Performance Evaluation of Web Services Orchestrated with WS-BPEL4People

Henrique Jorge A. Holanda[1,2], José Merseguer Hernáiz[2], Giovanni Cordeiro Barroso[3], A. B. Serra[4]

[1] Departamento de Informática UERN, Mossoró, Brazil
henriqueholanda@uern.br

[2] Dpto. de Inf. e Ing. de Sistemas, UNIZAR, Zaragoza, Spain
jmerse@unizar.es

[3]Universidade Federal do Ceará
Campus do Pici, Fortaleza,- CE 60455 760 – Brasil
gcb@ufc.br

[4] Instituto Federal de Educação Tecnológica do Ceará
Av. 13 de Maio, 2081, Fátima, Fortaleza – CE 60040-531 – Brasil
prof.serra@gmail.com

Abstract

WS-BPEL Extension for People (BPEL4People) introduces human activity to Web Services Business Process Execution Language (WS-BPEL/BPEL). Some works have been done on the performance verification of WS-BPEL, but there aren't works on the performance verification of WS-BPEL4People. In this paper, we propose a model method for the transformation of WS-BPEL4People/BPEL4People into Generalized Stochastic Petri Nets (GSPN) and this way evaluate its performance. During the transformation, Petri Net is used to model BPEL activities and the human activity of a human task. Then, by validating the generated model, some potential problems with performance of WS-BPEL4People can be detected.

Keywords

BPEL4People, Petri Net, Performance, Web Services, SOA

1. Introduction

Living in a competitive world, businesses are naturally interested in information technology supporting them for competitive advantage [1]. As cooperation becomes increasingly important for companies, new challenges arise for the support of business to business scenarios by information technology. While enterprises already profit from the use of traditional workflow management systems (WfMS) , the business process execution language (BPEL) permits formal specification of processes and enables companies to collaborate with each other by interacting business processes. BPEL can be used for automated processes between businesses using respective services. However, the obvious scenario of a business process that depends on a person to fulfil a certain human task as a kind of process activity, is not covered by BPEL [5].

BPEL (Business Process Execution Language) has become the standard for specifying and executing workflow specifications for Web Service (WS) composition invocation. A major weakness of BPEL is the lack of so-called “human workflow” support. The BPEL4People specification tries to amend this by adding human task support to BPEL.

DOI : 10.5121/ijcnc.2010.2609

A weakness of BPEL is the lack of “human workflow” support. In long-running business processes, tasks that require human involvement exist widely. A process may wait for input from human participants or WS, and the input must be collected within a certain number of hours. When a timeout occurs, a user must be notified to decide how the process proceeds. As quoted from the whitepaper of BPEL4People [3]: “Human user interactions are currently not covered by WS-BPEL, which is primarily designed to support automated business processes based on WS. In practice, however, many business process scenarios require user interaction.” The introduction of human interaction also leads to other interesting concepts such as roles and permissions, which makes the design and verification of human workflows even harder. Because of the importance, there were several attempts that try to add human support into BPEL [4].

To support and standardize human activities in BPEL, BPEL4People is proposed by IBM-SAP, which describes features for supporting People Activities within existing BPEL standards, and introduces the principle of manual tasks executed by human participants. Some newly aroused problems such as task authorization and escalation are also described in BPEL4People specification. Strictly speaking, BPEL4People includes two specifications: WS-BPEL Extension for People and WS-HumanTask. The former extends BPEL language with “people activity” and makes it as a normal Web service invocation. The latter defines concepts such as human roles, tasks, and permissions used within people activities. The reason for the separation is mainly commercial: it allows vendors to create separate products and users to choose from different products. Since both of them must be used together to compose a human workflow, we refer to them as a whole in this paper.

As BPEL4People being used, formal models (as GSPN) of BPEL4People have been proposed [7], that not only help people to better understand the specification, but also provide insight to the subtle but important issues of BPEL4People. Furthermore, based on the formal model, formal analysis such as performance-model-checking can be done to discover potential problems in BPEL4People source codes. GSPN is a natural extension of the classical Petri Net.There are several reasons for selecting GSPN as the language for modeling work distribution in the context of BPEL4People. First of all, GSPN have formal semantics and allow for different types of analysis, e.g., state-space analysis and invariants [2]. Second, GSPN are executable and allow for rapid prototyping, gaming, and simulation. Third, GSPN are graphical and their notation is similar to existing workflow languages. Finally, the GSPN language is supported by GREATSPN Tool [8] a graphical environment to model, enact and analyze GSPN.

In this paper, we are motivated by issues related to the definition of a model method for the transformation of BPEL4People into GSPN and this way evaluate its performance.

The remainder of this paper is structured as follows. A backgroud from the technology used in the construction of our model method for the transformation of BPEL4People into GSPN is presented in Section 2. Section 3 provides the details of transformation WS-BPEL4People into GSPN. In section 4, we discuss one example of transformation of WS-BPEL4People into GSPN. Section 5 talks about related works. Section 6 gives a conclusion of this paper.

2. Background: SOA, SPE, PETRI NETS (PN), BPEL and BPEL4People

In this section will be made a retrospectively of technologies used in the construction of our model method for the transformation of BPEL4People into GSPN .

2.1. SOA

Service-Oriented Architecture (SOA) is an architecture for building distributed software systems on the basis of autonomous, interoperable, discoverable and reusable software entities called services [11]. Services incapsulate business functionalities of involved parties and in that way make the Business Process Management (BPM) to a natural application of the Service-Oriented Architecture.

The SOA Architecture expresses a concept where applications or routines are available as services in a computer network (Internet or Intranet) and they communicate through open standards [12]. Most implementations of SOA are used for Web Services. A SOA can be used in any standard based on web technology.

The basis of the standard Web Services relevant to the SOA include:

- **XML (*eXtensible Markup Language*)** - is a markup language to describe data in loads of messages in a document format;
- **SOAP (*Simple Object Access Protocol*)** - is an XML-based protocol used for exchanging information in a distributed environment;
- **WSDL (*Web Services Description Language*)** - is an XML document that describes a set of SOAP messages and how these messages are exchanged. As the WSDL is XML, it is readable and editable, but in most cases, it is generated and consumed by software; and
- **UDDI (*Universal Description, Discovery, and Integration)*** - is a way to locate a Web Service in a register, like a catalog of yellow pages, so that a program in search of a particular service can easily find and understand what the service does.

According to [13] the SOA can be well represented from the following model involving three main parts: the Service Providers, the Service Consumer and the Registry. This model is also called "find-bind-execute paradigm" as best shown in Figure 1.

