Automated Generic Weak Point Analysis of Business Processes

Jamal Al-Tuwaijari and Aad van Moorsel.
School of Computing Science,
Centre for Cybercrime and Computer Security,
Newcastle University,
Newcastle upon Tyne,
NE1 7RU, UK.
E-mail: {j.m.a.al-tuwaijari, aad.vanmoorsel}@ncl.ac.uk

***Abstract*—Business processes describe the workflow of the critical administrative and business functions in an enterprise. Successful completion of a business process is a basic factor to be considered in business process analysis. In this paper we generalise weak point analysis, which has been proposed to determine the critical components for the successful completion of a business process. The existing work presented in [4] requires the modeller to identify the frequency with which components in the workflow are used, which is not automated and may require considerable computational effort. In the work presented here, we generalise weak-point analysis, such that it can be automatically conducted for any business process specified in BPEL, the Business Process Execution Language. Our methodology annotates a Petri Net model derived from the formal description of BPEL business processes with resource information, automating weak point analysis from that point onwards. We present a case study and initial tool support using SPNP and Matlab.**

***Index Terms*—Business Process, Availability, Stochastic Petri Net, BPEL, Tools.**

I. Introduction

Providing precise evaluation of business processes is important aspect for organizations in order to allow IT managers to balance the operational and economic cost of business processes and the used resources. Availability of resources is therefore a key metrics for business processes [1]. Delivering the right level of availability of the IT infrastructure allows an organisation to balance the reliability of the execution of a business process with the cost of resources [2]. Exceeding the right level of availability results in expensive services, while insufficient availability results in costly outages. For these reasons an organization would like to be able to predict the reliability of business process by distributed resources [3].

In this paper we present a business process analysis methodology to compute the availability for workflows. In [4] the authors propose weak point analysis, an approach to determine the availability bottlenecks in a system. the problem of the approach described in [4] is that the frequency with which resources are used is input to the analysis. That is, the analyst needs to identify by hand how often a business process uses certain resources. We improve on this situation by deriving the usage patterns of resources from the business process definition itself, automatically.

Our vision is to be able to automate weak point analysis for any business process specified in BPEL (Business Process Execution Language). Our approach translates BPEL to a Petri net model (as given in [5], [6]) and annotates the Petri Net model with resource information. Using existing tool support we compute from the Petri net how often resources are used based on the throughput of Petri net transitions. Note that in [4] resource usage has to be known by the analyst. In the current paper we present a tool implementation that assumes a stochastic Petri net model of the BPEL process is provided, using SPNP (Stochastic Petri Net Package) to determine the throughput of transitions. The throughput results are fed into Matlab for the actual weak point analysis. Future work will further expand on this initial tool implementation, including the translation from BPEL processes to Petri nets. This paper presents the overall framework, initial tool support and an example.

The rest of the paper is arranged as follows: In Section 2 we introduce the related works. In Section 3 we present a background and a brief explanation about Weak-point analysis methodology, Petri nets and SPNP and Business Process Execution Language. In Section 4 we introduce the framework and detail of our BPEL based weak-point analysis methodology. Section 5 shows experiment example to depict our analysis methodology. Finally, concluding remarks are presented in Section 6.

II. Related Work

Providing a methodology and framework to assess business process availability is essential for any organization to evaluate and improve its mission and gain high profits. There exists a considerable amount of research on availability analysis of business processes. This research attempts to evaluate the business process quantitatively or qualitatively. Our interest is in the quantitative evaluation; Lei et al [4] proposed a model to optimise resources redundancy to meet availability requirements. This work is important for our work, but, Lie et al does not provide a generic model that can represent a large class of business processes, and only partially automates the calculation of availability based on the business process specification.

Fenz et al [7] propose a model to evaluate the importance of business process resources. They focus on the availability. The limitation is that the model ignores the time factor, which is an important factor and should be considered when availability or performance is evaluated.

Van der Aalst et al [8] presents a product/data model, which provides a methodology to automatically calculate the value of the process elements. The model structure consists of various paths with nodes and costs, flow time, probability, and constraints, these can be quantitatively defined to provide cost or flow time, on the paths to provide the top level product.

The IBM WebSphere development group [2] has proposed work on availability in the enterprise IT resources. They used the Excel spreadsheet for plan and design for the availability solution in the enterprise.

