Composite web QoS with workflow conditional pathways using bounded sets

Houwayda Elfawal-Mansour, A. Mansour & T. Dillon
Your article is protected by copyright and all rights are held exclusively by Springer-Verlag London Limited. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your work, please use the accepted author’s version for posting to your own website or your institution’s repository. You may further deposit the accepted author’s version on a funder’s repository at a funder’s request, provided it is not made publicly available until 12 months after publication.
Composite web QoS with workflow conditional pathways using bounded sets

Houwayda Elfawal-Mansour · A. Mansour · T. Dillon

Received: 7 August 2010 / Revised: 29 April 2012 / Accepted: 5 May 2012
© Springer-Verlag London Limited 2012

Abstract In our previous work (Dillon and Mansour 2009), a stochastic reliability model of atomic web services was proposed. Using the well-known classic two-state bounded set technique, we developed a service-oriented model that dynamically calculates the reliability of composite web services with rollback recovery (Mansour and Dillon in IEEE Trans Serv Comput 4(4), 2011). In order to improve the Quality of Service, fault tolerance techniques have been introduced using recovery block adaptation. Our workflow was based on series-parallel structures that constitute parts of existing structures. It is worth mentioning that major service-oriented systems contain larger and more complex structures than the simple series and parallel ones. This is a limitation in our previous approach. In order to consider more realistic service-oriented systems, other main structures, such as AND, XOR and Loop, should be included into our model. In this article, our previous structures are generalized to include AND, XOR and Loops. In addition to generalized structures, we extended the existing two-state bounded set technique to include three-state systems. This extension was especially motivated by XOR-based structures. A comparative study between bounded set techniques and a new stochastic model is also presented. Our simulation results accurately reflect the performance of the new proposed model and confirm our theoretical studies. Furthermore, Monte Carlo simulations were performed and the results obtained clearly validate our stochastic model.

Keywords Bounded set · Composite services · Quality of service · Path-based · Reliability · Workflow

1 Introduction

Web services [1–4] are self-contained component applications that can be described, published, located and invoked over the internet. Generally, the requirements of a user cannot be met by a single web service (WS) and users may require the deployment of several component web services.

Dealing with composite web services raises two issues: the selection of the required component and execution coordination of the different web services. The Business Process Execution Language for Web Services (BPEL4WS) was created to deal with these issues and support some control structures like splits, joints and to describe the composition where a business process can be modelled as a workflow of atomic web services [5]. In a series composition, the components are represented by a flow graph where each node models an execution stage that must be completed before transition to the next node. In parallel composition, at least one of the web services must be completed before transition to the next node can take place (redundant components) [6, 7].

Composite web services can be characterized by their reliability, failure probability, security, efficiency, dependability of their components, etc. To study these properties, authors have used various approaches, see next section. Besides these classic methods, a promising original approach, the bounded set technique, was introduced by the pioneering work of Cheong and Dillon [8]. This method involves a systematic decomposition into mutually exclusive sets of states and the
development of the upper and lower bounds for these sets. A classification of these sets of states into working and failed sets is carried out by examination of the bounds on these sets. The bounded set approach, therefore, serves to complement existing techniques and enables large systems, containing modules having arbitrary failure-time distributions, to be modelled. This would otherwise be difficult to cope with using traditional techniques [9]. For example, when a system contains cold spares, which are powered up only when they become active modules, the failure characteristics of each spare module will depend on which module it replaces. Here, we are clearly faced with the problem of evaluating either the A or B set probabilities in the presence of these dependencies. The straightforward method is to evaluate them using the renewal-theory approach. However, this is, in general, not feasible for large systems due to the many combinations of sequences of failures of the modules and the difficulty of solutions inherent in the renewal-theory approach [9]. Using the bounded set technique, the authors in [10] prove that the bounded set can enable modelling Markov systems with un-powered spares and systems which would be difficult to model using traditional approaches.

In a previous work [11], we developed a new reliability model for atomic web services that depends on the environment variables, especially the loading on the server where the atomic web service resides. On the other hand, the dependability of service-oriented applications is an interesting and challenging problem. In fact, Damiani et al. write [12]: “Dependability is another crucial property of service-oriented applications. While design patterns for dependable atomic services have long been proposed, ensuring the dependability of a service composition is a much harder problem. The article Dependability and Rollback Recovery for Composite Web Services, by Mansour and Dillon [13], provides a new, promising solution to this difficult problem”. In order to simplify our discussion and gain insight, a generic study case is considered in this manuscript. In [14], the authors propose and discuss an interesting motivating example which can be adapted to fit our study. In fact, the actual study extends and generalizes the previous approaches in order to develop a unified approach considering conditional cases. On the other hand, we modify and adapt the bounded set technique to deal with complex WS combinations by considering the three-state case. Besides that, a new workflow “alternative” model has been proposed. Finally, a new probabilistic approach has also been developed and a comparison between the two approaches is provided.