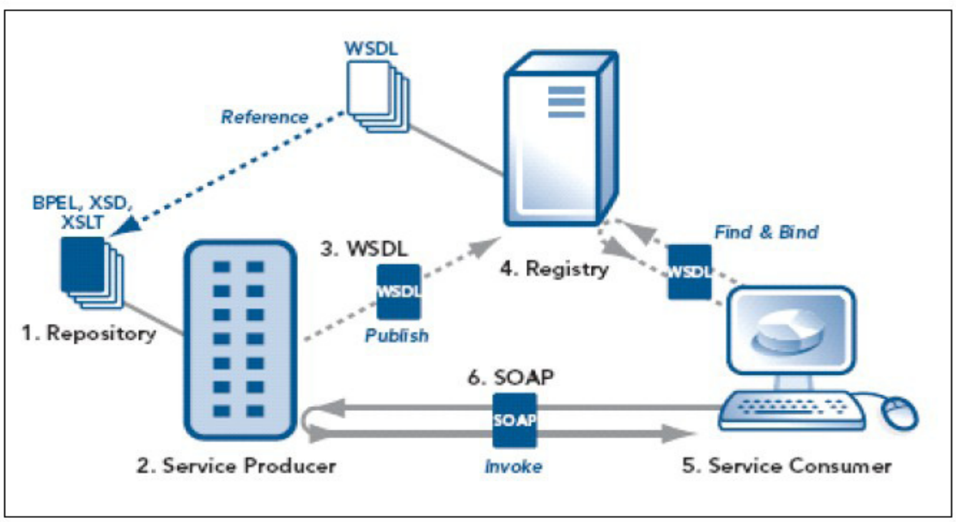

Figure 1. The SOA Architecture [13]

In this work we utilise the "find-bind-execute paradigm" for SOA implementation, i.e., SOA is implemented as Web Services.

2.2. SPE

Software Performance Engineering (SPE) has been defined as a method for constructing software systems to meet performance objectives. Smith's book, "Performance Solutions" [14], is an excellent bibliography to understand the methodologies of SPE. It states: "The process begins early in the software life cycle and uses quantitative methods to identify satisfactory designs and to eliminate those that are likely to have unacceptable performance, before developers invest significant time in implementation. SPE continues through the detailed design, coding and testing stages to predict and manage performance of the evolving software and to monitor and report actual performance against specifications and predictions. SPE methods cover performance data collection, quantitative analysis techniques, prediction strategies, management of uncertainties, data presentation and tracking, model verification and validation, critical success factors, and performance design principles" [14].

By the prospect of being able to evaluate the performance of *BPEL4People processes*, before it is implemented, we use the principles of SPE in our work.

2.3. PETRI NETS (PN)

A Petri Net is a mathematical formalism represented graphically by a bipartite graph containing *places* (drawn as circles) and *transitions* (drawn as rectangles). Places hold *tokens* and represent predicates about the world state. Transitions are the active component. When all of the places pointing into a transition contain an adequate number of tokens, the transition is *enabled* and may *fire*, removing tokens from its input places and depositing a new set of tokens in its output places. The most relevant features of Petri Nets for our purposes is their ability to model events and states in a distributed system and capture sequentiality, concurrency and event-based asynchronous control [17].

Extensions to the basic Petri Net formalism that include typed arcs, hierarchical control, timed transitions, parameterization, typed (individual) tokens and stochasticity are called SPN (Stochastic Petri nets).

Stochastic Petri nets are a mathematical formalism that allows graphical representation, and has powerful methods of formal analysis and performance (using its extensions timer) of a system. The "places" and "transitions" of Petri Nets are used to shape vision points of logic systems. The idea of involving random variables of delay exponentially distributed to transitions has been explored for the first time by Natkin [17] and Molloy [15], allowing the emergence of Stochastic Petri Nets (SPN´s) and its extensions. SPN´s have the property of being similar to Markov Chains of Continuous Time, allowing the same analysis stationary and transient known that can be applied to those nets. The extension Generalized Stochastic Petri Nets (GSPN), originally proposed by Marsan [16], is one of the most popular extensions.

A GSPN is defined by an 8-upla GSPN = (P, T, I, O, H, Π, W, M0) [16]. The set of places P is the availability of resources (marks on the net) system, states and local variables of the system. The set of transitions T is the set of actions that can cause change of state. This collection is divided in the subset of timed transitions and the subset of transitions that describe immediate instant actions, with higher priority, the immediate transitions. The function "W" combines a non-negative real number that indicates the rate of exponential distribution to each timed transition. The function "Π" defines the level of priority of each timed immediate transition (the priority of a immediate transition is zero). We have "I" and "O" functions such as mapping of places to transitions and transitions to places, respectively, while the function "H" represents the inhibitors arcs. The marking M0 is the initial state of the system. The GSPN´s have been widely used for modeling various types of systems, from manufacturing systems to wireless networks.

For this article, the crucial fact about the utilization of GSPN in representation is that it is active with a well defined durative transition semantics for service descriptions.

2.4. BPEL and BPEL4People

Business process management is designed to make business activity coordination easier and more cost effective [6]. WS-BPEL and BPEL4People extension together coordinate the WS and human activities within business process. With the development of globalisation organisations become more dynamic and the underlying business process are frequently optimising in today's business world. Adapting business processes to market changes and automating business services on demand are main necessities to facilitate business collaboration among existing and potential partners.

In business process oriented environment, a unified process specification language is significantly crucial in term of collaboration. WS-BPEL is one such language that provides the syntax and notations for specifying business processes behaviour based on WS. Besides automatic WS, most business processes still require human interactions. BPEL4People addresses this important aspect to provide human actor support.

2.4.1. BPEL

WS have become widely accepted as the defacto standard for distributed business applications [4]. They bring maximum interoperability, use an open and flexible architecture, and the implementation and complexity of a WS can be hidden towards a service requestor. Layered on top of these services, BPEL, the de–facto standard for orchestration, formally describes processes.

Web service composition refers to the creation of new (Web) services by combination of functionality provided by existing ones. This paradigm has gained significant attention in the WS community and is seen as a pillar for building service-oriented applications. A number of domain-specific languages for service composition have been proposed with consensus being formed around a process-oriented language known as WS-BPEL (or BPEL). The kernel of BPEL consists of simple communication primitives that may be combined using control-flow constructs expressing sequence, branching, parallelism, synchronisation, etc. As a result, BPEL process definitions lend themselves to static flow-based analysis techniques.

The business process execution language (BPEL) is an XML subset for specifying and executing business processes. As interactions are realised with WS for maximum interoperability between various heterogeneous systems, BPEL permits orchestration of WS.

2.4.2. BPEL4People

WS-BPEL focuses on business processes that orchestrate Web service interactions. However, in general, business processes are comprised of a broad spectrum of activities that most often require the participation of people to perform tasks, review or approve steps and enter data — for example, a credit approval scenario that may require approval on certain transaction limits or activity levels. These human interactions are now addressed in the new specifications [5].

