

Modeling and Analyzing Resource-Constrained Business Processes

Cesar Oliveira, Ricardo Lima
Center of Informatics
Federal University of Pernambuco
Recife, Brazil
{calo,rmfl}@cin.ufpe.br

Thiago Andre
Politechnique School of Pernambuco
University of Pernambuco
Recife, Brazil
tma@dsc.upe.br

Hajo A. Reijers
Information Systems
Eindhoven University of Technology
Eindhoven, The Netherlands
h.a.reijers@tue.nl

Abstract—In this paper we review several approaches for the performance evaluation of business workflows and propose an approach based on Generalized Stochastic Petri Nets, which presents desirable characteristics, such as analytic solutions. We present a case study demonstrating the application of our proposal and compare the results with that obtained by a colored Petri net model.

Index Terms—business process, performance evaluation, resource constraints, generalised stochastic petri nets

I. INTRODUCTION

Modeling Discrete-Event Dynamic Systems (DEDS) and predicting its performance are challenging tasks. For instance, traditional *queueing network* modeling framework can fail to capture precedence constraints or complex synchronization behavior. Moreover, models including detailed aspects of the system cannot be analyzed using methods based on the theory of continuous-time Markov chains on a finite or countably infinite state space.

In this context, Generalized Stochastic Petri Nets (GSPN) have proven to be a popular and useful tool for modeling and performance analysis of complex stochastic DEDS. The system under study is modeled using exponentially distributed firing delay with timed transitions, which occurs only at an increasing sequence of random time. Thus, through GSPN is relatively easy to capture precedence constraints, synchronization behavior, and random execution of preemptive tasks in a DEDS with limited resources.

Business Process Management (BPM) is the achievement of an organization's objective through the improvement, management and control of essential business process [9]. BPM explores techniques and workflow tools to design, enact and analyze operational processes. An operational processes designed and controlled through a workflow is a kind of DEDS. While workflow management aims at automating the business process, BPM emphasizes the analysis and optimization of the workflow performance. Usually, the evaluation of three metrics are essential in business processes: time constraints; time performance; and efficiency of resource utilization.

In this paper, we apply GSPN to analyze the time performance of business processes. We define a set of building blocks (GSPN models) to represent the elements of a workflow. The proposed models supports the modeling and performance

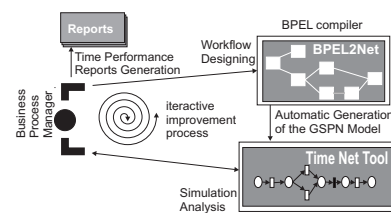


Fig. 1. General view of the modeling and analysis method

analysis of business process with multiple customers and limited resources. A computational tool, called *BPEL2Net*, was also developed. It aims to support the design of workflow and its posterior translation into a GSPN model composed of the building blocks described in this work. The generated GSPN model can then be used to qualitative analysis or performance evaluation.

We present closed-form formulae for computing the throughput at each point of the workflow. Furthermore, we define a formula to calculate the minimum number of resources required in each organizational role. Although these metrics can be calculated analytically, GSPN simulation is also employed in order to obtain other interesting metrics such as queueing time. We employ Time Net tool [6] to analyze and simulate GSPN models generated by the BPEL2Net tool. Figure 1 shows a general view of the modeling and analysis methodology proposed in this paper.

An application example demonstrates the use of the method for performance evaluation proposed in this paper. Results are compared against those obtained through an approach based on Colored Petri Nets (CPN).

II. LITERATURE REVIEW

In this section, we present existing approaches for the performance evaluation of business workflows. We classified these approaches according to three criteria: workflow scenario; nature of the results; modeling power.

The criterion **workflow scenario** evaluates if the modeling approach represents relevant elements of the real workflow environment. In particular, we classified the approaches according to the following parameters 1) *number of customers*; 2) *number of resources*, and; 3) *number of process definitions*.

The criterion **nature of results** evaluates the characteristics of results obtained through the modeling approach. We adopt two parameters with this propose: 1) *type*: the format employed to describe the features of the data gathered from an experimental study (simple average, probability distribution, etc); *metrics*: list of metrics that are directly obtainable from the model.