In our work (model) we represent business processes defined in Business Process Execution Language (BPEL) [9] using formal Petri Net specification. Petri Nets have already been considered as a modelling tool for workflow and workflow systems [10], which form the basis of all business processes definition languages, including BPEL.

III. Background

A. Weak Point Analysis

In business process applications, the availability of a single resource affects the overall availability of the business process. We use the term resource for any hardware, software or human element needed to execute the workflow. The weak-point analysis methodology presented in [4] determines the components of a workflow satisfy a predefined business availability requirement level. In this methodology, the modeller first extracts the relevant information from the business process workflows, and takes the business workflows specified in BPEL as input plus the availability requirement for each workflow. From the workflow specification and the resources involved in each workflow the modeller determines how often resources are used, by specifying the workflow mapping matrix. Based on the mapping matrix, plus the availability numbers for resources, it calculates whether the specified availability level has been reached.

To explain the analysis methodology mathematically, suppose there exits n workflows, $W_1, W_2, \cdots, W_n$, and the availability requirement for each workflow is $P_1, P_2, \cdots, P_n$. We also suppose that there are m resources, $C_1, C_2, \cdots, C_m$, and the availability for each resource is $P(C1), P(C2), \cdots, P(Cm)$. The relationship mapping matrix between workflow and resource is depicted as follow: the relationship between workflow W_i and resource C_j is $R_{i,j}$, where $R_{i,j}$ is an integer value depicting the number of references to resource C_i from workflow W_i in the workflow-resource relationship matrix ($R_{i,j}$ set to 1 for referenced resources and 0 for unreferenced resources). The current availability for each workflow is calculated according to the formula:

$$P(W_i) = \prod_{j=1}^{m} (P(C_j)^{R_{i,j}}), \qquad (1)$$

where, $P(W_i)$ is the current availability for workflow W_i, $P(C_j)$ is the availability of resource C_j. The current availability for workflow $P(W_i)$ compared with the workflows availability requirement P_i: if $P(W_i) \geqslant P_i$, the availability requirement is satisfied; otherwise, , it is not satisfied and some resources involved in workflow W_i need to be replaced or otherwise enhanced. As the methodology determines the workflows and the relevant resources which do not meet the required availability, it is called ”weak point analysis”.

B. Petri Nets

There has been an increasing interest in using Petri nets as a modelling language for workflows and workflow based applications [11, 10]. This is because Petri nets have a strong mathematical foundation formalism makes it possible to set up mathematical models describing the behaviour of the system. In addition, Petri nets allow a graphical representation to ease the understanding of the modelled system. Petri nets have been supported by many tools and applications make it suitable for many application areas such as workflow management systems [12, 13], business process management [14, 15] and web service technology [16, 17].

A Petri nets as defined in [18] is a 5-tuple, $N = (P, T, F, W, M_0)$, where: P is a finite set of places, T is a finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a set of flow relations,$W : F \rightarrow \mathbb{N}$ is a weight function, $M_0 : P \rightarrow \mathbb{N}_0$ is the initial marking. Where $\mathbb{N}$ is the set of natural numbers and $\mathbb{N}_0$ denotes $\mathbb{N} \cup \{0\}$.

The relation between places and transition are: $P \cap T = \phi \wedge P \cup T \neq \phi$. Graphically, places are depicted as circles and transitions are depicted as either bars or boxes. In a workflow application, a Petri net model can be use the transitions to represent the tasks and places represent the pre and post conditions of the tasks that involve in the system. Transition is enabled if each input place contains at least a number of tokens equal the weight of the flow relation from the places to the transition; and when transition is firing it consumes a number of tokens from each input place and produces a number of tokens equal to the weight of the flow relation from place to the transition. The arcs connect places with transitions or vice versa.

A Petri nets have many extensions such as, Stochastic Petri nets and its extension generalized Stochastic Petri nets [19], it have been used as a useful modelling formalism and proposed for modelling qualitative and quantitative analysis of systems [10]. Stochastic Petri nets are timed (transition) and the delay of transition firing are random variables with exponential distribution. This means; the transition is associated with a random firing delay whose probability density functions (pdf) are a negative exponential with specific rate. In GSPNs, immediate and timed transitions are coexisted. To analysis GSPTs, the reachability graph generated, i.e. all possible markings and transitions between markings in GSPNs. This allowed obtaining the corresponding Continuous Time Markov Chains (CTMC), in order to analysis and computing the interesting performance indices [20]. Moreover, there are several tools that support the evaluation of SPNs such as SPNP [21] which are used as a tools support for analysis and evaluate our model in this paper.