Human user interactions are currently not covered by the Web Services Business Processes Execution Language (WS-BPEL), which is primarily designed to support automated business processes based on WS. In practice, however, many business process scenarios require user interaction.

So far we've seen that user interaction in business processes can get quite complex. Although BPEL specification 1.1 (and the upcoming BPEL 2.0) doesn't specifically cover user interactions, BPEL is appropriate for human workflows. Workflow services that leverage the rich BPEL support for asynchronous services are created today. In this fashion, people and manual tasks become just another asynchronous service from the perspective of the orchestrating process and the BPEL processes stay 100% standard.

We now see the next generation of workflow specifications emerging around BPEL with the objective of standardizing the explicit inclusion of human tasks in BPEL processes. This proposal is called BPEL4People and was originally put forth by IBM and SAP in July 2005. Other companies, such as Oracle, have also indicated that they intend to participate in and support this effort.

The most important extensions introduced in BPEL4People are people activities and people links. People activity is a new BPEL activity used to define user interactions; in other words, tasks that a user has to perform. For each people activity, the BPEL server must create work items and distribute them to users eligible to execute them. People activities can have input and output variables and can specify deadlines.

To specify the implementation of people activities, BPEL4People introduced tasks. Tasks specify actions that users must perform. Tasks can have descriptions, priorities, deadlines, and other properties. To represent tasks to users, we need a client application that provides a user interface and interacts with tasks: it can query available tasks, claim and revoke them, and complete or fail them.

To associate people activities and the related tasks with users or groups of users, BPEL4People introduced people links. People links are somewhat similar to partner links; they associate users with one or more people activities. People links are usually associated with generic human roles, such as process initiator, process stakeholders, owners, and administrators [5].

BPEL4People extends the capabilities of WS-BPEL to support a broad range of human interaction patterns, allowing for expanded modeling of business processes within the WS-BPEL language.

BPEL4People is comprised of two specifications including:

- WS-BPEL Extension for People which layers features on top of WS-BPEL to describe human tasks as activities that may be incorporated as first-class components in WS-BPEL process definitions.
- Web Services Human Task introduces the definition of stand-alone human tasks, including the properties, behavior and operations used to manipulate them. Capabilities provided by Web Services Human Task may be utilized by Web services-based applications beyond WS-BPEL processes.

In the next sections we utilise this technologies to define our model method for the transformation of BPEL4People into GSPN, that is the objective of this work.

3. Transformation from WS-BPEL4People into GSPN

In this section, we introduce the transformation from BPEL4People into GSPN. Since a WS-BPEL4People includes both BPEL activities and new elements about human task, our work has two parts: transformation for BPEL activities and transformation for the authorization of human task.

3.1 Transformation of BPEL Activities into GSPN

BPEL has structural activities and basic activities. Structural activities and links are used to construct the control flows in business processes. Petri Net is proven to be appropriate to model workflow or business process, especially to model their control flows [1] .

3.1.1 Representation of Basic Activities into GSPN

The Basic Activities are those that describe the steps of an elementary activity. BPEL defines the following Basic Activities: <Process>, <Invoke>, <Receive>, <Reply>, <Wait>, <Empty> and so on. The representation of the Basic Activities into GSPN is shown in Figure 2.

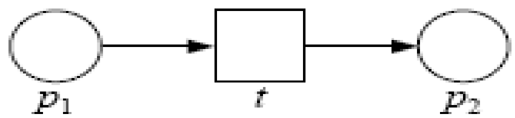

Figure 2. Representation of Basic Activities

3.1.2 Representation of Structure Ativities

The Structured Activities prescribe the order in which a set of Basic Activities is executed. To enable the representation of complex structures, BPEL defines the following Structured Activities: <Sequence>, <Switch>, <While>, <Pick>, <Flow> and <Control Link>. Here we present their representation into GSPN.

● **Sequece Structure**: this structure contains one or more activities that are carrie out consecutivety. Its representation is show in Figure 3.

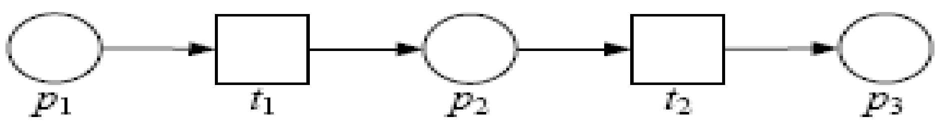

Figure 3. Logic of Representation of Sequence Structure

● **Switch Structure*:*** this structure supports conditional choices. Where only one of the transitions ("t1" to "tn") are fired when the arrival of a token on "p1". Its representation is shown in Figure 4.

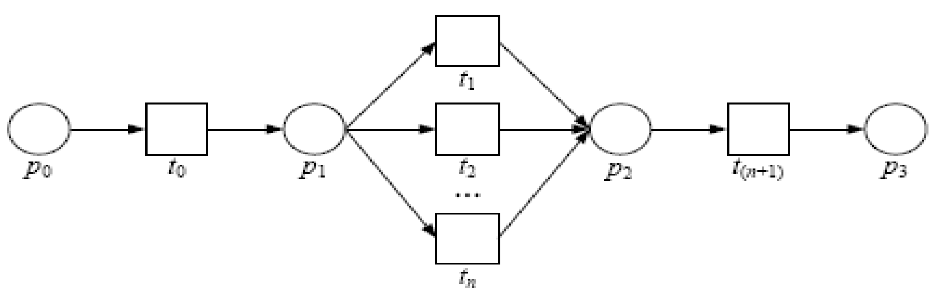

Figure 4. Logic of Representation of Switch Structure

● **While Structure*:*** this structure allows one or a series of activities executives: none, one or more times. Figure 5 shows the representation of this structure. The transitions "t2" to "tn" can fire in a repetitive way, until the transition "t1" fires and shut down the cycle of repetitions.

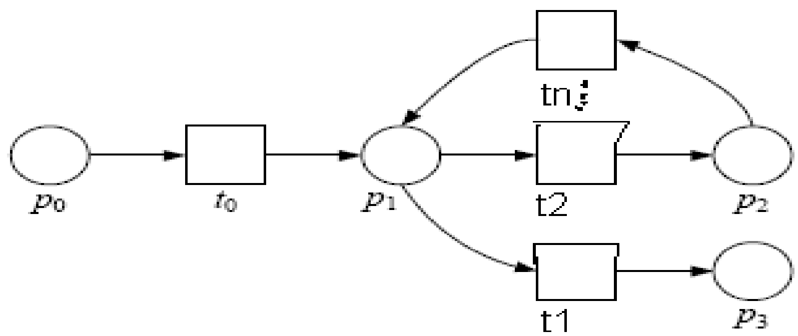

Figure 5. Logic of Representation of While Structure

● **Pick Structure:** the *pick* construct awaits the occurrence of one of a set of events and then performs the activity associated with the event that occurred. The representation of the *pick* structure is the same as the representation of the *switch* structure shown in Figure 4.

