Petri Net Workflow Modeling for Digital Publishing Measuring Quantitative Dependability Attributes

Gustavo A. Chaparro-Baquero, Nayda G. Santiago, Wilson Rivera, J. Fernando Vega-Riveros
University of Puerto Rico, Mayagüez Campus
Department of Electrical and
Computer Engineering
P.O. Box 9042, Mayagüez, PR, U.S.A. 00681
{gustavo.chaparro, nayda.santiago, wrivera, fvega}@ece.uprm.edu

Abstract

This work describes the concept of workflow modeling using Workflow-nets and Generalized Stochastic Petri Nets (GSPN) for the Digital Publishing business process and how the attributes of dependability are measured in a quantitative form. In our novel approach, these are measured from the workflow model itself, improving the analysis of a workflow model. Applying these measure concepts to the general Digital Publishing pre-press process provides a better workflow management in this area since this process is based on its trustworthiness. Once the methodology for workflow modeling is introduced, the results for a case study on the preflight stage of the Digital Publishing workflow are presented.

1 Introduction

Digital publishing (DP) allows the linking of printing presses to computers offering the potential to raise the quality level for short-run printing. However, the realization of this potential has been seriously hampered by a number of difficulties including the management of faults and errors of incoming jobs. The pre-press process in DP consists of different job treatment stages involving correct set up of each job in order to be printed, which in turn means to decrease the number of faults and errors to a minimum level. The typical pre-press stages in a DP workflow are described in Table 1 [4].

Table 1. Pre-press process stages in DP

Stage	Description
Intent	Track the document specifications provided by the client, like type of job , tolerance of quality and due date of the job.
Preflight	Check if the digital document has all the elements required to perform well in the production workflow.
Trapping	Overlap colors to compensate press registration. Register is the accurate positioning of two or more colors of ink in a printed sheet.
Imposition	Arrange individual pages on a press sheet, so that when they are folded and trimmed, the pages are in the correct orientation and order.
Proofing	Check physically if there are faults remaining in the job verifying the output before being printed.
Ripping	Decode Postscript, create an intermediate list of objects and instructions, and finally convert graphic elements into bitmaps for rendering on an output device.

Even though there are software tools that work on each stage in the pre-press process for DP, these individual packages cannot guarantee that a print job is correctly completed. There are some packages available aimed at the managing of a DP workflow, however, those packages cannot guarantee the automation of the entire process with an acceptable level of reliability. The integration of the processes into a production workflow or a supply chain path reduces the costs, increases productivity, and serves customers better [7].

Creating a workflow process definition of the DP pre-press process improves the global process itself and consequently its trustworthiness. Besides, if the system needs to be critically trustable or if its failures are decreasing the throughput of the system, it is useful to analyze its dependability. Measuring dependability quantitatively aids in the analysis of the behavior of the system in the presence of faults and estimates which parameters provide the system with a higher trustworthiness. Performance is useful to characterize the system and its throughput, but quantitative measures of dependability shows the probabilistic estimates of the future incidence of the faults. This measure helps justify the functional specifications that the system has to meet.

To create a workflow process definition, it is necessary to know the business process definition to be modeled, to map it to a workflow model. From the final resulting workflow model we may analyze the dependability of the system, measuring quantitatively attributes of dependability such as reliability, maintainability, availability or safety.

This work describes the concept of workflow modeling using Petri Nets for the DP business process and how the attributes of dependability are measured in a quantitative form. We also propose a methodology for measuring dependability from a workflow process definition. The paper is organized as follows. Section 2 provides background related to Petri Nets, workflow modeling and dependability. Section 3 presents a DP workflow based on the informal description of the business process. Section 4 describes the application of an alternative methodology to measure dependability for DP, and the use of Generalized Stochastic Petri Nets for the analysis of workflow model characteristics. Attributes of dependability are measured in a quantitative form, from the workflow model itself to improve modeling. Finally we point out a comparison between results from both methodologies, and concluding remarks in sections 5, and 6, respectively.

2 Preliminary concepts

There are three basic concepts involved in this work: *Petri Nets*, *modeling of workflows*, and *dependability*. In the following paragraphs these concepts are described.