The criterion **modeling power** is useful to understand both how accurately the model can express system's characteristics and how much effort is required to design the model. Here, six parameters are explored: 1) *time representation*: determines if the approach deals with discrete or continuous time; 2) *time variability*: captures if the approach allow for the direct representation of time with different probability distributions - if it allows, it is marked as *yes*; otherwise, it is marked as *"no"*; the cases where the variability can be achieved with extra effort are marked as *"maybe"*; 3) *readability*: indicates how easy is to understand, read and maintain the model; 4) *effort*: evaluates the abstraction level of the modeling language as well as the effort required to calculate desired metrics - we assume three levels of modeling effort: *high*, *medium*, and *low*; 5) *tool*: indicates if the approach is supported by a computational tool (we take in consideration only tools created for the approach itself); 6) *scalability*: indicates how the approach scale with the size of the system.

A. Overview of Related Approaches

Rud et al. [15] propose a model based on operational research techniques to estimate the performance of BPEL processes and the workload of web services. Statistical information is collected through an on-line system that monitors network and service operations. Their model supports multiple customers and multiple processes concurring for limited resources (server capacity). Mean value is used in equations developed to compute response time and to estimate resource utilization.

Reijers [14] proposes a Petri net-based model, called Stochastic Workflow Net (SWN), which is able to numerically compute the distribution of workflow execution time. The system processes a single customer in this model and resources are unlimited. The model evaluation mechanism takes into account a single process. The time representation is discrete, which allows for an easier computation of time distributions. Independently, Hao [7] presents a model that has the same characteristics proposed earlier by Reijers, without presenting relevant differences or advantages.

Van der Aalst et al. [2][1] show the application of queueing theory for performance evaluation of Workflow Net (WF-Net) models. WF-Nets are a widely known Petri net representation of workflow used for qualitative analysis. However, queueing theory is not useful for performance evaluation of complex workflows because it does not support parallelism and synchronization. In these cases, WF-Nets can be analyzed through the use of colored Petri nets. Token colors represent different customer orders and simulation is employed to retrieve approximated performance measures with certain confidence level.

Ferscha [5] was, as far as we know, the first to propose the use of GSPN for performance evaluation of business processes. His model represents a set of agents concurring for resources in order to execute the processes that they are responsible for. The interactions and dependencies between processes are taken in consideration (e.g., producer-consumer relations). There is not a clear notion of customer in the model proposed by Ferscha, as the agents are working continuously independent of any customer demand. Also, it is not expressed how a single agent executes parallel activities, which makes the model confusing when trying to compare to today's workflow concepts and practices. The only metric mentioned in this work is the throughput of the system.

Schomig & Rau [16] present a colored GSPN for performance evaluation of workflow that is aligned with concepts recognized by the Workflow Management Coalition (WfMC). They argue that token differentiation is important because decisions made at one point of the workflow can affect decisions that will be made at another point in the future. Four basic branch structures are modeled: AND-Split (fork), AND-Join (synchronization), OR-Split (exclusive decision) and OR-Join (path merging). Similar to Ferscha's model and Reijer's SWN, this approach does not take into account customer demand. Process is executing continuously and a single customer is being served in each run. Resource constraints are considered, but once a single customer is being served, these constraints affect only the execution of parallel activities. They also show that state-space explosion seriously limits the application of this technique for practical applications today.

Shuxia Li & Zhu [11] also present a GSPN model for the analysis of workflow performance, which they call Generalized Stochastic Workflow Net (GSWN). The model presents single customer and infinite resources. They argue that it is reasonable to think in infinite resources, since human resources can deal with several tasks in parallel. They consider the same four routing structures as Schomig & Rau. They also recognize that state-space explosion impairs the application of the approach for complex workflow models.

JianQiang Li et al. [10] present an hybrid approach, called Multi-dimension Workflow Net (MWF-net). Their work represent a set of independent processes that are executed by a shared set of resources. Each process is represented by a time-extended WF-Net. These processes are linked together by mapping them to a common set of organizational roles. Each timed transition is associated to a role in the organizational structure. In a third layer, these roles are mapped to resource pools, which represent the workforce available in each role. By applying decomposition and combination algorithms, the authors show how to obtain information about resource utilization and a lower bound for the process performance. These algorithms employ both Petri net analysis and complementary analytic formulae based on queueing theory. Multiple customers are considered to arrive independently at each workflow.

Simulation of workflow is a common practice in industry. Many industrial workflow systems provide simulation features. These applications employ different Discrete Event Simulation

(DES) algorithms. Due to this distinction, results obtained from one tool can significantly differ from another. Scientific works that employ simulation of workflow mostly use Petri net based simulations, due to its formal semantics.

As previously mentioned, van der Aalst et al. extend their WF-Net models with color in order to obtain a colored Petri net model that can be used for performance evaluation [2].