● ***Flow Structure*:** the BPEL flow lets specify one or more activities to be carried out simultaneously. This fact leads to the definition of Flow Structure which is shown in Figure 6. In this representation the weight of the arc output of the transition “t0“ is “n”, then the transitions “t1” to “tn” can fire simultaneously.

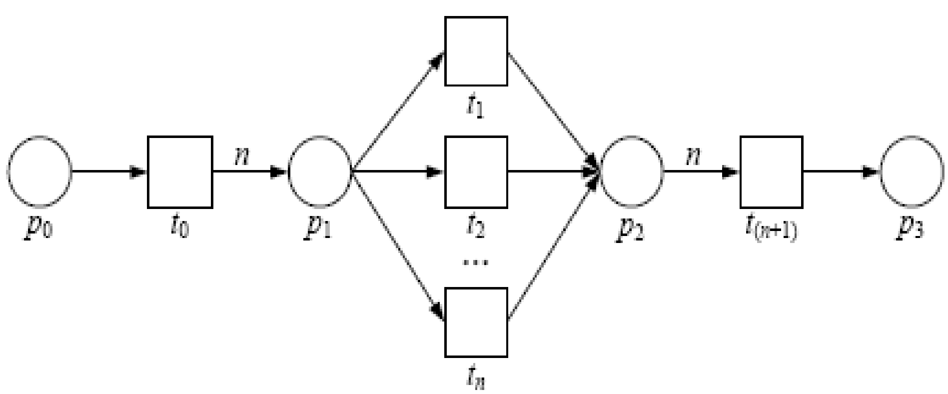

Figure 6. Logic of Representation of Flow Structure

● **Control Link Translation***: more generally, the Flow activities allows the dependence of synchronization between the activities that directly or indirectly are nested within it. The Control Link structure is used to express these dependencies of synchronization. The sequence of representation of this structure is shown in Figure 7. This representation shows that there is a synchronism between the transitions “t2” and “tn”. The transition tn will fire after “t2” finishes its processing to be put a token in the place “p5” and therefore make the transition tn enabled.*

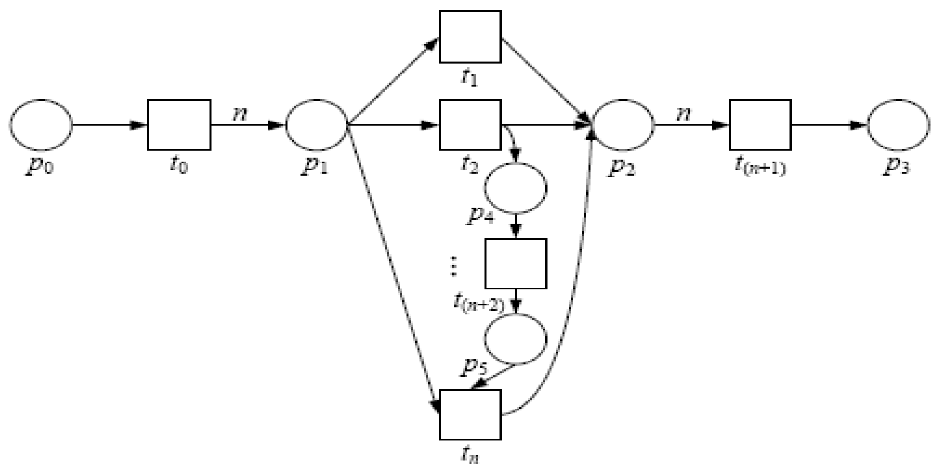

Figure 7. Logic of Representation of Control Link Translation

3.1.3 Attribution Time to GSPN

In the transformation of BPEL codes into GSPN, the firing of transitions is immediate, except in the transitions that represent the Basic Activity <Invoke> that recive as Delay Time, the values of PDF of the response time of each of these Service Providers (SP), where services is executed [1].

To model the estocastic behavior of response time of Service Providers (SP´s), we will make use of PDF.

As entries of the PDF, will be used the Average and Standard Deviation of the response time of Service Providers, while as the output is expected the value of Delay Time of the transition (λ).

These response times of SP´s provide a sample with unknown distribution with Average (μ) and Standard Deviation (σ).

The Average (μ) is calculated as the arithmetic average of the response time of Service Providers and the Standard Deviation (σ) is calculated as shown in Figure 8.

Depending on the value of Coefficient of Variation (CV), these response time are approximate to one of the distributions: Erlang, Hiperexponential or Hipoexponential. This makes it possible to represent the probability issue involved in approximation of these response times of Service Providers for a Delay Time (λ) of the transition that it model.

If the Coefficient of Variation (CV) is greater than 1 (CV> 1) and it the same is an integer value, the sample must be empirical approximate with Erlang Distribution. In this case the Delay Time (λ) of the transition that shapes this Service Providers will be calculated as shown in Figure 8.

If CV> 1 (CV is not a integer number), the distribution should be approximated with Hiperexponecial Distribution and the Delay Time (λ) of the transition that shapes this Service Providers will be calculated as shown in Figure 8.

And if CV< 1, the distribution should be approximated with Hipoexponential Distribution and the Delay Time (λ) of the transition that shapes this Service Providers will be calculated as shown in Figure 8.

