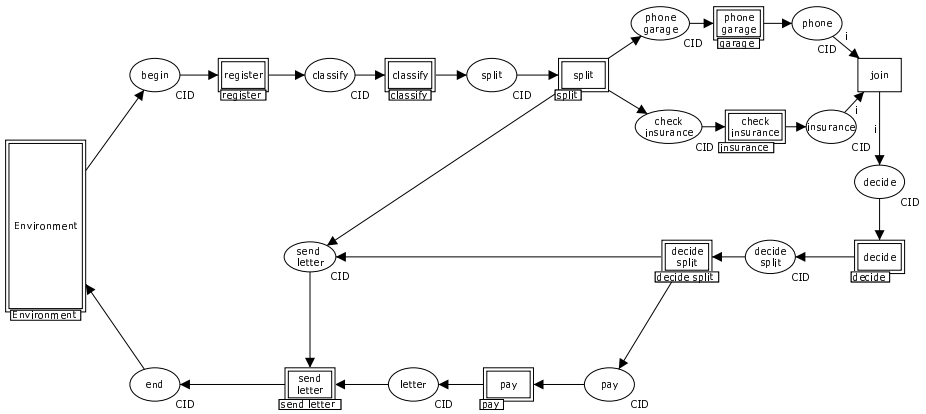

Fig. 14. The *Main* page. The sub page *Environment* creates cases. After completing all the steps in the process, cases are sent back to the environment to measure their flow times.

periments were carried out to determine the sensitivity of these parameters with respect to the flow time. For example, we were interested to see whether the flow time was affected by larger chunk sizes or not. Table 1 summarizes the values of the flow times obtained when experiments with different parameters were varied. Appendix B lists the parameters of the individual tasks, e.g., task *register* takes on average 18 minutes ($\mu_a = \frac{1}{18}$) and the time between two subsequent arrivals is 50 minutes on average ($\lambda_a = \frac{1}{50}$). Since the two choices in the model split the flow with equal probabilities, only 25% of the cases that arrive are actually payed (i.e., 50% is classified as relevant and of these 50% again 50% is rejected). Tasks have different processing times, but, for simplicity, all the tasks share the same value for chunk size (c), horizon (h), and availability (a). In the base scenario: $c = 5$, $h = 2000$, and $a = 0.4$. The flow time is with 90% confidence within $757.6 - 65.0$ and $757.6 + 65.0$ minutes.

The results shown in Table 1 indeed confirm that the parameters for chunk size (c), horizon (h), and availability (a) have dramatic effects on the flow times. Based on the initial values, variations were made and different flow time values were obtained. For example, when the chunk size was increased from $c = 5$ to $c = 100$ the flow time more than doubled. When the availability and the horizon were varied, the effects were as expected. For example, when the availability and arrival rate decrease by a factor 4 (i.e., the relative utilization ρ/a remains unchanged) the flow time goes up from approx. 757 to approx. 3420. Our experiments confirm that the parameters identified in this paper are relevant. In fact, it is easy to see that the effects accumulate when the workflow is larger.

Table 1. Results of experiments carried out to determine the effect of varying different parameters against the flow time.

		Parameters	**Flow Time**
a)		Base Case Scenario ($c = 5$, $h = 2000$, $\lambda = \frac{1}{50}$ and $a = 0.4$, see Appendix B for all other parameters)	757.6 ±65.0
b)	i)	Divide the horizon by 20 ($h = 100$)	1218.9 ±72.3
	ii)	Divide the horizon by 40 ($h = 50$)	1247.8 ±51.8
c)	i)	Multiply the chunk size by 5 ($c = 25$)	1158.7 ±47.2
	ii)	Multiply the chunk size by 20 ($c = 100$)	1698 ±139
	iii)	Multiply the chunk size by 80 ($c = 400$)	1950 ±83.7
	iv)	Multiply the chunk size by 160 ($c = 800$)	2025 ±99
d)	i)	Decrease availability and arrival rate by 2 ($a = 0.2$, $\lambda = \frac{1}{100}$)	1634 ±105
	ii)	Decrease availability and arrival rate by 4 ($a = 0.1$, $\lambda = \frac{1}{200}$)	3420.32 ±252

3.5 Lesson Learned

There are a number of lessons to be learned from our experiments and CPN model. It is important to note that the modeling of resources is typically done in a naive way. There are issues characterized by parameters such as a, c, and h that dramatically affect performance and these have to be considered in order to make simulations more realistic.

- First of all, it is important not to assume that people are always available and eager to work when cases arrive. In real-life situations, this is not true because people are available for only specific times and may let work accumulate before commencing it. This heavily impacts performance as shown in Figure 9.
- Secondly, when people are available to work, they will do this work in chunks whose size may vary between different people. The bigger the chunk size, the longer the flow times of cases. So, even if the availability is the same, the flow time heavily depends on this parameter and it cannot be ignored as shown in Figure 10.
- Chunks are divided over a particular horizon and so the larger the horizon, the shorter the flow times because of increased flexibility. Increasing the length of the horizon corresponds to making chunks (relatively) smaller.
- Utilization of people is also an important factor that greatly affects the flow times of cases. When it is high, then the flow times increase.