1) SPNP: The Stochastic Petri Net Package (SPNP) is a modelling tool used for performance, dependability and performability analysis of complex systems [22]. CSPL(C-based SPN Language) which is an extension of C programming language with some facilitate used as input language for SPNP. It is a Stochastic Reward Nets (SRNs) based on the Markov Reward Models (MRM) and provides modelling environment for: dependability (reliability, availability, safety) analysis, performance analysis and performability modelling. SPNP facilitate the construction of modelling complex system through utilization of essential Petri nets constructs such as marking dependency, enabling functions and variable cardinality arc. The SRN solved for steady-state metrics or transient metrics. It allows for standard set of measures as well as custom measures. The important features that had supported by SRN model are: marking dependent arc multiplicities, enabling function firing probabilities and firing rates.

IV. Our Approach

The important aspect of our approach is that we automatically determine the frequency with which resources are being used based on the business process specification. This considerably improves on the work in [4] where the modeller needs to input resources usage by hand. This leads to inaccuracies and to extra work. If the business process specification is available, it should be beneficial to rely on this definition to determine the usage patterns for resources.

The generic methodology we propose in this paper uses the BPEL specification of a business process, and translates this into Petri nets. Resources are associated to the relevant transitions of Petri Net model. From this, we automatically extract the transition-resource relationship which determines the number of times the involved resources used in the relevant workflows. The main framework of our weak-point analysis methodology is shown in Fig. 1. The framework consists of the following modules:

A. The BPEL Business Process Specification Module

The business processes in BPEL specify the flow of process at business process level in BPEL files structure. The structure of BPEL file specifies the basic building blocks that used to defining the process and services implemented in business process workflows. The basic workflow building blocks are shown in Fig. 2; they include: the sequential routing, parallel routing, conditional routing and iterative routing [23].

These basic building blocks are with high importance and to be considering when construct the Petri nets model in the next module.

B. The Petri Nets and SPNP Module

The extracted Petri Nets model represent the definition of BPEL business process workflows which specifies the flow of process as depicted in the BPEL structure files. The importance of this module lays in extracting the relationship between transitions and relevant resources involved in business process workflows. The mapping relationship transition-resource specifies the necessary information for the relevant resources for each transition at the Petri Nets model. The relationship determines number of times that the resource is used to complete transition firing per unit of time. Therefore, the firing rule which indicates under which conditions a transition may fire, and what the effect of the firing[18], is on the marking is an important aspect to determine how often does transition fire per unit time, and how long the relevant resource is used to complete transition firing. This is done by the use of SPNP as a powerful tool for analysis a Petri Nets model. Our important interested is the measuring of throughput for each transition of model as the transition-resource relationship is the key contribution of our of Petri nets model. A transition T is said to be enabled if each input place P of t is marked with at least $w(P,T)$ is the weight of the arc from P to T. In CSPL file we use the rate rewards to compute the average throughput $E[TR]$ for transition T in two ways: The sequences, AND-split, OR-split, and OR-join transitions are computed according to the formula:

$$E[TR] = \sum_{X=1}^{N} (P_r(\#P_1 \geqslant X) * X) * \lambda \tag{2}$$

where, P_r is the probability of marking place P_1, X is an integer number and λ is the rate of the transition T. The AND-join transition is computed according to the formula:

$$\begin{aligned} E[TR] &= \sum_{X=1}^{N} (P_r(\#P_1 = X \&\& \#P_2 \geqslant X \cdots \\ &\quad \#P_m \geqslant X) * X) * \lambda \end{aligned} \tag{3}$$

where, P_r is the probability of marking places $P_1, P_2 \cdots P_m$ which are preceding the AND-join transition. Now we have computed the throughput of each transition and determined the number of times that each transition T_i fires per unit time. This is the basis for transition-resource relationship matrix which will be used in the next module.