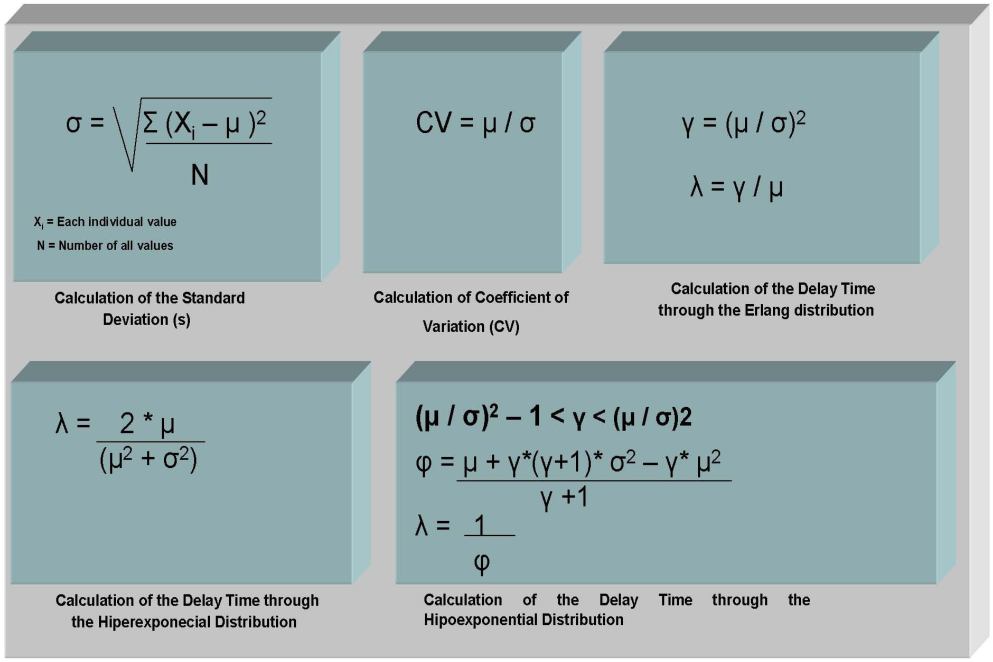

Figure 8. Calculation of the σ, cv and λ

3.2 Transformation for the Authorization of Human Task

As a complement of BPEL, BPEL4People introduces new concepts to support human task. *People links* are used to bind a group of people to a business process similar to the manner that partner links are used to bind Web services to processes. When the business process engine encounters a *people activity*, it may suspend the business process until a person of the group completes the associated task. The details of a people activity, i.e. who can execute this activity, are defined in the specification of WS-HumanTask.

To define a complete human business process, both specification must be used, where BPEL4People can be seen as a bridge between BPEL and WS-HumanTask. Terms “human task” and “people activity” are synonyms used by these two specifications, while we will use “human task” uniformly in the paper. WS-HumanTask introduces also the idea of logical people groups. Each group includes a set of people, and each human task can only be executed by some specified groups. This can be viewed as a simple RBAC (Role Based Access Control) model, where logical people group is the synonym of “role”. In the following, we will use the shorter term “role” instead of “logical people group”.

```
<b4p:peopleActivity
<htd:task name="votingTask">
<htd:interface operation="vote"portType="el:votingPT"/>
<htd:peopleAssignments>
<htd:potentialOwners>
<htd:user> NmUser1 </htd:user>
<htd:user> NmUser2 </htd:user>
</htd:potentialOwners>
<htd:excludedOwners>
<htd:user> NmUser3 </htd:user>
</htd:excludedOwners>
</htd:peopleAssignments>
</htd:task>
</b4p:peopleActivity>
```

Figure 9. An Example of People Activity

As shown in Figure 9, a people activity wraps a human task as an activity that is similar to a normal service invocation in BPEL. The human task defines the permission of the task with the potentialOwner element, which refers to some people link defined elsewhere. To resume the business process from a working human task, the person who executes the task (in BPEL4People terminology, the "actual owner" of the task) must notify the business process engine when the task is successfully/unsuccessfully completed.

3.2.1 Integrating Authorization Constraints

BPEL4People support features to exclude some users from performing a task because of some tasks they had done before, or force some user to perform a sequence of tasks. We call such requirement as *authorization constraint*, as the term is widely used in access control literature. In this section we will use GSPN to express the authorization constraints to facilitate formal analysis.

Two kinds of authorization constraints, namely "4-eyes principle" and "chained execution", are proposed in BPEL4People specification. The "4-eyes principle", also known as "separation of duty", is a common scenario in many application areas when a decision must be made by two or more people independently of one another, often for the security reasons, and "chained execution" refers a process fragment where a sequence of steps must be executed by one person.

3.2.1.1 Separation of duty

The separation of duty (SoD) is a well-known principle in authorization to prevent fraud or error by requiring that at least two individuals are involved in some specific work. SoD is also useful when two persons have to co-operate in a work but none of them should know all the details.

The basic form of SoD states that two given distinct tasks "t1" and "t2" must be performed by different individuals. This can be defined as states that person "p0" cannot perform both "t1" and "t2". We can define variations of this similarly, e.g., "task t1 and t2 must be performed by different roles". We can also define SoD constraint for a specific person, e.g., "person A cannot invoke both task t1 and t2".

3.2.1.2 Binding of duty

"Binding of duty" (BoD) is the dual of SoD, which states that some distinct tasks must be performed by one person. BoD is used to define the responsibility of a person, e.g.: It states that if "p0" performs "t1", then "p0" must also perform "t2", and vice versa.

SoD and BoD may be combined to define more complex constraints.

3.2.2 Transformation of SoD and BoD into GSPN

The representation of the transformation of each users of **<potentialOwner>** into GSPN is modeled by a transition "t", by two places "p1" and "p2", and two arcs linking each places to a transition, as shown in Figure 3. A token in place "p1" represents that the user modeled by the transition "t" is able to execute the task. The place "p2" will contain tokens after the firing of transition "t", and this represent that user executed the task.

In the Attribution Time to GSPN, the tansition "t" is not immediate, transition "t" should recive as Delay Time (λ), the values of Probability Distribution Function (PDF) of the response time of each user, who executed the task. To model the estochastic behavior of response time of users, we will make use of PDF as showed in section .1.3.

For each user presented in clause **<excludedOwners>** of a task, it should be eliminate the module that modeled that user in those task in the model.

To illustrate the transformation of WS-BPEL4People into GSPN, suppose that you have the following WS-BPEL4People code, as shown in Figure 10.

4. Example of the Transformation of WS-BPEL4People into GSPN

Now we show some further issues related to one example of transformation from BPEL4People into GSPN, and in general, to the modeling of human process.

This example is a BPEL4People source code for a purchasing process "WS- PurchSys" that is showed in Figure 10. Four tasks are defined: *manager approve*, *finance approve*, *notify staff*, and *purchase*. The potential owners of each task are: *manager approve* (*Alan*); *finance approve* (*Ben*); *purchase* (*Ben*, *Cindy*, *Diana*); *notify staff* (*Diana*, *Edward*). The excluded owner of purchase is the actual owner of *finance approve*. The excluded owner of *notify staff* is the actual owner of purchase.

```
<process name="purchasing">
<b4p:humanInteractions><htd:tasks>
<htd:task name="manager_approve">
<htd:peopleAssignments><htd:potentialOwners>
<htd:user>Alan</htd:user></htd:potentialOwners>
</htd:peopleAssignments></htd:task>
<htd:task name="finance_approve">
<htd:peopleAssignments><htd:potentialOwners>
<htd:user>Ben</htd:user></htd:potentialOwners>
</htd:peopleAssignments></htd:task>
<htd:task name="purchase"><htd:peopleAssignments>
<htd:potentialOwners>
<htd:user>Ben</htd:user><htd:user>Cindy</htd:user><htd:user>Diana</htd:user>
</htd:potentialOwners><htd:excludedOwners>
<htd:getActualOwner("tns:finance_approve")
</htd:excludedOwners>
</htd:peopleAssignments></htd:task>
<htd:task name="notify_staff"><htd:peopleAssignments>
<htd:potentialOwners>
<htd:user>Diana</htd:user><htd:user>Edward</htd:user></htd:potentialOwners>
<htd:excludedOwners>
<htd:getActualOwner("tns:purchase")
</htd:excludedOwners>
</htd:peopleAssignments></htd:task>
</htd:tasks></b4p:humanInteractions>
<sequence> <receive createInstance="yes"/>
<extensionActivity>
<b4p:peopleActivity name="manager_approve"
outputVariable="result"/>
<b4p:localTask reference="tns:manager_approve"/>
</b4p:peopleActivity>
</extensionActivity>
<Switch>
   <case>
        <Invoque name “result"/>
    <case>
       <Invoque name “approved"/>
</Switch>
</sequence>
</process>
```