Reijers presents a Resource-Extended SWN [14], which adds resource constraints to the original SWN model and employs colored tokens for representing multiple different customers in the system. However, the algorithms adopted in the SWN model are not valid for the resource-extended version. Results in this new model are obtained by simulating the colored Petri net.

Netjes et al. [12] provide a model for evaluating resource allocation alternatives for optimizing workflow performance. Again, colored Petri nets are employed and results are obtained by simulation.

Dehnert et al. [4] present a model that employs colored GSPN to evaluate workflow performance. The model is divided into two parts: the resources model and the workflow model. The former represents every communication and document transport between the departments and employees. It also represents employee vacancies or holidays. The workflow model represents the activities and dependencies between activities. These two models are merged for analysis purpose. Multiple processes can be evaluated in the same model, sharing the resources. Customer demand is not represented. Resource utilization and execution time can be estimated from this model through analytical solution based on the state-space generation or by simulation.

Table I summarizes the characteristics of the works mentioned above.

III. THE CHOICE FOR GSPN

Many works exist on performance evaluation of workflow using Colored Petri Nets (CPN). In this section we explain the reasons of our choice for a GSPN based approach. We highlight key advantages of GSPN over CPN for the purpose of performance evaluation, considering CPN Tools' [8] implementation as reference.

There are some characteristics found in CPN models that are undesirable when dealing with timed models:

- 1) time is represented as integer. Once the time is tracked by a global integer variable, time calculated by the model must be rounded to integer values, resulting in a lost of precision that can cause undesirable effects in a complex model;
- 2) it is difficult to investigate the model analytically with the objective of finding mathematical relations between the elements and metrics.

Time is not a natural concept in CPN, its progression must be manipulated by the modeler. Therefore, the semantics of this progression is of responsibility of the user. Moreover, the necessity of rounding timestamps to integer is an additional concern. For example, rounding an exponentially distributed

TABLE I
COMPARATIVE STUDY: CRITERIA WORKFLOW SCENARIO, NATURE OF RESULTS, AND MODELING POWER

Work	Workflow Scenario			Nature of Results		
	Cust.	Resourc.	Proc.Def.	Type	Metrics	
Rud [15]	mult.	limit.	mult.	average	time, utiliz.	
Reijers [14] (SWN)	single	unlim.	single	distrib.	time	
Ferscha [5]	unclear	limit.	mult.	average	throughput	
Schomig [16]	single	limit.	single	average	time	
Shuxia [11]	single	unlim.	single	average	time	
JianQiang [10]	mult.	limit.	mult.	lower bound	time, utiliz.	
Reijers [14] (RESWN)	mult.	limit.	mult.	conf.interv.	time, queues, utiliz.	
vdAalst [2] (CPN)	mult.	limit.	mult.	conf.interv.	time, queues, utiliz.	
Netjes [12]	mult.	limit.	mult.	conf.interv.	time, queues, utiliz.	
Denhart [4]	single	limit.	mult.	average	time, queues, utiliz.	
<i>Our</i>	mult.	limit.	mult.	average, conf.interv.	time, queues, utiliz.	
Modeling Power						
Work	Time Repres.	Time Variab.	Readab.	Effort	Scalab.	Tool Support
Rud [15]	cont.	no	high	low	high	no
Reijers [14] (SWN)	discrete	yes	high	med.	med.	no
Ferscha [5]	cont.	maybe	high	low	low	no
Schomig [16]	cont.	maybe	high	low	low	no
Shuxia [11]	cont.	maybe	high	low	low	no
JianQiang [10]	cont.	maybe	low	high	low	no
Reijers [14] (RESWN)	discrete	yes	low	med.	high	no
vdAalst [2] (CPN)	discrete	yes	low	med.	high	modeling
Netjes [12]	discrete	yes	low	med.	high	no
Dehnert [4]	cont.	maybe	med.	med.	high	no
<i>Our</i>	cont.	maybe	high	low	high	yes

random variable to the next integer will lead to a geometric distribution instead of an exponential. The modeler must be sure that this does not affect the behavior of his/her model. Even when the modeler takes care about this, a long simulation can be influenced by numerical errors due to this rounding and the final results can be seriously affected.

The second point observed comes from the fact that it is difficult to define an stochastic process corresponding to the CPN model. So, if one intends to study properties of the system in addition to simulating it, the CPN will give little support for that task. For example, one could be interested in the impact of changing the arrival distribution to the overall response time. In a CPN model, the only option the modeler have is to simulate the model with several distributions and measure this impact. On the other hand, a stochastic model could be studied in order to find a mathematical relation between the arrival distribution and the response time, if such a relation exists.