C. The Weak-Point Analysis Module

This module carries out the weak-point analysis based on the mapping transition-resource relation matrix derived from the Petri net model. Now we can calculate the current availability for each work flow according to its involved resource. We assume there exist n business process workflows denoted by $W_1, W_2, W_3, \cdots, W_n$, and the availability requirements for each workflows are specified with $P_1, P_2, P_3, \cdots, P_n$, where $0 < P_n < 1$. We also assume that there are m resources available for workflows denoted by $r_1, r_2, r_3, \cdots, r_m$. For a

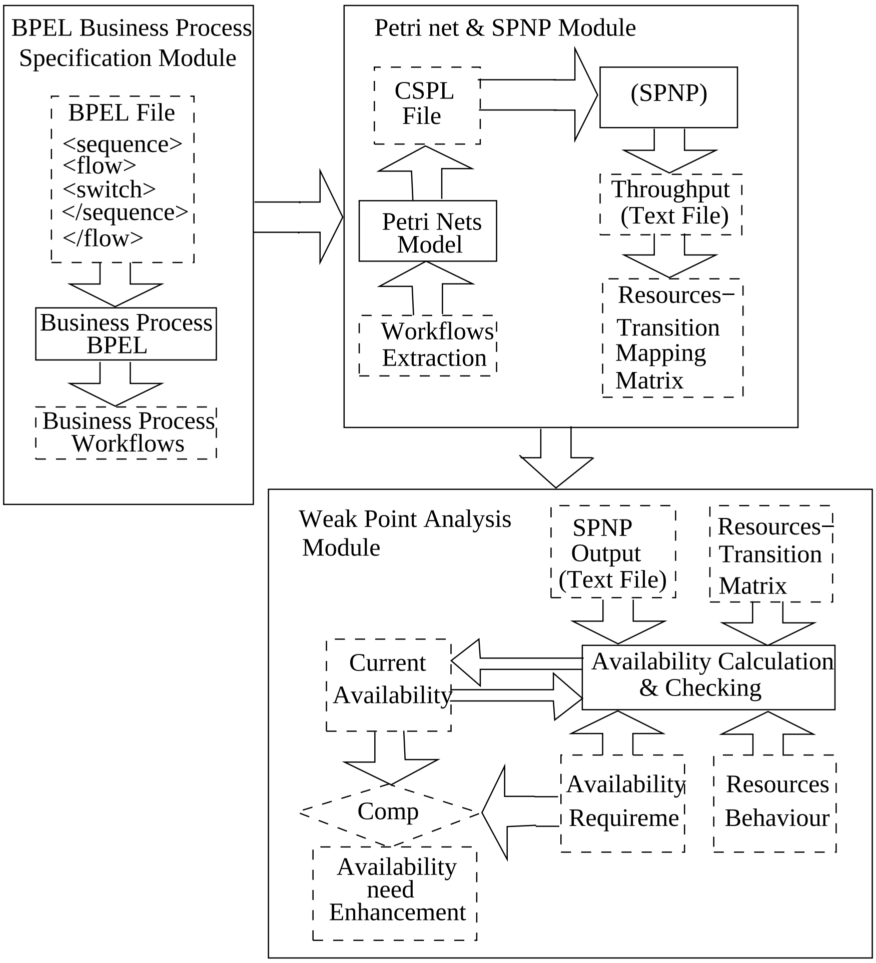

Fig. 1: Framework for weak-point analysis.

Petri Net model derived from the formal description of BPEL business processes, we assume that there are k transitions denoted by $t_1, t_2, t_3, \cdots, t_k$. We construct a matrix to capture the transition-resource relationship. Table I shows the matrix; the relationship between transition ti and resource r_j is $R_{i,j}$, where $R_{i,j}$ is the value of throughput of transition t_i references to resource r_j and depicted the number of times that resource r_j is used in relevant t_i transition, and set to 0 when resource is not included in the transition-resource matrix. For example, let the Petri nets model consist of four transitions t_1, t_2, t_3 and t_4, which are mapped to five IT resources r_1, r_2, r_3, r_4 and r_5. The throughput of each transition is R_1, R_2, R_3 and R_4 respectively and the availability of each resource is $P(r_1), P(r_2), P(r_3), P(r_4)$ and $P(r_5)$ respectively. The transition-resource relationship matrix is $(R_{1,1}, R_{2,3}, R_{3,4}, R_{4,5})$, which mean that resource 2 is not include in the resource list of transitions; thus the availability for the workflow is :

$$P(W_i) = P(r_1)^{R_{1,1}} * P(r_3)^{R_{2,3}} * P(r_4)^{R_{3,4}} * P(r_5)^{R_{4,5}} \quad (4)$$

$R_{i,j}$ is set to 0 for unreferenced resources.