Figure 10. BPEL4People Source Code for the "WS-PurchSys"

4.1. GSPN Model

Figure 11 shows the GSPN of the BPEL4People code for the "WS-PurchSys" presented in Figure 10, modeled inside the GREATSPN Tool [8].

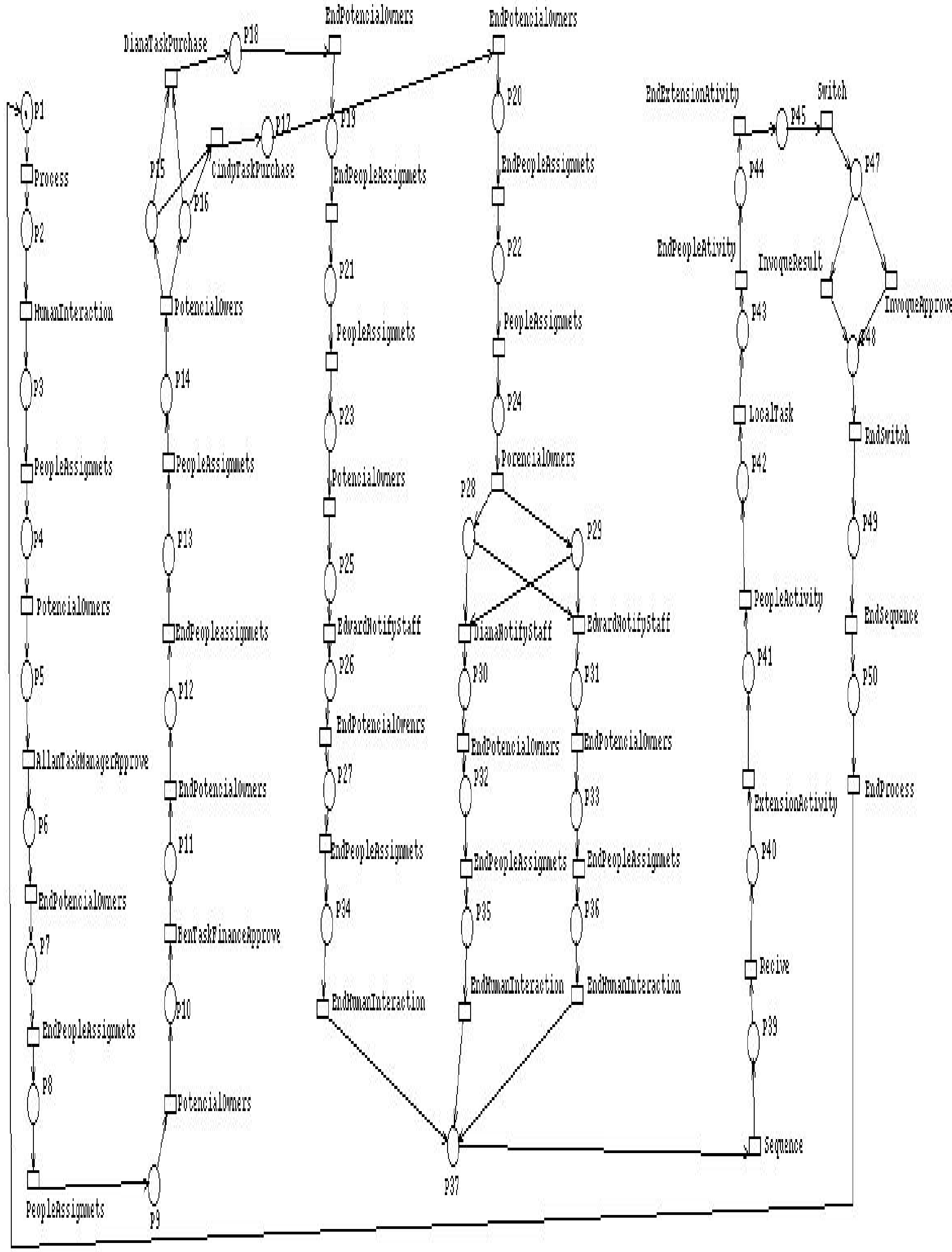

Figure 11. Model GSPN for BPEL4People Source Code for the "WS-PurchSys"

4.2. Performance Analysis of Model

With the GSPN for BPEL4People Source Code for the "WS-PurchSys" loaded on GREATSPN tool it begins the activities of the performance evaluation. The performance analysis of the model is made from simulations with the same amount of requests made in the "WS-PurchSys".

These results shows that the response times of the model generated by our model method and the response times of the "WS-PurchSys" not differ by more than 6.7%, proving itself as the usability and validity of our model method in Performance Analysis of WS-BPEL4People.

Figure 12 presents a comparison of response times of the model generated by our model method for "WS-PurchSys" and the response times of the "WS-PurchSys".