Nevertheless, CPN has some good characteristics for performance evaluation. It provides more freedom to express time behavior. It allows for the creation of arbitrary discrete time distributions. Moreover, the model can be rapidly adapted to fit other time distributions.

Generalized stochastic Petri nets, on the other hand, is a formalism designed to represent stochastic systems. They are isomorphic to Continuous Time Markov Chains (CTMC), an

analytical model used for stochastic study for decades. Time is a natural concept in GSPN models, associated to timed transitions. Therefore, no manipulation of timing variables is necessary. Moreover, the CTMC associated to a GSPN can be clearly defined, which makes it possible to study its properties over a formal basis.

The drawback is that GSPN transitions can only be associated to exponentially distributed times. Methods for approximating other distributions do exist [3] and are widely used. These methods require the addition of auxiliary structures to model the desired distribution.

IV. MODELS DESCRIPTION

In this section we summarize our proposal for a GSPN representation of business workflows. The complete formalization is published in a technical report [13].

A business workflow is basically comprised of atomic activities, order relations between these activities and the definition of roles that are responsible for executing them. At runtime, we need to consider also resources that are assigned to these roles and process instances that are in execution.

For representing the process definition, we provide basic models for activities and roles, and composition rules that can be applied for combining several activities, making it possible to model complex workflows. For representing the runtime environment of the workflow, we define a *workflow system* model, which adds customer and resource information to the model in order to allow performance evaluation.

A. Basic Models

We model each role i by a place R_i , which holds tokens representing resources assigned to that role. An activity is modeled by two places and two transitions. Place W represents the worklist, holding one token for each item that demands processing. Place S contains tokens that are being processed by a resource. Transition q is enabled as soon as an item is put in the worklist and a resource is available for handling it. Transition $T_{ACTIVITY}$ is an exponential transition with mean delay set to the activity's expected execution time. This model can be seen in Fig. 2, surrounded by a dashed square.

The *workflow system* model is created by adding customer arrival and departure structures and putting resources in role places. Fig. 2 displays a workflow system with a single activity (inside dashed square). The customer arrival is modeled by the $T_{ARRIVAL}$ transition. This transition produces tokens according to a Poisson process with rate λ . A delay $d_\lambda = 1/\lambda$ is assigned to it in order to obtain the desired arrival rate.

B. Composition Operations

The composition operations are uniformly defined such that every composed structure contains a single *starting place* and a set of *departing transitions*. Every token that arrives at that *starting place* must eventually depart through one of the *departing transitions*.

We call such structures *sub-processes*. An activity model is the most simple sub-process structure. Every composition

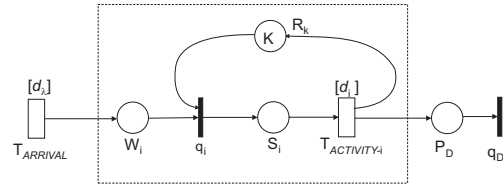


Fig. 2. Workflow System with a single activity

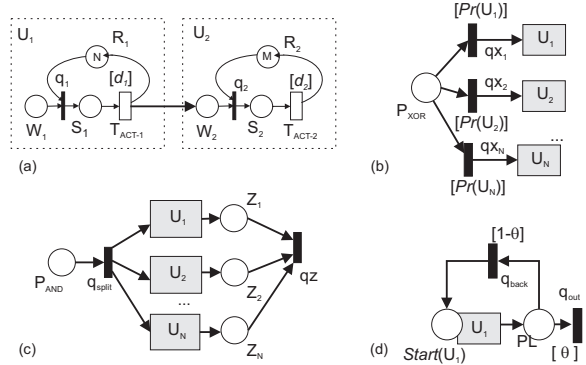


Fig. 3. Composition operations: a) SEQ; b) XOR; c) AND; d) LOOP

operation is defined as a function that maps one or more sub-process operands to a resulting sub-process.

The composition operations are:

- Sequence (SEQ) - two or more sub-processes are executed in sequence;
- Alternative Path (XOR) - a decision is made about one from a set of sub-processes that can be executed;
- Parallelism (AND) - a set of sub-processes are executed in parallel and synchronized at the end;
- Simple Iteration (LOOP) - one sub-process is executed several times;
- Grid-form Iteration (GRID-LOOP) - a set of sub-processes are executed several times, but there is an exit point after each sub-process that lets the iteration finish after the execution of that sub-process, without completing the whole cycle;
- Multiple Path (OR) - there are two sub-processes that can be executed in a non-exclusive way. If both are executed, they must be synchronized at the end of the structure;
- Interleaving (INTER) - a set of sub-processes can be executed in any order, but two sub-processes from this set can not be executed in parallel for the same process instance.