We compare the current availability for workflow $P(W_i)$ with the workflows availability requirement P_{req}: if $P(W_i) \geqslant P_{req}$, the requirement availability is meet; otherwise, the availability requirement is not satisfied, and this indicates that some resources in the transition-resource relationship is need their availability enhanced to meet the availability requirement. Which resources should be enhanced for availability depends on the resource components behaviour and cost.

TABLE I: The transition-resource relationship matrix.

	r_1	r_2	r_3	$\cdots$	r_m
t_1	$R_{1,1}$	$R_{1,2}$	$R_{1,3}$	$\cdots$	$R_{1,m}$
t_2	$R_{2,1}$	$R_{2,2}$	$R_{2,3}$	$\cdots$	$R_{2,m}$
t_3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$	$\cdots$	$R_{3,m}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
t_k	$R_{k,1}$	$R_{k,2}$	$R_{k,3}$	$\cdots$	$R_{k,m}$

V. Case Study

A. Example

In this section, we give a simple example to depict our methodology. The definition of our BPEL business process example is shown in Fig. 3, which is a BPEL business travel scenario [24] comprising the following partners: client web

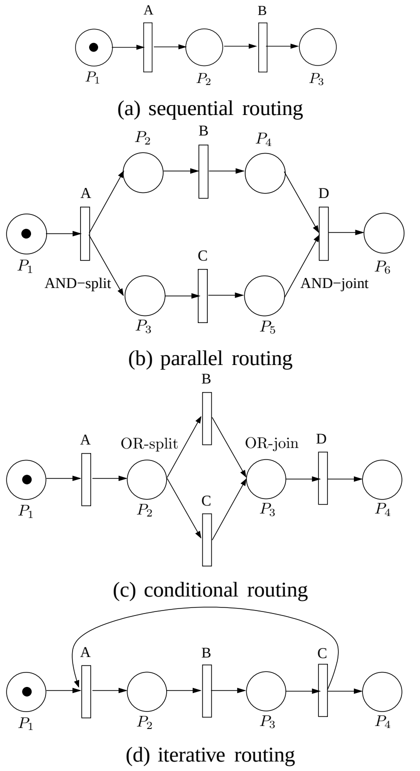

Fig. 2: Basic workflow building blocks

service, employee travel status web service, and two airline web services: American and Delta Airlines (Fig. 4).

A Petri nets model that we have derived from the formal specifications of BPEL business processes specified the order of activities constitute the BPEL business processes (Fig. 5). The model consists of ten places and seven transitions (four timed transitions and three immediate transitions). Timed transitions are associated with random exponentially distributed firing delays; while, immediate transitions are fire in zero time. The places are: P_{cli_av}, $P_{req}, P_{comp1}, P_{comp2}, P_{empl1}, P_{empl2}, P_{prc1}, P_{prc2}, P_{sel1}$ and P_{sel2}. The timed transitions are: $T_{client}, T_{prc1}, T_{prc2}$ and T_{best}. Finally, the immediate transitions are t_0, t_1 and t_2. The resources are associated to the timed transitions to capture the transition-resource relation matrix.

Using SPNP as a powerful tool for solution of Stochastic Petri nets models, we compute the throughput of timed transitions $T_{client}, T_{prc1}, T_{prc2}$ in the CSPL file according to equation (2), and the throughput of timed transitions T_{best} according to equation (3).