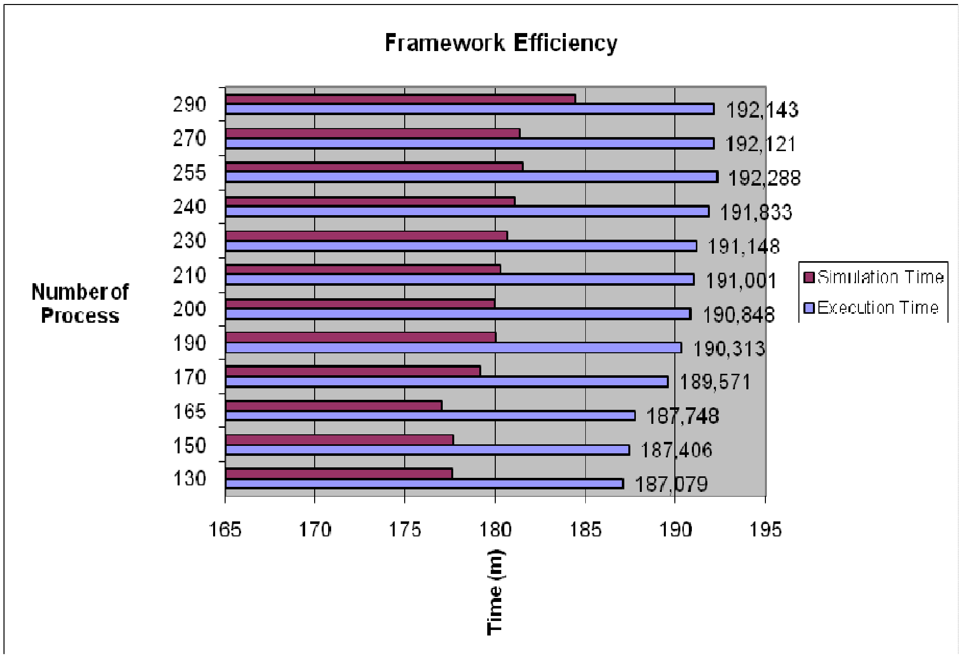

Figure 12. Our Model Method Efficiency

5. Related Works

Software Performance Engineering (SPE) and QoS in the context of Web Service is the subject of many studies.

In [18], the authors developed a methodology issue of performance evaluation of Web Services. Their methodology is focussed on capacity planning using Queuing networks (QN). We aim at evaluating performance of WS using GSPN.

The Web Service Trust Center (WSTC) is a platform for development and evaluation of measurement tools and techniques in the field of Service Oriented Architectures (SOA) and web services. One of their publishers titled "Performance Modeling of WS-BPEL-Based Web Service Compositions" [19], addresses quality of service aspects of web service orchestrations created using WS-BPEL from the standpoint of a web service integrator. A mathematical model based on operations research techniques and formal semantics of WS-BPEL is proposed to estimate and forecast the influence of the execution of orchestrated processes on utilization and throughput of individual involved nodes and of the whole system. This model is applied to the optimization of Service Levels Agreement process between the involved parties [19].

Our work is different from the work presented in [19], in fact we use GSPN to evaluate the performance of WS orchestrating with BPEL4People and not a pure mathematical model as the authors of that proposal. The difference of using GSPN is that they are also mathematical models with the advantage of providing a good view of the system model.

Much work has been done on analyzing the interactions of BPEL processes, applying the BPEL analysis, and workflow authorization modeling. In paper [6], the authors develop Petri Net semantics for BPEL which support the translation from BPEL into Petri Net and analysis of BPEL processes. In paper [9] a framework is provided to analyze interactions of WS-BPEL. WS-BPEL are eventually translated into PROMELA and LTL formulas, which are then checked with SPIN. Based on model Checker Bogor, Bianculli *et al.* [10] present an approach for the formal verification of workflow-based compositions of web services, described in BPEL. The paper [7] present a formal framework that integrates RBAC into BPEL and verifies BPEL workflows under authorization constraints with model Checker SAL.

The related work showed that there has been a lot of studied and researched in the performance of Web Services. However most of these studies and research promote the evaluation of the performance of Web Services focussing on optimizing their composition when created using WS-BPEL. In our work we want to address the issue, wich has not been explored yet, concerning the performance evaluation of the WS-BPEL4People.

Others related works of interest are cited next.

For Silva and Lins, Web Services have played an important role in the development of Distributed Systems. In particular, the possibility of composing already implemented Web Services in order to provide a new functionality is an interesting approach for building Distributed Systems. However, choosing the better composition is still a challenger as different qualities may be observed in the composition, such as security, performance, fault tolerance, and so on. In this context, the paper [20] proposes a methodology based on Stochastic Petri Nets to model, evaluate and help to choose Web Service compositions considering performance aspects.

For P. Bera, Pallab Dasgupta and S. K. Ghosh, in their paper [25], a fault analysis module is incorporated along the verification framework which as a whole can derive a correct ACL implementation with respectto a given security policy specification and can ensure that a correct security implementation is fault tolerant to certain number of link failures. The verification module can find the correct security implementation and the fault analysis module can find the number of link failures the existing security implementation can tolerate and still satisfy the security policy of the network.

In [21], Chandrasekaran proposes a simulation technique for analyzing performance of composite web services in order to obtain efficient web processes. Their paper describes the Service Composition and Execution Tool (SCET) and various methodologies that could be adopted for evaluating the performance of a Web process. SCET allows for composing services statically using its designer and storing them as Web Service Flow Language (WSFL) based specifications. Executing a process enables one to realize its functionality and also analyze its performance.

In [22], Narayanan and his group take the DAMLS ontology for describing the capabilities of Web Services and define the semantics for a relevant subset of DAML-S in terms of a first-order logical language. With the semantics in hand, they encode service descriptions in Petri Net formalism and provide decision procedures for Web Service simulation, verification and composition.

Elena Gómez-Martí nez Elena and José Merseguer recall, from the literature, a performance study of a Web Service. This study, based on the Layered Queuing Network (LQN) paradigm, is now addressed following the PUMA approach to obtain a new performance model, in this case in terms of Petri Nets, for the target Web Service. Such Petri Net model is used to extend the previous LQN results with respect to some key Web Service performance aspects: the SOAP toolkit and the XML parsers [23].

In [24], for the authors, Web Service composition involves the combination of a number of existing Web Services to create a value-added service in way that may not be foreseen at the time when a Web Service is written. BPEL is a promising language which describes Web Service composition in form of Business Processes. However, BPEL is an XML-based language and may suffer from ambiguities or some erroneous properties. It is necessary to analyze business processes specified in BPEL with a formal tool. In their paper, the authors put forward an approach to model and verify BPEL based on ServiceNet, a special class of Petri nets. They present some transformation rules of BPEL business processes into ServiceNet. Then the throughness of a BPEL business process can be verified by reducing the ServiceNet based on some reduction rules.

6. Conclusion of this Paper and Future Works

Service orchestrations enable complex applications to be put together in a variety of ways. Each possible service selection of services brings different levels of QoS. Thus, there is a need to devise fast and efficient mechanisms that can be used for performance analysis of WS-BPEL4People among a set of service providers. In this paper, we provide a model method for the transformation of BPEL4People into GSPN and this way evaluate its performance. This paper presented also such an efficient mechanism that, in all experiments reported, comes very close to the real response time of WS-BPEL4People (less than 8% worse) after having compared with the time of the model generated by our model method.

Our future work includes developing a automatic transformation tool (from WS-BPEL4People into GSPN), and this way analyzing and classifying performance constraints on WS-BPEL4People.