2.1 Petri Nets

A Petri Net (PN) is a five-tuple (*P*, *T*, *I*, *O*, *MP*) where *P* represents a set of places, *P* = $\{p_1, p_2, ..., p_n\}$, with one place for each circle in the Petri Net graph; *T* represents a set of transitions, *T* = $\{t_1, t_2, ..., t_m\}$, with one for each bar in the Petri Net graph; *I* represents an input function that defines directed arcs from places to transitions; *O* represents an output function that defines directed arcs from transitions to places; and finally *MP* represents the marking of places with tokens. Tokens are represented as small dots or integer numbers and the diminution of tokens over the places determine the state of a Petri Net.

Petri Nets provide a uniform environment for modeling, formal analysis, and design of discrete event systems. Petri Nets models are used for the analysis of behavioral properties and performance evaluation, as well as for systematic construction of discrete-event simulators [11]. Digital Printing involves a combination of separate stages that manipulates a job in order to ensure its printability and its correct delivery to the client. The arrival, manipulation and print out of those jobs are discrete events [4]. For that reason Digital Publishing is modeled as a discrete event system.

Petri nets have evolved to incorporate more detailed techniques for modeling. Those techniques have been called *Extensions*.

2.1.1 Petri Net Extensions

One of those extensions is called *High level Petri nets*. These nets involve three extensions useful for describing workflow models and mapping business process to them: Colored Petri Nets, Hierarchical Petri Nets, and Petri Nets with time. These associate time whether to a variable of time carried by each token or to a firing delay in transitions.

2.1.2 Timed Petri Nets

To study performance and dependability issues of systems it is necessary to include a timing concept into the model, because an ordinary PN only describes the structure of the model, but performance and dependability analysis involves also time evolution study. There are several possibilities to do this for a Petri net. However, the most common way is to associate a firing delay with each transition. This delay specifies the time that the transition has to be enabled, before it can actually fire. If the delay follows a random distribution function, the resulting net class is called *stochastic Petri net*. Different types of transitions can be distinguished depending on their associated delay. These include immediate transitions, exponential transitions, and deterministic transitions.

Stochastic Petri Nets A *Stochastic Petri Net* (SPN) has associated a firing delay to all of its transition, and this delay of time is associated with a random variable exponentially distributed. This means that the distribution of the random variable X_i of the firing time of a transition is given by $F_{X_i}(X) = 1 - e^{-\lambda_i \cdot X}$. The average time of firing of the transition t_i is $\frac{1}{\lambda_i}$. The quantitative analysis of a SPN is made analyzing the corresponding Markovian process. This is done by adding to each arc of the reachability graph, a weight equivalent to the exponential distribution rate (λ_i) of each transition firing. This results in obtaining a Markov chain from the SPN [2]. Achieving the steady state distribution of the Markov chain, is possible to compute performance measures like the probability of being in a subset of markings, the mean number of tokens and the probability of firing any transition.

Generalized Stochastic Petri Nets Stochastic Petri Nets are helpful for evaluating in terms of probabilities the extent to which some attributes like availability, maintainability, safety and reliability are satisfied into a system [1]. It is not always useful to associate a random variable of time to each firing transition in the net, because either the execution time of this transition is zero (immediate) or this execution time could be approximated to zero. The inclusion of immediate transitions makes it easier the analysis of the net reducing the states that have to be computed. A Petri Net that involves exponentially distributed transitions and immediate transitions is called a *Generalized Stochastic Petri Net* (GSPN) [2].

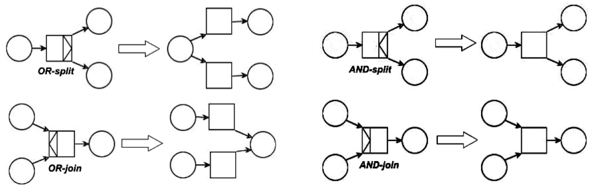
2.2 Workflow Modeling

Workflow is referred to the study of operational aspects of a specific activity in a workable environment. Many classes of PN for workflow modeling have been proposed. One of those classes is the Workflow-net, which is an extension of a PN proposed by Wil van der Aalst [9].