There are many states for our Petri nets model. The state of a Petri nets model depends on the initial marking of the model (number of tokens jobs in the initial places) which determines the number of jobs to be executed per unit time. The number of tokens in the Petri nets places represented by symbols that are parameters of the model. An initial marking with one or more parameters represents markings that can be obtained by assigning different legal values to the parameters. Accordingly,

```
<sequence>
    <receive partnerlink="client"
            portType="trv:TravelApprovalPT"
            operation="TravelApproval"
            variable="TravelRequest"
            createInstance="yes"/>
    <assign>
        <copy>
            <from variable="TravelRequest"
                    part="employee"/>
            <to variable="EmployeeTravelStatusRequest"
                        part="employee"/>
        </copy>
    </asign>
      <invoke partnerLink="employeeTravelStatus"
              portType="emp:EmployeeTravelStatusPT"
              operation="EmployeeTravelStatus"
          inputVariable="EmployeeTravelStatusRequest"
          oputputVariable="employeeTravelStatusResponse"/>
    <asign>
        <copy>
          <from variable="TravelRequest"
          operation="TravelApproval"
          <to variable="FlightDetails"
                      part="flightData"/>
         </copy>
         <copy>
          <from variable="EmployeeTravelStatusResponse"
                 part="travelClass"/>
          <to variable="FlightDetails"
                 part="travelClass"/>
         </copy>
    </asign>
    <flow>
       <sequence>
            <invoke partnerLink="MMAirlines"
                    portType="alnFlightAvailabilityPT"
                    operation="FlightAvailability"
                    inputVariable="FlightDetails"/>
            <receive partnerlink="MMAirline"
                    portType="alnFlightCallbackPT"
                   operation="FlightTicketCallback"
                   variable="FlightResponseMM"
       </sequence>
       <sequence>
            <invoke partnerLink="NNAirlines"
                    portType="alnFlightAvailabilityPT"
                    operation="FlightAvailability"
                    inputVariable="FlightDetails"/>
           <receive partnerlink="NNAirline"
                     portType="alnFlightCallbackPT"
                    operation="FlightTicketCallback"
                    variable="FlightResponseNN"
       </sequence>
    </flow>
    <switch>
       <case condition="bpws:getVariableData('FlightResponseMM','confirmationData',
        '/confirmationData/price')
        &It:=bpws:getvariableData('FlightResponseNN',confirmationDAta',
        '/confirmationData/price')
    <asign>
       <copy><from variable="FlightResponseMM"/>
             <to variable="TravelResponse"/>
       </copy>
    </asign>
  </case>
  <otherwise>
     <asign>
        <copy>
            <from variable="FlightResponseNN"/>
            <to variable="TravelResponse"/>
        </copy>
    </asign>
  </otherwise>
 </switch>
       <invoke partnerLink="client"
               portType="trv:ClientCallbackPT"
               operation="ClientCallback"
               inputVariable="TravelResponse"/>
 </sequence>
</process>
```

Fig. 3: BPEL Business Process Travels Example.

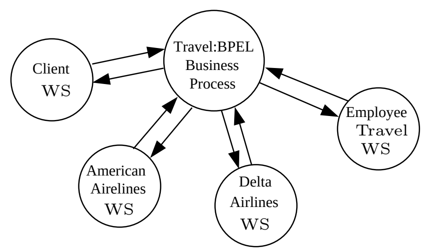

Fig. 4: Business process parties.

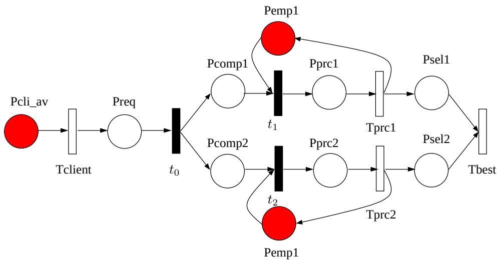

Fig. 5: Graphical representation of Petri nets model.

```
NET
=======================================
======================
        discrete places: 10
        immediate transitions: 3
        time transitions: 4
        constant input arcs: 6
        constant output arcs: 5
        constant inhibitor arcs: 0
        variable input arcs: 4
        variable output arcs: 5
        variable inhibitor arcs: 0
=======================================
======================
RG:
=======================================
======================
        tangible markings: 204
        vanishing markings: 0
        marking−to−marking transitions: 616
=======================================
======================
=======================================
======================
                        TIME: INFINITY
=======================================
======================
EXPECTED: Throughput of the transitions
Tclient in steady−state= 4.3930226041
EXPECTED: Throughput of the transition Tprcl
in steady−state = 3.26733175852
EXPECTED: Throughput of the transition Tprc2
in steady−state = 3.67242819116
EXPECTED: Throughput of the transition Tbest
in steady−state = 1.34755162718
```

Fig. 6: SPNP output text file.

TABLE II: Resource components.

	c_1	c_2	c_3	$\cdots$	c_j
r_1	$K_{1,1}$	$K_{1,2}$	$K_{1,3}$	$\cdots$	$K_{1,j}$
r_2	$K_{2,1}$	$K_{2,2}$	$K_{2,3}$	$\cdots$	$K_{2,j}$
r_3	$K_{3,1}$	$K_{3,2}$	$K_{3,3}$	$\cdots$	$K_{3,j}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
r_n	$K_{n,1}$	$K_{n,2}$	$K_{n,3}$	$\cdots$	$K_{n,j}$

TABLE III: Component failure and costs.