ACKNOWLEDGEMENTS

We would like to acknowledge our colleagues in the Centro Politécnico Superior(CPS) of University of Zaragoza (UNZAR) in the help and collaboration in the development of a model method for the transformation of WS-BPEL4People/BPEL4People into Generalized Stochastic Petri Nets (GSPN) and this way evaluate its performance. We would also like to thanks the Professor Giovanni Cordeiro Barroso for our interesting discussions on various aspects of this work. We particularly thank the Professor José Merseguer Hernáiz for believing in my abilities and always be present in the development of this work and for your review for an informative and thorough review of this article. Finally we gratefully acknowledge all those who directly and indirectly involved in execution of this work.

References

[1] Holanda, Henrique Jorge A., Barroso, Giovanni Cordeiro, Serra, Antonio de Barros (2009) "Performance Analysis of SOriented Software", In: iSys – Revista Brasileira de Sistemas de Informação, Volume II, ISSN eletrônico 1984-2902, 2009.

[2] Henrique J. Holanda, G. Barroso and Antonio de B. Serra, "SOASPE: a Framework for the Performance Analysis of Service Oriented Software", SBSI, pp.204-215, May 2009, Brasilia.

[3] Dyer, S., Martin, J. and Zulauf, J. (2005) "Motion Capture White Paper", http://reality.sgi.com/employees/jam_sb/mocap/MoCapWP_v2.0.html, December.

[4] Martin Vasko, “VieBOP: Extending BPEL Engines with BPEL4People”, 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing Ta’id Holmes, Schahram Dustdar Distributed Systems Group, Institute of Information Systems Vienna University of Technology, 2007,Vienna, Austria.

[5] A. Agrawal, M. Amend, M. Das, M. Ford, C. Keller, M. Kloppmannand, D. Konig, F. Leymann, R. Muller, G. Pfau, K. Plosser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, and M. Zeller, “Web Services Human Task (WS-HumanTask) “, 1st ed., Jun 2007.

[6] E. Bertino, J. Crampton, and F. Paci. “Access control andauthorization constraints for WS-BPEL“. In ICWS 2006.

[7] X. Zhao, A. Cerone, and P. Krishnan. “Verifying BPEL workflows under authorisation Constraints“. In Proc. of BPM’06, volume 4102 of LNCS. Springer.

[8] PE group (2006). “GreatSPN User’s Manual (version 2.0.2) “. University of Torino, Italy. http://www.di.whito.it/greatspn.

[9] X . Fu, T. Bultan, and J. Su, “Analysis of interacting BPEL web services,” in Thirteenth International World Wide Web Conference Proceedings, WWW2004, May 2004, pp. 621–630.

[10] D. Bianculli, C. Ghezzi, and P. Spoletini, “A model checking approach to verify BPEL4WS workflows,” in IEEE International Conference on Service-OrientedComputing and Applications, 2007, pp. 13–20.

[11] W3C. (2002) “Web Services Description Requirements“.

[12] Menascé. (2004) “Composing Web Services: A QoS View“. IEEE Internet Computing, (6):88–90.

[13] Arsanjani, A. (2004) “Service-oriented modeling and architecture: How to identify, , and realize services for your SOA“. IBM developerWorks.

[14] C. Smith and L. Williams. (2001) “Performance Solutions“. Addison-Wesley. Natkin, S. (1980) “Les Réseaux de Petri Stochastiques et Leur Application a L’evaluation des Systemes Informatiques”, Thèse de Docteur Ingegneur, CNAM, Paris, France.

[15] Molloy, M. (1981) “On the Integration of Delay and Throughput Measures In Processing Models”, Ph.D.Thesis, UCLA.

[16] Marsan, M.A. et al. (1995) “Modelling with generalized stochastic Petri Nets”, à degli studi di Torino, Dipartimento di informática.

[17] Natkin, S. (1980) “Les Réseaux de Petri Stochastiques et Leur Application a L’evaluation des Systemes Informatiques”, Thèse de Docteur Ingegneur, CNAM, Paris, France.

[18] Menascé, D. A. and Almeida, V. F. (2001) “Capacity Planning for Web Service: models, and métodos“. Prentice Hall. 608P.

[19] Rud D., Schmietendorf A. and Dumke R. (2006) “Performance modeling of WS-BPELbased web service compositions“. In Proceedings of the IEEE Service Computing Workshops (SCW 2006), pages 140-147, Los Alamitos, CA, USA, September 2006.

[20] Silva, A. N. And Lins, F. A. (2006) “Avaliação de Desempenho da Composição de *Web* Services Usando Redes de Petri“. In: SBRC, 2006, Curitiba. 24o. Simpósio Brasileiro de Redes de Computadores.

[21] S. Chandrasekaran, J. Miller, G. Silver, I. Arpinar, and A. Sheth. (2003) “Performance Analysis and Simulation of Composite Web Services”. *Electronic Markets*.

[22] Srini Narayanan**,** McIlraith. (2002) “Analysis and Simulation of Web Services“. Eleventh International World Wide Web Conference (WWW11).

[23] Gómez-MartíıElena, Merseguer José. (2006) “Impact of SOAP implementation in the performance of a Web-Server-based Application“. LNCS 4331, pp. 884–896.

[24] Haiqiang Dun, Haiying Xu, Lifu Wang. (2008) “Transformation of BPEL Processes to Petri Nets“. 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering.

[25] P Bera, Pallab Dasgupta and S K Ghosh. (2009), “Formal Analysis of Security Policy Implementations in Enterprise Networks “. In International Journal of Computer Networks & Communications (IJCNC). July, Volume 1, Number 2. ISSN Online: 0974 – 9322.

Authors

Henrique Jorge Amorim Holanda is an assistant professor at the Computer Science Department of the University do Estado do Rio Grande do Norte, where she teaches software engineering. He is a Master Degree in Computer Science, PhD student in Federal University of Ceara, and works with performance evaluation of web services.

José **Merseguer Hernáiz** is an Assistant Professor at the Dpto. de Inf. e Ing. de Sistemas, UNIZAR, Zaragoza, Spain, where he teaches Ingenieria del Software II. He is a member of **Group of Discrete Event Systems Engineering** – GISED in UNIZAR

Giovanni Cordeiro Barroso is an adjunct professor at the Physics Department of the Federal University of Ceara, where he teaches modeling hybrid systems. He is a Pós-PhD in Teleinformatica pelo Institut National des Télécommunications - INT, Evry-France and works in Petri nets, modeling and analysis, coloured petri nets, distributed systems and discrete event systems.

Antônio de Barros Serra is a professor at the Telematica Engineer Computer Science Department of the Federal Institute of Educ. Ciencia e Tecnologia, where he teaches computer networks. He is a PhD in Réseaux Avancés de Connaissances et Organisations of Institut National des Télécommunications - INT, Evry-France, and works in Cluster Computing.