2.2.1 Workflow-nets

A Workflow-net specifies the dynamic behavior of a single case in isolation. It must have a place with no incoming arcs, which identifies the beginning of the process, and a place with no outgoing arcs which identifies the end of the process. A workflow-net must be strongly connected, which means that any node can be reached from the starting place following a certain path.

The theory of Workflow-nets has additional classes of transitions that aids to clarify the routing rules described by the workflow model. These transitions are AND-split, AND-join, OR-split and OR-join, and they are shown in the Fig. 1 along with their corresponding PN meaning.



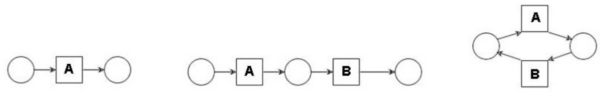
(a) (b)

Figure 1. Additional transitions of a Workflow-net and their corresponding PN meaning

With an *AND-Split* a token must be produced for each of the output places under all circumstances and with an *AND-join* the task can only take place once there is a token at each of the input places. With an *OR-split* a token must be produced for just one of the output places and a decision rule must be adopted to solve the corresponding firing conflict. Finally, with an *OR-join* the task take place once the single token reach one of the input places.

A Workflow-net must be *sound*, which means that it must not have unnecessary tasks and every case treated by the process must not make any reference to this case once the case reach the final state, i.e., remaining tokens must not be leaved in the process.

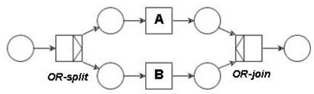
In order to make a Workflow-net sound, its construction should be done with sound processes. The theory identifies four basic constructions for routing tasks, which fulfill with the soundness property. These constructions are described in Fig. 2. It is noteworthy to say that a Workflow-net could be made also using any sound process.



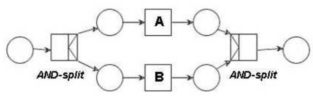
(a) Basic building block

(b) Sequential routing: Task A is executed before Task B

(c) Iterative routing: Task B is repeated



(d) Alternative routing: Either Task A or Task B is executed



(e) Parallel routing: Task A and Task B, both are executed in any order

Figure 2. Description of the basic Constructions for routing tasks

2.3 Measures Taken from a workflow model based on Petri Nets

Workflow modeling and analysis based on PN have been used in many settings in industry. In many opportunities, PN analysis has helped to verify the soundness of the model and repair errors in the PN itself, proving that the implementation of PN in the modeling and analysis of workflow systems provides a standard design method approach [6]. Some of those studies have been concentrated in workflow performance issues [5]. Not only the task and resources are important to be managed in a workflow process, but also the time of completion of those tasks is important. Time management is essential in determining and controlling the life cycle of each activity involved in the business process.

Because the workflow model is the heart of the workflow management system, it must be carefully designed. PN theory have been used only to debug the model itself and to analyze it, which involves adding time variables to the net, and measuring performance issues. It has been used to improve the model in early stages of the model creation. However, the analysis made over a WF-net based on PN reduces to soundness and performance study of the model, but other measures like dependability have not been introduced in that examination of the workflow model.

2.4 Dependability

Dependability is the ability to deliver service that can justifiably be trusted. This concept includes measures such as reliability, availability, maintainability, or safety. There are known tools and techniques for dependability analysis such as: Static and dynamic fault trees, Stochastic Petri Nets, Markov and queuing models, and Reliability block diagrams [1]. The methodology behind dependability focuses on identifying, treating, and classifying the different types of faults, errors, and failures that could be found in a system. Moreover, a failure is when the system stops providing the service it was intended to do. An error is the possible cause of a failure when it reaches the service interface of the system and it could be seen on some parts of the system. Finally, a fault is the hypothetical cause of an error.

The concepts used in dependability map adequately with the digital printing workflow. One of these concepts is acceptability of errors. Errors are acceptable as long as they do not cause problems to the user. Another concept is trustworthiness. For DP, benefits are based on clients trust.

There are four means to attain dependability in a system, each of them used in a different stage of the establishment or design of the system. Those means are: *Fault Prevention*, *Fault Removal*, *Fault Tolerance*, and *Fault Forecasting*. *Fault Prevention* refers to the avoidance of faults in the fists stages of the system, which means system entries with less or no faults. *Fault removal* refers to verify the system looking for faults and correct them. *Fault tolerance* refers to make the system strong enough to detect a fault or an error, and recover from it by itself. *Fault forecasting* refers to the identification and classification of the possible faults and errors that the system could show, which is a qualitative evaluation.