Component	ColdCost	ActivCost	RepairCost	MTTR	MTTF
c_1	cc_1	ac_1	rc_1	MTTR_1	MTTF_1
c_2	cc_2	ac_2	rc_2	MTTR_2	MTTF_2
c_3	cc_3	ac_3	rc_3	MTTR_3	MTTF_3
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
c_j	cc_j	ac_j	rc_j	MTTR_j	MTTF_j

```
The availability of Resources
AVRS =
   0.99974833144886
   0.99989732491491
   0.99975528000621
   0.99975189206566
The Throughput of Transitions from SPNP
new3 =
   4.39302260410000
   3.26733175852000
   3.67242819116000
   1.34755162718000

The Transition−Resources Relationship
TTR =
   0   0   1   0
   1   0   0   1
   0   1   1   0
   1   0   1   1
The Availability of Workflow
AVWF =
   0.99502480103268
```

Fig. 7: MATLAB output results.

we have the following initial states:

- One token (job)
- One token, continuously cycling
- Multiple tokens

In each of above state the initial marking of the place P_{cli_av} determine the state of our model depending on the value of the parameter $client\_no$ which represent the number of token in the initial place. For each of above state SPNP produce a different output text file with different results for computing the throughput of each transition of the model. In our model we concentrated on the computing of throughput of transitions as our aim is to associate each transition to the relevant resource result to capture the resource-transition relationship. To explain our experimental example we have executed the first state (one token job) by give the parameter $client\_no$ the value one on the Petri net model. We have obtained the output text file with the results shown in (Fig.6.):

The current availability for resources is calculated in our scenario with respect to the components of resource as depicted in Table II, and the component failure behaviour and cost depicted in Table III, which consist the fundamental parameters for resources availability. The availability of single component calculated as MTTF / (MTTF + MTTR), where MTTF specifies the mean time to failures and MTTR specifies the mean time to repair after each failure. Given P_{rm} the availability of resources and the transition-resource relationship matrix we can calculate the availability of workflows in our methodology according to the formula:

$$P(W_i) = \prod_{j=1}^{m} (P_{rj}^{R_{i,j}}) \tag{5}$$

Where P_{rj} the availability of resource j and $R_{i,j}$ is the value of throughput of transition t_i references to resource r_j in transition-resource matrix. For the solution of our methodology we use MATLAB [25] to perform the calculations. Fig. 7 shows the final MATLAB output result file.

B. Tools Implementation

The robust of our approach is come from the used of the powerful tools, Stochastic Petri Net Package SPNP and MATLAB. Our current tool integrates SPNP with MATLAB: first we used SPNP as a tool determines resource usage through computing the throughput of the transitions of our Petri net model as that the basic to capture the resource-transition relationship, and then the obtained result of the transition throughput feed into MATLAB program to calculate the availability of resources and the overall workflow result to conducts the weak-point analysis .

VI. CONCLUSION

Business processes are widely used as a structured flow of business activities to manage workflows and the use of resources. Availability is a basic and important metric to be considered for business process analysis. In this paper we have automated weak-point analysis, and approach to determine the availability and bottlenecks in business processes. We use stochastic Petri nets as a modelling tool to represent the workflow specified in BPEL and map resources on firing of transitions of the Petri net model. Computation of the throughput using standard stochastic Petri net tools (SPNP in our case) determines the frequency with which resources are used. Our current tool integrates SPNP with Matlab: first SPNP determines resource usage through computing the throughput of a transition, and then Matlab conducts the weak-point analysis. Future work will provide an integrated tool, automating all aspects of weak-point analysis, including the translation from BPEL to Petri nets and the analysis of the bottleneck.

REFERENCES

[1] F. Kobylanski, B. Lundhild, R. Mau, J. Meeks, et al, "Oracle Database High Availability Overview," 11g Release 2007.

[2] R. Robinson and A. Polozoff, "Planning for Availability in the Enterprise IBM Software Services for WebSphere," *IBM WebSphere Developr Technical Journal*, 2003.