The structures for these composition operations are represented in Figures 3 and 4. The sub-processes are represented by gray-filled rectangles and denoted by U . Each arc entering a sub-process is considered to be connected to its *starting place*. Each arc going out from the sub-process is considered to be connected to *all* of its *departing transitions*. This is illustrated in the SEQ operator (Fig. 3.a), where two activities are being connected.

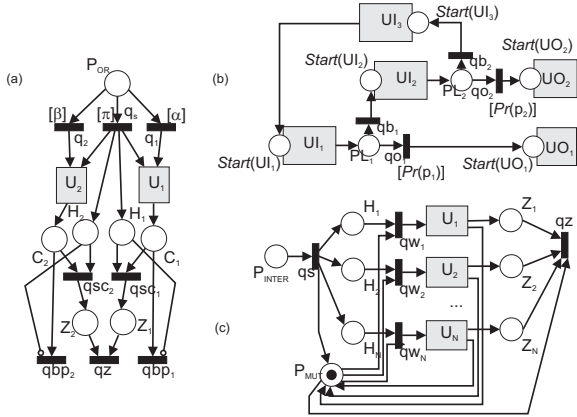


Fig. 4. Other composition operations: a) OR; b) GRID-LOOP; c) INTER

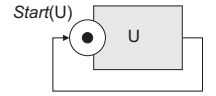


Fig. 5. Soundness model

C. Metrics and Analysis

The proposed model can be used for both qualitative (correctness) and quantitative (performance) analysis. The correctness of a sub-process is analyzed by using the *soundness model* shown in Fig. 5. This model allows the verification of the *Soundness* property, as stated by van der Aalst. All role places receive resource tokens according to the scenario under study. If the model is *live* and *bound*, then it is *Sound*.

Once the sub-process is correct, a workflow system can be modeled, by adding the customer arrival and departure structures. Hence, several performance metrics can be retrieved.

For each role R_K in the workflow that is responsible for activities A_1, \dots, A_N , we can compute the minimum number K of resources necessary for this role in order to avoid infinite queues in the system as:

$$K > \sum_{i=1}^N \lambda_i d_i .$$

Once the minimum number of resources is provided, the system reaches an equilibrium state, from where we can retrieve stationary metrics. Table II summarizes the metrics for the basic activity model. These metrics also allow for the computation of the expected number of available resources, given by $E(R) = K - \sum_{i=1}^N E(S_i)$.

An important metric that must be calculated for each activity or sub-process is the local arrival rate, i.e., the customer

TABLE II
METRICS FOR THE BASIC MODELS

Metric	Expression
Expected Number of Activity Instances	$E(S) = \lambda d$
Expected Number of Work Items	$E(W) = \text{expectation of place } W$
Expected Number of Cases	$E(n) = E(W) + E(S)$
Mean Response Time	$E(\tau) = E(n)/\lambda$

arrival rate at that specific point in the workflow. These rates can be obtained by the formulae below, on the basis of the composition operations applied.

Let U_R be a sub-process composed by a set of minor sub-processes U_i and λ be the customer arrival rate at the beginning of U_R , the local arrival rate λ_i at each sub-process U_i can be computed as follows.

- Sequence $U_R = SEQ(U_1, U_2) : \lambda_1 = \lambda_2 = \lambda$.
- Alternative Path $U_R = XOR(U_1, \dots, U_N, Pr)$, where $Pr(U)$ is the probability of choosing the path $U : \lambda_i = \lambda Pr(U_i)$, $i = 1, \dots, N$.
- Parallelism $U_R = AND(U_1, \dots, U_N) : \lambda_i = \lambda$, $i = 1, \dots, N$.
- Simple Iteration $U_R = LOOP(U_1, \theta)$, where θ is the probability of leaving the iteration: $\lambda_1 = \frac{\lambda}{\theta}$.
- Grid-form Iteration $U_R = GRID - LOOP(\{U_{I_1}, \dots, U_{I_{k+1}}\}, \{U_{O_1}, \dots, U_{O_k}\}, Pr)$, where each U_{I_i} is a sub-process in the iteration cycle, each U_{O_j} is a sub-process executed after exiting from the exit point p_j and Pr maps a probability to each exit point to be taken