Fault Prevention and Fault Removal for the DP pre-press process are not meaningful to be studied, because it is assumed that each job incoming to the press will contain faults and errors making the reduction of the severity of those faults and errors difficult. For the DP pre-press process fault tolerance could be studied as mean to attain dependability, but that study is out of the scope of this work. This work focuses on obtaining quantitative measures of the attributes of dependability from a workflow process definition, particularly from the workflow process definition of the digital printing pre-press business process.

3 Developing a WF-net model for Digital Publishing

From the informal description of the business process of DP, it is possible to identify each part of the workflow model such as tasks, cases, processes, and routes, and then, construct the workflow model. Thus, this model can be introduced into a workflow engine of a workflow management system. The following paragraphs depict the essential stages in a Digital Printing process [4].

Once the document is tracked in the *Intent* stage, it is checked in *Preflight*. The preflight stage selects the best profile (set of characteristics that will be checked) for the job treatment. Thus, a preflight technician decides based on the report whether a fault is fixable or not. Because of the faults in the document that are not repairable, the job must be sent back to the client. After this stage, the document is submitted to the *Trapping* stage. Then, it is sent to the preliminary *Proofing* stage, where the job is checked physically to see if there are faults remaining on it. If there are such faults, the job is sent back to the preflight stage in order to correct it, otherwise it is sent to the *Imposition* stage. A final proof of accuracy is made after the imposition stage. This proof is a legally binding sample of how the job is expected to appear when printed. Subsequently, the job is sent to the *RIP* or *Ripping* stage. Although the job is supposed to be fault-free in this stage, if a job fails ripping, some procedures could be applied to fix the problem. Nevertheless, if the RIP process definitely fails, the job should be sent to an early stage in order to determine the exact problem cause and correct it.

Based on this description of the system it is possible to map it into a workflow model based on Petri Nets. Dependent upon the informal description of the system, we created the corresponding process, mapping this description of the system into a workflow model based on Petri Nets. The result is shown in the Fig. 3. The boxes marked with a box inside indicate that this task is a subprocess.

We verified the properties of our DP Workflow model using a couple of software tools: WoPeD 1.0 [8] and Woflan 2.2 [10]. WoPeD was developed at the University of Cooperative Education (Berufsakademie) Karlsruhe, it is a tool for editing and simulating WF-nets, and it uses Woflan as analysis tool. Woflan was developed at the Technische Universiteit Eindhoven and it checks for soundness in WF-nets.

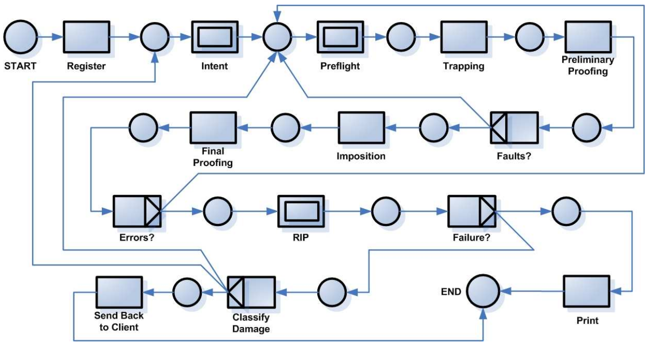


Figure 3. Workflow model for the Digital Publishing pre-press process

4 Measuring dependability attributes using a workflow model

We obtained values of dependability attributes in a quantitative form from the workflow model instead from a combinatorial model. In order to do this, we adapted some methodologies and strategies that build models to the creation of the workflow process definition.

This approach will lead to the addition of parameters to the model, in order to make it more accurate. Constrasting the analysis provided in [5] and [11], our work proposes an analysis of quantitative dependability over a workflow model, and not only an analysis of its structure and performance. Furthermore, we suggest a methodology to include parameters for dependability analysis into the workflow process definition mapped from the business process of the high dependable system. We want to obtain the qualitative dependability attributes of the system making a *fault forecasting analysis*, which is performed by evaluating the system behavior with respect to fault occurrence or activation. The qualitative evaluation of a system aims to identify, classify and rank the failure modes (the different ways a system can fail) or the event combinations (component failures or environmental conditions) that would lead to system failures.