[3] M. Kamath and G. Alonso, "Providing High Availability in Very Large Workflow Management Systems," *The Fifth International Conference on Extending Database Technology*, (EDBT 96).

[4] L. Xie, J. Luo, J. Qiu, J. A. Pershing, Y. Li, and Y. Chen, " Availability weak point" analysis over a SOA deployment framework," *In NOMS IEEE*, 2008.

[5] C. Ouyang, E. Verbeek, W. M. P. van der Aalst, S. Breutel, M. Dumas and A. H. M. Hofsted, " Formal Semantics and Analysis of Control Flow in WS-BPEL," *Technical report (revised version), Queensland University of Technology*, 2005.

[6] C. Stahl, "A Petri net semantics for BPEL," *Technical Report 188, Humboldt-Universitat zu Berlin*, 2005.

[7] S. Fenz, A. Ekelhart, and T. Neubauer, "Business Process-based Resource Importance Determination," *Proceedings of the 7th International Conference on Business Process Management (BPM'2009)*, 2009.

[8] W. M. P. van der Aalst, H. A. Reijers and S. Limam, "Product-based workflow design," *Journal of Management Information Systems*, vol 20, issue 1, pp 229-262, 2003.

[9] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic and S. Weerawarana, "Business Process Execution Language for Web Services Version 1.1," *Standards proposal by BEA Systems, International Business Machines Corporation and Microsoft Corporation*, 2003.

[10] W. M. P. van der Aalst, "The application of Petri Nets to Workflow Management," *The journal of Circuits Systems and Computers*, vol 8, issue 1, pp 21-66, 1998.

[11] N. R. Adam, V. Atluri and W. K. Huang, "Modelling and analysis of workflows using Petri nets," *Journal of Intelligent Information Systems*, vol 10, pp 131-158, 1998.

[12] K. Salimifard and M. Wright, "Petri net-based modelling of workflow systems: An Overview," *European Journal of Operational Research*, vol 134, issue 3, pp 664-676, 2001.

[13] V. Atluri and W. K. Huang, "A Petri net based safety analysis of workflow authorization models," *The Journal of Computer Security*, vol 8, issue 3, pp 209-240, 2000.

[14] K. Sarshar, T. Theling, P. Loos and M. Jerrentrup, "Integrating process and organization models of collaborations through object Petri nets," *In Proc. of the Third GI-Workshop XML4BPM XML Integration and Transformation For Business Process Management*, 2006.

[15] J. Dehnert and P. Rittgen, "Relaxed soundness of business processes," *In Proc. (CAiSE) Of the 13th International Conference on Advanced Information Systems Engineering*, 2001.

[16] H. Schlingloff, A. Martens and K. Schmidt, "Modelling and model checking Webservices," *Electronic Notes in Theoretical Computer Science*, vol 126, Issue on Logic and Communication in Multi-Agent Systems, pp 3-26, 2005.

[17] Y. Tang, L. Chen, K. T. He and N. J. Srn, "An extended Petri-net-based workflow model for web Service composition," *In Proc. of the IEEE International Conference on Web Services*, 2004.

[18] T. Murata, "Petri Nets: Properties, Analysis and Applications ," *Proceedings of the IEEE*, vol 77, issue 4, pp 541-580, 1989.

[19] M. Ajmone Marsan, G. Balbo and G. Conte, "A class of generalized stochastic Petri nets for the performance analysis of multiprocessor systems," *ACM Transactions on Computer Systems*, vol 2, issue 1, pp 93-122, 1984.

[20] M. Ajmone-Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis, "*Modelling with Generalized Stochastic Petri Nets*," Wiley Series on Parallel Computing, 1995.

[21] G. Ciardo, J. K. Muppala and K. S. Trivedi, "SPNP: Stochastic Petri Net Package," 151, 1989.

[22] K. S. Trivedi et al, "SPNP Users Manual, Version 6.0," *(http://www.ee.duke.edu/ ~ kst/software_ packages.html)*.

[23] N. Russell, A. ter Hofstede, W. M. P. van der Aalst and N. Mulyar, "Workow Control- Flow Patterns: A Revised View," *Technical report, BPMcenter.org* , 2006.

[24] M. B. Juric, B. Mathew and P. Sarang, "Business process Execution Language for Web Services," 1st ed, 2004.

[25] MATLAB, "The Language of Technical Computing, " *(http://www.mathworks.com/products/ matlab/)*.