$$\lambda^I_1 = \frac{\lambda}{\sum_{j=1}^k Pr(p_j)} ; \lambda^I_i = \lambda^I_1 \prod_{v=1}^{i-1} (1 - Pr(p_v)) ;$$

$$\lambda^O_i = \lambda^I_1 \prod_{v=1}^i Pr(p_v) .$$

- Multiple Path $U_R = OR(U_1, U_2, \alpha, \pi, \beta)$, where α is the probability of only U_1 be chosen, β is the probability of only U_2 be chosen and π is the probability of both be chosen ($\alpha + \beta + \pi = 1$): $\lambda_1 = (\alpha + \pi)\lambda$; $\lambda_2 = (\beta + \pi)\lambda$.
- Interleaving $U_R = INTER(U_1, U_2, \dots, U_N) : \lambda_i = \lambda$, $i = 1, \dots, N$.

After computing the arrival rates, every metric from the activity model can be analytically obtained from the formulae presented in Table II, except by the worklist sizes. Also, the time spent at synchronization points can not be obtained from formulae. For obtaining these metrics, the GSPN must be evaluated. Notice that the *workflow system* is unbounded. Therefore, the GSPN must be evaluated through simulation. After obtaining the expected markings of the places of interest, the complete set of metrics can be computed.

V. CASE STUDY

This section presents an application of the GSPN model to evaluate a real Workflow process. This Workflow consists in a *Modification Request* process executed in a software house. We used the process in a hypothetical scenario, where the enterprise is applying efforts to increase the productivity of its programming team and reduce costs with salary. In the initial scenario, there are five general purpose programmers. The organization is planning to select some of them and provide training such that they become specialized in one of two areas: 1) *interface development* (front-end engineer); or 2) *database development* (database engineer). Their salaries will also be updated according to their new role. Notice that a generalist programmer, if present, can perform the same tasks of the specialists, although with less efficiency and quality. The company wants to figure out how many specialists in each area is required in order to achieve the desired objective.

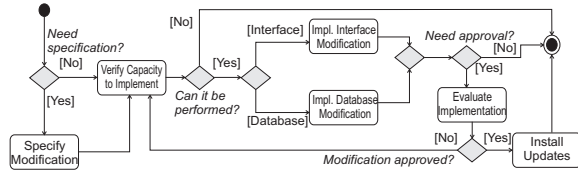


Fig. 6. Workflow of the software development process

TABLE III
ACTIVITIES AND ROLES

Activity	Label	Role
Specify Modification	A_{SM}	Stakeholder (SH)
Verify Capacity to Implement	A_{VC}	Analyst (AN)
Implement Interface Modification	A_{IIM}	Front-end Engineer (ID)
Implement Database Modification	A_{IDM}	Database Engineer (DB)
Evaluate Implementation	A_{EI}	Stakeholder (SH)
Install Updates	A_{IU}	Any Programmer (GB)

Fig. 6 presents the Workflow description of the software development process. Activities that were automated in the original process, i.e. executed by the system, were removed from the model, leaving only the human-performed activities.

Table III presents the labels assigned to each Activity along with the roles responsible for each one of them. The generalist programmer can execute any of the implementation Activities.

Tables IV and V present the (hypothetical) statistical data about current process performance. Table IV presents the mean execution time for each Activity, which were assigned to GSPN transitions (as so as time between case arrivals). Notice that the implementation Activities have different mean values depending on whether the resource is a specialist programmer or the generalist programmer. Table V shows the probabilities of choices in each decision point. Since there are only two alternatives in every branch, only the probability of one alternative is shown. The probability of the other alternative is just the complement of that value.

For the cost analysis, we assigned hypothetical salaries to each type of programmer: \$ 15 per-hour for the front-end engineer; \$ 20 per-hour for database engineer; and \$ 10 per-hour for generalist programmer. Although generalist programmers are able to perform the two Activities, they have no expertise and provide an inferior QoS.