A study over a variety of PDF documents produced by a diversity of software tools was done in order to identify the most common critical faults present in a job. For this, the reports produced by a commercial preflight tool were analyzed, finding that faults related with not embedded fonts, low image resolution, objects overlapping safety zones and images using the wrong color base were the most common ones. According to our study, the mean probability of find a fault related with fonts not embedded is 67%, the mean probability of find a fault related with a wrong color base is 38%, and the mean probability of find a fault related with a low image resolution is 61%.

Table 2 shows the most relevant faults that could be present in a document (based on [4] and on our own study). Faults are classified by the kind of failure that they are able to generate. For instance, a fault related with fonts will lead to a failure that could be classified in domain as a content failure (C) or a timing failure (T). A content failure refers to a failure in the content of the service (in DP, for instance, if an image is printed out of the paper margins) and a timing failure refers to a failure in the time of completion of the service (in DP, a job that takes an overdue time in being completed).

The consistency (C) or inconsistency (I) of a failure is seen in if this failure is perceived by all final users in the same way or not, respectively (in DP, a change in the font type could be seen in a different way by each client, and it depends on the clients job, for example, a brochure, a book or a magazine). The consequences of a fault could be classified in minor, medium, and catastrophic, dependent on the severity of degradation in the final service provided. Faults such as incomplete or corrupted files could lead to catastrophic failures (C). In contrast, if an image placed in a document requires being in CMYK color process, and is in RGB, these fault could lead to a medium (Md) or minor (Mn) failures.

Table 2. Possible faults present in preflight

	Domain	Consistency	Consequences
Not embedded Fonts	Content Timing	Inconsist.	Medium
Low image resolution	Content Timing	Inconsist.	Medium
Wrong color base	Content Timing	Consist.	Minor
Missing images	Timing	Consist.	Catastrophic
Incomplete or corrupted files	Timing	Consist.	Catastrophic

The first stage that we modeled was the preflight prepress process, serving as our case of study. Fig. 4 shows the model representation for the preflight sub-process. The job is registered in the preflight module, then a preflight report about it is made, according to a preflight profile preselected. The report is reviewed by the preflight technician or expert. The technician search and repair, if available, faults in the document, for instance, faults related with fonts, image resolution, or color bases. In next sections we discuss in detail the preflight stage serving as our case of study.

We first show an alternative model to measure quantitative dependability attributes from the workflow model of this process. Then, we illustrate our methodology to create a workflow model able to measure quantitative dependability attributes from the system.

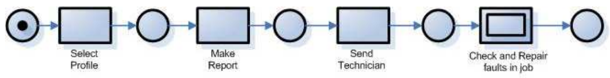


Figure 4. WF-net for the preflight subprocess

4.1 Alternative model to measure dependability for DP

In our case of study we assume that the pre-pres process of preflight has five fault sources that could lead to a failure of the system, i.e., the pre-flight pre-press process rejects the job analyzed. Fig. 5 shows the fault tree representation of the preflight subprocess. The couple of events *A*, *B* and *C* represent faults that are fixable in the preflight station. *A* represents faults related with fonts, *B* faults related with image resolutions and *C* faults related with wrong color bases. The first event of each couple represents the presence of the fault in the job and the second event of the couple represents the ability of repair that kind of fault. Events *D* and *E* represent non-fixable faults. *D* represents faults related with corrupted files and *E* faults related with missing images.