– The example in Section 3.4 shows that these effects may accumulate in larger workflows. The typical assumptions made in today's simulation tools (i.e., $a = 1$, $c \to 0$, and $h \to \infty$), may result in flow times of minutes or hours while with more realistic settings for a, c, and h the flow time may go up to weeks or months and actually coincide with the actual flow times observed.

4 Complementary Approaches and Related Work

Simulation has been used for the analysis of business processes since the seventies [28]. In fact, the simulation language SIMULA was developed in the sixties and influenced the development of general purpose programming languages [11]. Hence, it is fair to say that simulation is one of the earliest and most established applications of computing. While the initial focus was on programming languages extended with simulation capabilities, gradually more and more simulation packages became available that offered some graphical environment to design business processes. These languages provide simulation building blocks that can be composed graphically (e.g. Arena). Today, most business process modeling tools provide some form of simulation (cf. Protos and ARIS). Moreover, the more mature workflow management systems also provide simulation capabilities (cf. FileNet, FLOWer, WebSphere, COSA, etc.). In parallel with the development of simulation tools and embedding of simulation capabilities in larger systems, the analysis of simulation data and the setting up of experiments was investigated in detail [19–21, 24, 28]. In some cases it is possible to use analytical models [9], however, in most cases one needs to resort to simulation.

The use of simulation was also stimulated by management approaches such as Business Process Reengineering [15, 12], Business Process Improvement [16], Business Process Intelligence [14], etc. When reengineering a process from scratch or when improving an existing process design, simulation can be very valuable [7]. Despite the interest in simulation and the potential applicability of simulation, its actual use by end-users is limited.

In Section 2 we mentioned some of the main pitfalls of simulation. The core contribution of this paper is to provide an overview of these problems and to address one particular problem in detail (resource availability).

The results presented complement our earlier work on "short-term simulation", i.e., the analysis of transient behavior using the actual state as a starting point. The idea of doing short-term simulation was raised in [22] using a setting involving Protos (modeling), ExSpect (simulation), and COSA (workflow management). This idea was revisited in [32], but not implemented. Recently, the approach has been implemented using ProM [2], YAWL [1], and CPN Tools [18] (cf. [26]). Processes are modeled and enacted using YAWL and YAWL provides the four types of data mentioned in Figure 2. This information is taken by ProM to create a refined simulation model that includes information about control-flow, data-flow, and resources. Moreover, temporal information is extracted from the log to fit probability distributions. ProM generates a colored Petri net that can be simulated by CPN Tools. Moreover, CPN Tools can load the current state to

allow for transient analysis. Interestingly, both the real behavior and the simulated behavior can be analyzed and visualized using ProM. This means that decision makers view the real process and the simulated processes using the same type of dashboard. This further supports operational decision making [26].

The approach presented in [26] heavily relies on process mining techniques developed in the context of ProM [2]. Of particular importance is the work presented in [25] where simulation models are extracted from event logs. Process mining [5] is a tool to extract non-trivial and useful information from process execution logs. These event logs are the starting point for various discovery and analysis techniques that help to gain insight into certain characteristics of the process. In [25] we use a combination of process mining techniques to discover multiple perspectives (namely, the control-flow, data, performance, and resource perspective) of the process from historical data, and we integrate them into a comprehensive simulation model that can be analyzed using CPN Tools.

When discussing the factors influencing the speed at which people work, we mentioned the Yerkes-Dodson law [31]. Some authors have been trying to operationalize this "law" using mathematical models or simulation models. For example, in [8] both empirical data and simulation are used to explore the relationship between workload and shop performance. Also related is the work presented in [29] where the authors present a different view on business processes, namely describing work as a practice, a collection of psychologically and socially situated collaborative activities of the members of a group. In this view, people are concurrently involved in multiple processes and activities. However, in this work modeling aims at describing collaboration rather than focusing on performance analysis.

Finally, we would like to mention the work reported in [23] where the effectiveness of workflow management technology is analyzed by comparing the process performance before and after introduction of a workflow management system. In this study sixteen business processes from six Dutch organizations were investigated. Interestingly, the processes before and after were analyzed using both empirical data and simulated data. This study showed how difficult it is to calibrate business process simulation models such that they match reality. These and other real-life simulation studies motivated the work reported in this paper.