The *BPEL2Net* tool uses the basic models and composition operations proposed in this work to generate the corresponding GSPN model. The TimeNet tool [6] was employed for evaluating the model. The model was simulated with confidence level of 99%. We simulated seven different possible configurations

TABLE IV
MEAN DELAYS ASSIGNED TO TRANSITIONS (IN HOURS)

Activity	Transition	Delay
Case Arrival	T_A	15
Specify Modification	T_{SM}	6
Verify Capacity to Implement	T_{VCI}	4
Interface Modification (spec.)	T_{IIM-S}	40
Interface Modification (gen.)	T_{IIM-G}	60
Database Modification (spec.)	T_{IDM-S}	35
Database Modification (gen.)	T_{IDM-G}	55
Evaluate Implementation	T_{EI}	2
Install Updates	$T_{IU-I}, T_{IU-D}, T_{IU-G}$	4

TABLE V
PROBABILITIES IN THE ALTERNATIVE STRUCTURES

Decision	Decided to Yes	Decision	Decided to Yes
Need specification?	80%	Need approval?	80%
Can it be performed?	85%	Modif. approved?	70%
Interface change?	60%		

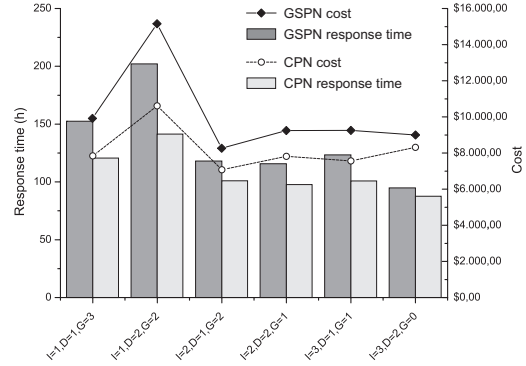


Fig. 7. Response times and costs

of specialist/generalist employment. Table VI presents the numerical results.

Results showed that the configuration with 3 front-end engineers, 2 database engineers, and no generalist programmer provided the best response time. This configuration lead to an average response time around 94 hours, or 12 days, with cost of \$ 8,959.98 dollars, for each modification request. However, taking into account the costs, the less expensive configuration was obtained with 2 front-end engineers, 1 database engineers, and 2 generalist programmers. This configuration lead to an average response time of 117 hour, or 15 days, with a cost of \$ 8,224.59 dollars for each modification request. The cost for each modification request is obtained through the following equation: $E(\tau) \times \sum C_i$, where C_i is the cost-per-hour of resource i , and $E(\tau)$ is the average time required for the modification to be performed. Figure 7 presents a plot of these results along with the results obtained from a CPN model, described in next subsection. We omitted the worst configuration from the graph.

The worst performance was the one in which only generalists are employed. In this case, the response time was of 1,447 hours, with a cost of \$ 72,353.11 dollars per modification. This shows that an investment on specialized workers for this organization can provide excellent returns.

TABLE VI
MEASURES OBTAINED FROM THE CASE STUDY

Configuration	$\sum E(W_i)$	$\sum E(S_i)$	$E(\tau)$	Cost
1 ID; 1 DB; 3 GD	5.133373	4.993536	152 h	\$ 9,873
1 ID; 2 DB; 2 GD	8.733033	4.697900	201 h	\$ 15,109
2 ID; 1 DB; 2 GD	3.192793	4.640148	117 h	\$ 8,224
2 ID; 2 DB; 1 GD	3.291249	4.382592	115 h	\$ 9,208
3 ID; 1 DB; 1 GD	3.790802	4.396272	123 h	\$ 9,210
3 ID; 2 DB; 0 GD	2.257113	4.030590	94 h	\$ 8,959
0 ID; 0 DB; 5 GD	90.683658	5.787160	1447 h	\$ 72,353

A. Comparison to the CPN approach

We applied colored Petri nets to model and evaluate the same Workflow. The key difference between the two models is that, in the CPN, the decisions taken in every alternative structure is based on Case Data instead of being stochastic. However, we generated Cases with data that fit in the statistics presented in Table V. The decision for exiting from the loop is randomized in every iteration, also according to the probabilities defined in Table V.

CPN Tools was employed to simulate the model with confidence level of 99%. As can be seen in Fig.7, the average results were very similar. Taking into account the response time, the best configuration was the same as that obtained with the GSPN model. Moreover, results for the cost analysis were also similar to that calculated through our GSPN model. The values obtained from the CPN presented a difference of approximately 18% in average from the values calculated by the GSPN. Further experiments should be conducted in order to find the cause of this difference. Errors due to CPN's necessity of rounding time may have influenced. Nevertheless, the results were coherent to that obtained through the GSPN models.

VI. CONCLUSION

With the purpose of evaluating the performance of business processes, we proposed a set of GSPN models to represent key components of workflow languages. Our approach can represent complex workflow systems through the composition of the basic models.