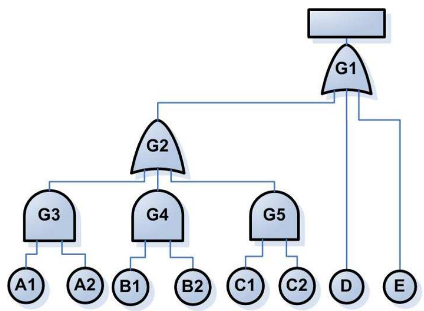


Figure 5. Fault Tree for the Preflight pre-press process

The analytic equation that describes the probability of failure of the preflight pre-press process, analyzing the fault tree in the Fig. 5, was obtained. Thus, the expression of the probability of failure of the preflight stage is the equation (1). We used the corresponding letter instead of use the events multiplication (for instance: $A \Rightarrow A_1 \cdot A_2$)

$$P_F = (A + B + C - AB - AC - BC + ABC) \cdot \\ \cdot (1 - D - E + DE) + D + E - DE \quad (1)$$

The equation for the reliability of the preflight stage is: $R = 1 - P_F$. Replacing into this equation, we have:

$$R = 1 - [(A + B + C - AB - AC - BC + \\ + ABC) \cdot (1 - D - E + DE) + D + E - DE] \quad (2)$$

4.2 Develop a WF-net including fault parameters for Preflight

To analyze quantitatively dependability attributes in a WF-net, it is necessary to add the identified possible faults of the system into the model generated. We include those faults into the model of the Fig. 4 by replacing the task *check and repair faults in the job* by five parallel sub-processes, using an AND-split/AND-join construction. Each of them check the presence of a kind of one of the most relevant faults that a job could have. The resulting net is shown in the Fig. 6

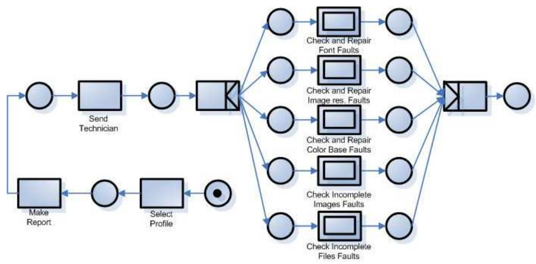


Figure 6. Replacing the subprocess ”check and repair faults in the job” in the preflight stage WF-net

We defined a fault treatment sub-process, shown in the Fig. 7. For constructing this WF-net we add an OR-split that checks for the presence of faults. Later, we include another OR-split checking if the fault can be repairable. Finally, we put a task for each of the three situations (no faults, repairable and not-repairable). Subsequently, we replace each sub-process in Fig. 6 for our fault treatment subprocess. This give us as result, a better WF-net for the preflight prepress process, which is shown in the Fig. 8

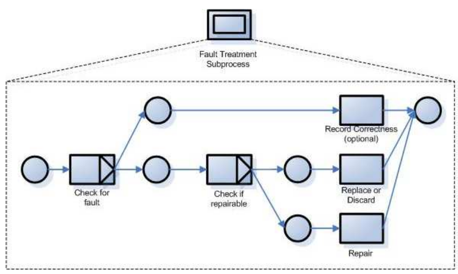


Figure 7. Fault treatment subprocess

In the same manner that we test correctness in the DP workflow, we test the correctness of the WF-net for the preflight pre-press process using WoPeD and Woflan. The WF-net for preflight complies with workflow model parameters.

To obtain either performance or dependability measures

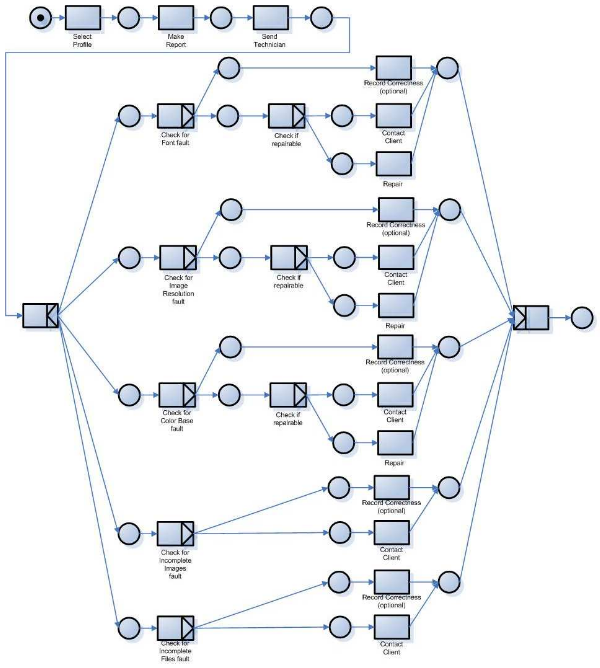


Figure 8. Complete WF-net of the preflight process after include the fault treatment sub-process