5 Conclusion

Although simulation is an established way of analyzing processes and one of the oldest applications of computing (cf. SIMULA), the practical relevance of business process simulation is limited. The reason is that it is time-consuming to construct and maintain simulation models and that often the simulation results do not match with reality. Hence, simulation is expensive and cannot be trusted. This paper summarizes the main pitfalls. Moreover, it addresses one particular problem in detail, namely the availability of resources. It is shown that resources are typically modeled in a naive manner and that this highly influences the sim-

ulation results. The fact that people may be involved in multiple processes, and that they tend to work in batches, has dramatic effects on the key performance indicators of a process.

In this paper, we provide a simple model to characterize resource availability. Using this model important insights into the effectiveness of resources are provided. Moreover, it is shown that that these characteristics can be embedded in existing simulation approaches.

Using Figure 2, we discussed the role of different information sources and how information systems and simulation tools can be integrated. This enables new ways of process support. For example, we are working on predictions and recommendations in the context of a PAIS. Using simulation, we can predict when a running case is finished. Based on historical information, we calibrate the model and do transient analysis from the current state loaded from the PAIS. Similarly, we can provide recommendations. For example, by using simulation and historical data, we can predict the execution path that is most likely to lead to a fast result. Initial ideas with respect to prediction and recommendation have been implemented in ProM [2, 30].

References

1. W.M.P. van der Aalst, L. Aldred, M. Dumas, and A.H.M. ter Hofstede. Design and Implementation of the YAWL System. In A. Persson and J. Stirna, editors, *Advanced Information Systems Engineering, Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSE'04)*, volume 3084 of *Lecture Notes in Computer Science*, pages 142–159. Springer-Verlag, Berlin, 2004.
2. W.M.P. van der Aalst, B.F. van Dongen, C.W. Günther, R.S. Mans, A.K. Alves de Medeiros, A. Rozinat, V. Rubin, M. Song, H.M.W. Verbeek, and A.J.M.M. Weijters. ProM 4.0: Comprehensive Support for Real Process Analysis. In J. Kleijn and A. Yakovlev, editors, *Application and Theory of Petri Nets and Other Models of Concurrency (ICATPN 2007)*, volume 4546 of *Lecture Notes in Computer Science*, pages 484–494. Springer-Verlag, Berlin, 2007.
3. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.
4. W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.
5. W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business Process Mining: An Industrial Application. *Information Systems*, 32(5):713–732, 2007.
6. W.M.P. van der Aalst, M. Rosemann, and M. Dumas. Deadline-based Escalation in Process-Aware Information Systems. *Decision Support Systems*, 43(2):492–511, 2007.
7. R. Ardhaldjian and M. Fahner. Using simulation in the business process reengineering effort. *Industrial engineering*, pages 60–61, July 1994.
8. J.W.M. Bertrand and H.P.G. van Ooijen. Workload based order release and productivity: A missing link. *Production Planning and Control*, 13(7):665–678, 2002.
9. J.A. Buzacott. Commonalities in Reengineerd Business Processes: Models and Issues. *Management Science*, 42(5):768–782, 1996.

10. CPN Group, University of Aarhus, Denmark. CPN Tools Home Page. http://wiki.daimi.au.dk/cpntools/.
11. O.J. Dahl and K. Nygaard. SIMULA: An ALGOL Based Simulation Language. *Communications of the ACM*, 1:671–678, Sept 1966.
12. T.H. Davenport. *Process Innovation: Reengineering Work Through Information Technology*. Harvard Business School Press, Boston, 1993.
13. M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede. *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley & Sons, 2005.
14. D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M.C. Shan. Business Process Intelligence. *Computers in Industry*, 53(3):321–343, 2004.
15. M. Hammer and J. Champy. *Reengineering the corporation*. Nicolas Brealey Publishing, London, 1993.
16. J. Harrington. *Business Process Improvement: The Breakthrough Strategy for Total Quality*. McGraw-Hill, 1991.
17. K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. EATCS monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1992.
18. K. Jensen, L.M. Kristensen, and L. Wells. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4):213–254, 2007.
19. J. Kleijnen and W. van Groenendaal. *Simulation: a statistical perspective*. John Wiley and Sons, New York, 1992.
20. A.M. Law and D.W. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, 1982.
21. M. Pidd. *Computer Modelling for Discrete Simulation*. John Wiley and Sons, New York, 1989.
22. H.A. Reijers and W.M.P. van der Aalst. Short-Term Simulation: Bridging the Gap between Operational Control and Strategic Decision Making. In M.H. Hamza, editor, *Proceedings of the IASTED International Conference on Modelling and Simulation*, pages 417–421. IASTED/Acta Press, Anaheim, USA, 1999.
23. H.A. Reijers and W.M.P. van der Aalst. The Effectiveness of Workflow Management Systems: Predictions and Lessons Learned. *International Journal of Information Management*, 25(5):458–472, 2005.
24. S.M. Ross. *A Course in Simulation*. Macmillan, New York, 1990.
25. A. Rozinat, R.S. Mans, M. Song, and W.M.P. van der Aalst. Discovering Colored Petri Nets From Event Logs. *International Journal on Software Tools for Technology Transfer*, 10(1):57–74, 2008.
26. A. Rozinat, M. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and C. Fidge. Workflow Simulation for Operational Decision Support using YAWL and ProM. BPM Center Report BPM-08-04, BPMcenter.org, 2008.
27. N. Russell, W.M.P.van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Workflow Resource Patterns: Identification, Representation and Tool Support. In O. Pastor and J. Falcao e Cunha, editors, *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05)*, volume 3520 of *Lecture Notes in Computer Science*, pages 216–232. Springer-Verlag, Berlin, 2005.
28. R.E. Shannon. *Systems Simulation: The Art and Science*. Prentice-Hall, Englewood Cliffs, 1975.
29. M. Sierhuis and W.J. Clancey. Modeling and Simulating Work Practice: A Method for Work Systems Design. *IEEE Intelligent Systems*, 17(5):32–41, 2002.