The suggestion of criteria for comparing methods developed to analyze the performance of business processes is an additional contribution of this work. Such criteria were employed to compare eleven different works, including our approach. CPN based approaches demonstrated to cover most of the good characteristics prescribed by the criteria adopted. However, CPN models represent time as integer. This potentially causes a loss of precision due to necessity of rounding timestamp values to integer. GSPN deals with continuous time directly, providing more accurate results. Eventually, observing the set of criteria proposed, our approach achieved at least the same level of quality as that found on higher quality CPN based approaches. This is not the case of other works which also employ the GSPN formalism. Therefore, the analysis revealed that our work includes more good characteristics than those observed in other works which adopt GSPN for modeling and analyzing the performance of business processes. Moreover, the choice for GSPN provides more accurate results.

A case study was conducted to validate our approach. The results were very satisfactory, since they were similar to that obtained through a well known CPN based approach.

As a final contribution, we developed a tool, called *BPEL2Net*, for translating BPEL workflow descriptions into GSPN, using the basic models and composition rules proposed in this paper. The tool and the case study presented in this paper can be accessed at www.cin.ufpe.br/~calo/bpel2net.

As a future work, we will implement a framework with several plugins to support the design and performance analy-

sis business processes using our proposal. We are currently working on a plugin to support the graphical modeling of business process workflows. We plan to develop plugins for communicating with the TimeNet tool from our graphical interface and importing workflows modeled using different tools and languages (BPMN, BPEL, UML, etc).

VII. ACKNOWLEDGMENTS

This work was partially supported by FACEPE, FINEP, and Cenpes/Petrobras.

REFERENCES

- [1] W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
- [2] W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.
- [3] Steven C. Bruell, Pozung Chen, and Gianfranco Balbo. Alternative methods for incorporating non-exponential distributions into stochastic timed petri nets. In *PNPM*, pages 187–197. IEEE Computer Society, 1989.
- [4] Juliane Dehnert, Jorn Freiheit, and Armin Zimmermann. Modeling and performance evaluation of workflow systems, 2000.
- [5] A. Ferscha. Qualitative and quantitative analysis of business workflows using generalized stochastic petri nets, 1994.
- [6] Reinhard German, Christian Kelling, Armin Zimmermann, and Günter Hommel. TimeNET: A toolkit for evaluating non-markovian stochastic petri nets. *Perform. Eval.*, 24(1-2):69–87, 1995.
- [7] Jiang Hao and Wang Pei-an. An approach for workflow performance evaluation based on discrete stochastic petri net. In *ICEBE '07: Proceedings of the IEEE International Conference on e-Business Engineering*, pages 327–330, Washington, DC, USA, 2007. IEEE Computer Society.
- [8] Kurt Jensen. An introduction to the practical use of coloured petri. In *Lectures on Petri Nets II: Applications*, 1998.
- [9] John Jeston and Johan Nelis. *Business Process Management : Practical Guidelines to Successful Implementations*. Elsevier/Butterworth-Heinemann, Amsterdam, 2006.
- [10] JianQiang Li, Yushun Fan, and MengChu Zhou. Performance modeling and analysis of workflow. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 34(2):229–242, 2004.
- [11] Shuxia Li and Haiping Zhu. Generalized stochastic workflow net-based quantitative analysis of business process performance. In *ICIA '08: Proceedings of the IEEE International Conference on Information and Automation*, pages 1040–1044, Washington, DC, USA, 2008. IEEE Computer Society.
- [12] M. Netjes, W.M.P. van der Aalst, and H. A. Reijers. Analysis of resource-constrained processes with colored petri nets. In K. Jensen, editor, *Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2005)*, volume 576 of *DAIMI*, pages 251–266, Aarhus, Denmark, October 2005. University of Aarhus.
- [13] Cesar Oliveira and Ricardo Lima. Performance analysis of resource-constrained business processes: A formal approach based on stochastic petri nets. Technical report, Federal University of Pernambuco, 2009.
- [14] H.A. Reijers. *Design and Control of Workflow Processes: Business Process Management for the Service Industry*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2002.
- [15] Dmytro Rud, Andreas Schmietendorf, and Reiner Dumke. Performance modeling of ws-bpel-based web service compositions. In *SCW '06: Proceedings of the IEEE Services Computing Workshops*, pages 140–147, Washington, DC, USA, 2006. IEEE Computer Society.
- [16] A. K. Schomig and H. Rau. A petri net approach for the performance analysis of business processes. Technical report, University of Würzburg, 1995.