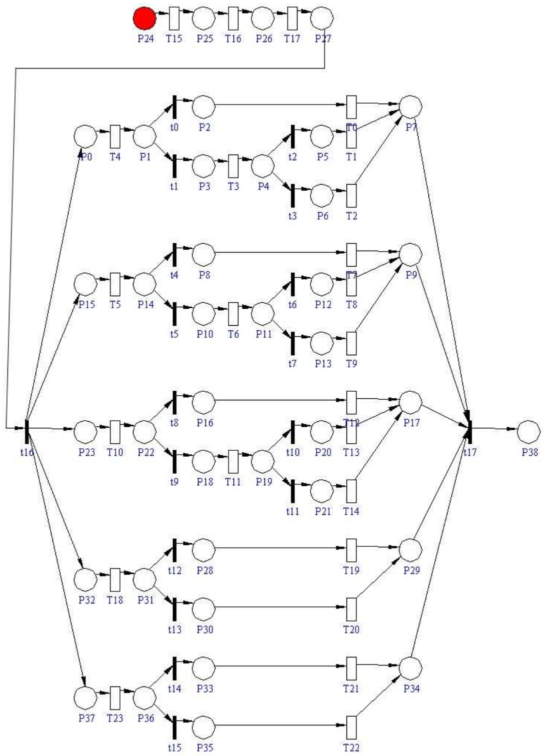


Figure 9. Workflow Model of the Preflight pre-press process based on GSPN

from a WF-net model, time and fault parameters must be included. Our methodology includes time as a random variable exponentially distributed on some transitions of the net, and include fault parameters as immediate transition weighs. Consequently, the net must be converted into a GSPN. To do so, it is necessary to replace the OR-split blocks by a combination of transitions. The outcome is governed by a fault probability . The OR-split block is replaced by an exponential transition simulating the action followed by two immediate transitions disposed in parallel, assigning the fault probability to the weight of one of the immediate transitions and its complement to the other one. The resultant model replacing the OR-split blocks is shown in Fig. 9. This net is a Quantitative Dependability WF-net based Model (QDWM).

In order to obtain the quantitative dependability attributes of the system, it is necessary to follow a series of steps. First, it is necessary to generate the reachability graph of the GSPN. From this reachability graph it is possible to deduce the associated Continuous Time Markov Chain (CTMC). Doing the steady-state analysis of the CTMC is obtained the probability of the system of being in any of its states, thus it is achievable to work out the dependability attributes of the system checking the corresponding combination of probabilities of being in certain states that belongs to each attribute. This analysis was done using a software tool named SHARPE [3]. SHARPE is used to model and validate distributed systems using GSPN, among other kind of models. This tool offers a multi-environment graphical interface and provides a specification language and solution methods for performance and reliability modeling.

5 Comparative results

We have shown two methodologies to analyze dependability on a system so far. The first one is a fault tree model that involves the faults and the final system conditions to reach a failure. This combinatorial model gives the opportunity of analyzing the quantitative attributes but it does not give the opportunity to analyze either the performance or the structure of the system.

The second one, our proposed methodology, incorporates the combinatorial model of the system based on the faults that this model itself could contain into a workflow model preserving the characteristic routing rules, and behavior of a workflow model, enhancing the model. The resultant net will contain not only fault and failure states, but also normal workflow states, making the model more accurate. Besides structure, behavior, performance, and dependability could be analyzed using the same net.

In this section are shown the results from a QDWM for the preflight pre-press process, compared with the results obtained from a Fault Tree model. These results are obtained introducing different vectors of fault parameters to each model and comparing the outputs. For our case of study, the eight fault parameters on each vector are: Probability of find a fault related with fonts, image resolution, wrong color base, incomplete or corrupted files and, incomplete or missing images. Besides, probability of not repair faults related with fonts, images resolution, and wrong color bases. For each model the fault parameters are the same. In the fault tree case, outputs are obtained from the equation (2), whereas in the QDWM case, outputs are acquired from the analysis software tool. To do so, a dummy place is introduced into the net and every task that treats the case of non-repairable fault is connected to this place. Subsequently, it is analyzed the probability that this place is empty in steady-state. Thus, we obtain the reliability of the whole process in steady-state. For instance, in Fig. 9, transitions $T2$, $T9$, $T14$, $T20$, and $T22$ would be connected to a dummy place (not depicted).