30. B. Weber, B.F. van Dongen, M. Pesic, C.W. Günther, and W.M.P. van der Aalst. Supporting Flexible Processes Through Recommendations Based on History. BETA Working Paper Series, WP 212, Eindhoven University of Technology, Eindhoven, 2007.
31. C.D. Wickens. *Engineering Psychology and Human Performance*. Harper, 1992.
32. M.T. Wynn, M. Dumas, C.J. Fidge, A.H.M. ter Hofstede, and W.M.P. van der Aalst. Business Process Simulation for Operational Decision Support. In A. ter Hofstede, B. Benatallah, and H.Y. Paik, editors, *BPM 2007 International Workshops (BPI, BPD, CBP, ProHealth, RefMod, Semantics4ws)*, volume 4928 of *Lecture Notes in Computer Science*, pages 66–77. Springer-Verlag, Berlin, 2008.

A Declarations for CPN Model in Section 3.2

The colset, variable and function declarations of the CPN model have been listed in the ML language.

A.1 Colset Declarations

```
colset CID = int timed;
colset Tm = int;
colset Work= int;
colset Case = product CID * Tm * Work timed;
colset Queue = list Case;
colset Res= string timed;
colset Hor = int;
colset Av = int with 1..100;
colset Chunk = int;
colset Info = product Hor * Av * Chunk;
colset RWC = product Res * Work * Chunk timed;
colset RT = product Res * Tm timed;
colset RI = product Res * Info timed;
colset CR = product Case * RT timed;
```

A.2 Variable Declarations

```
var i:CID;
var t,t1,t2,done:Tm;
var w,w1,w2:Work;
var r:Res;
var h:Hor;
var a:Av;
var c,c1:Chunk;
var q:Queue;
var hac : Info;
val Rinit = [("r1",(1000,20,200))];
```

A.3 Function Declarations

```
fun x1([]) =[] | x1((x,(h,a,c))::r) = (x,0,c)::x1(r);
fun x2([]) =[] | x2((x,y)::r) = x::x2(r);
fun Mtime() = IntInf.toInt(time()):int;
fun Dur()= floor(exponential(1.0/15.0));
fun IAT()= floor(exponential(1.0/100.0));
fun min(x,y) = if x<y then x else y;
```

B Task Parameters for Base Scenario Described in Section 3.4

	Task	**Parameters**
a)	Register	Resources $r_a = 1$ Arrival rate $\lambda_a = \frac{1}{50}$ Service rate $\mu_a = \frac{1}{18}$ Utilization $\rho_a = 0.36$
b)	Classify	Resources $r_b = 2$ Arrival rate $\lambda_b = \frac{1}{50}$ Service rate $\mu_b = \frac{1}{36}$ Utilization $\rho_b = 0.36$
c)	Phone Garage	Resources $r_c = 3$ Arrival rate $\lambda_c = \frac{1}{100}$ Service rate $\mu_c = \frac{1}{100}$ Utilization $\rho_c = 0.33$
d)	Check Insurance	Resources $r_d = 2$ Arrival rate $\lambda_d = \frac{1}{100}$ Service rate $\mu_d = \frac{1}{70}$ Utilization $\rho_d = 0.35$
e)	Decide	Resources $r_e = 2$ Arrival rate $\lambda_e = \frac{1}{100}$ Service rate $\mu_e = \frac{1}{70}$ Utilization $\rho_e = 0.35$
f)	Pay	Resources $r_f = 1$ Arrival rate $\lambda_f = \frac{1}{200}$ Service rate $\mu_f = \frac{1}{70}$ Utilization $\rho_f = 0.35$
g)	Send Letter	Resources $r_g = 2$ Arrival rate $\lambda_g = \frac{1}{50}$ Service rate $\mu_g = \frac{1}{36}$ Utilization $\rho_g = 0.36$