Using ten different vectors generated randomly, we can see that the steady-state reliability obtained from the QDWM of the preflight pre-press process, is totally related with the reliability obtained from the Fault Tree model. The mean error between both measures is around $8.0 \cdot 10^{-5}$ and the correlation between them is 0.9999, which is highly close to one. For these reasons, we can tell that a QDWM is able to measure reliability from a process. Due to its WF-net properties were not altered, this QDWM also allows to measure performance attributes of the process and it conserves its main intended function: to be a workflow model.

Concerning to our case of study, we assume a input vector in order to analyze the reliability of the preflight pre-press process. This vector is as follows: probability of find a fault related with fonts, 0.67; image resolution, 0.39; wrong color base, 0.61; incomplete or corrupted files, 0.05; and incomplete or missing images, 0.05. Besides, probability of not repair faults related with fonts, 0.08; image resolution, 0.05; and wrong color base, 0.03. The resulting reliability is 0.82. The reliability for this pre-press stage is around 0.90 according to local printshops.

6 Concluding remarks

The resulting WF-net from the inclusion of the fault treatment subprocess into the initial workflow model is the QDWM once it is translated into GSPN. Every subprocess used to create these WF-nets is sound, and those nets were tested using specialized software. Accordingly, the QDWM created is completely sound. Therefore, the QDWM allows to analyze the dependability of the system and its methodology of creation preserves the its routing structures and its function to measure performance too.

It was shown that structural and performance analysis are important in a workflow model in order to guarantee the best workflow process definition to be implemented in a workflow management system. It is also important to measure dependability attributes and to refine the design of that workflow process definition. Dependability attributes have been measured using combinatorial models based on the faults and their probability of occurrence in the system. The idea proposed is to measure quantitatively dependability attributes from the workflow model itself, improving the analysis tools necessary to achieve a good workflow process definition. Applying this new measure concepts to the general Digital Printing prepress process ensures a better workflow management in this area, because this process is highly based on its trustworthiness. Those measures of dependability are intended to help in the design of a more reliable system.

References

[1] A. Avizienis, J. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 01(1):11–33, January-March 2004.

[2] F. Bause and P. Kritzinger. *Stochastic Petri Nets, an introduction to the theory*. Friedr. Vieweg & Son, 2002.

[3] C. Hirel, R. Sahner, X. Zang, and K. Trivedi. Reliability and performability modeling using sharpe 2000. In *Proceedings of the 11th International Conference on Computer Performance Evaluation: Modelling Techniques and Tools*, volume 1786, pages 345–349.

[4] M. L. Kleper. *The Handbook of Digital Publishing - Vol. II*. Prentice-Hall, 2001.

[5] J. Li, Y. Fan, and M. Zhou. Performance modeling and analysis of workflow. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 34(2):229– 242, March 2004.

[6] B. Mikolajczak and D. Byrne. Workflow modeling and diagnosis with petri nets - a case study of a manufacturing process. In *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, page 6 pp. vol.5, October 2002.

[7] D. Monkerud. Realizing the promise of workflow automation. *Digital Publishing Solutions Magazine Online*, May 2004.

[8] U. of Cooperative Education (Berufsakademie) Karlsruhe. Woped ("workflow petrinet designer"). http://www.woped.org, Accessed: March-2006.

[9] W. M. P. van der Aalst and K. M. van Hee. *Workflow Management: Models, Methods, ans Systems*. The MIT Press, Cambridge, Massachusetts, 2002.

[10] H. Verbeek, T. Basten, and W. van der Aalst. Diagnosing workflow processes using woflan. In *The Computer Journal. British Computer Society*, volume 44, pages 246–279, 2001.

[11] R. Zurawski and M. Zhou. Petri nets and industrial applications - a tutorial. *IEEE Trans. on Industrial Electronics*, 41(6):567–583, 1994.