In the first part of this article, we briefly introduce reliability of atomic and complex combinations of WS and propose a model for dynamic reliability calculations. In the second part, we extend the bounded set technique to the three-state case. Finally, simulation results for the general composite services are presented and discussed.

2 Related work and discussion

Many software reliability prediction models have been proposed for stand-alone software systems [15–17] and more reliability prediction approaches are proposed for component-based systems and for service-oriented systems [18–20]. However, most of these previous approaches focus on system-level reliability compositional analysis and assume that the component reliabilities are known.

Authors in several articles [21–23] consider component-level reliability prediction, which is mainly designed for traditional component-based systems. Goseva-Popstojanova et al. [22] calculate reliability risk of a component by the component complexity and the levels of its failures. Reussner et al. [23] compute component reliability by the reliabilities of its services, which are assumed to be known. Both these approaches require internal information of the components for making component reliability predictions. Cheung et al. [21] predict the component reliability at architecture design phase by exploiting behavioural models from sources of information available at design time.

2.1 Zheng and Lyu’s approach for reliability prediction

In [24], Zheng and Lyu propose a new and different approach on collaborative reliability prediction of SOA. Their approach focuses on system-level reliability aggregation and also on component-level web service failure probability prediction.

The main idea of this approach is to use the previous known failure history of different users to predict the failure in the system of a chosen user who is using similar systems or services.

A service-oriented system is presented as a service flow and the average failure probability of a web service is defined as the probability that an invocation to a web service will fail. For each task, an optimal candidate is selected from a set of functionally equivalent web service candidates. An execution plan is obtained by composing the selected web services which will be invoked to implement the abstract tasks.

In order to achieve their approach, their method heavily relies on similitude among used systems and posthistory. Their work consists of the following four phases:

- First phase: Using the collected data (posthistory), a correlation function generates a similarity function between two service users based on their commonly invoked web services.
– Second phase: After calculation, the users that have larger correlation values will be selected as similar users service, and users that have little correlation with the current user will be excluded.

– Third phase: In this phase, the probability prediction is done based on the mean value of the similarity functions.

– Fourth phase: In this final phase, the aggregated failure probability of the chosen service flow (including sequence branch, loop, parallel), is obtained and used with the exponential reliability function for the calculation of the reliability of the service flow.

It is worth noting that in the case of secure private local networks (i.e. defence applications, bank systems, surveillance), it is extremely hard to apply their approach as all these systems have these major common characteristics:

– Original design: Specific hardware and software dedicated to specific applications.

– Extremely secure data: In this case the service managers could not communicate or share these failure experiences (for reasons such as security, reputation, confidentiality).

Different from the above-mentioned systems, we assume in our approach that the system is a stand-alone, having dynamic reliability calculated independently from the history data and the complexity could be reduced in case of large systems. It means that our approach could be applied as a pre-system to the above-mentioned systems. Another advantage of our approach relies on the fact that we are operating in SP systems.

2.2 Guo et al. approach for dynamic reliability calculation

Concerning the dynamic reliability model, work on adaptive maintenance of composite services is done by Guo et al. [25]. In this article, the authors propose a system that constructs a closed-loop control for adaptive maintenance of composite services. By modelling the control process as a Markov decision process, they designed a Kalman-Filter-based algorithm for service state prediction. Considering the case of composite service, each component service is represented by numerical values of availability, reliability, trust and price at time $t$, respectively. Data are collected from the service information manager (SIM) which collects feedback results of some selected sampling clients. Concerning the reliability, it is measured by the ratio of the successful execution number and the total number of execution. Their work is mainly done following 6 phases:

– First phase: Collecting the sampling clients using report data about service execution and summarizing the state and value of component services

– Second phase: Calculating the dependability of composite services and comparing with the expectation value

– Third phase: Estimating the value of every selected component service by the Kalman-filter formula

– Fourth phase: Computing the immediate reward of every action and choosing the corresponding action

– Fifth phase: Executing a special algorithm which was independently developed by Cardoso et al. [18]

– Sixth phase: Selecting new services and replacing declining or failed services to implement the composite service’s re-structuring

Similarly to the work of Guo et al., our approach of dynamic reliability uses the parameters delivered by the service information managers but the reliability is modelled in a novel and more realistic way using the doubly stochastic model with randomly varying environments. This represents a breakthrough for the reliable, safe, successful configuration and deployment of such systems.

3 Stochastic and dynamic reliability models for WS

Services that are required by users must reflect quality according to different perspectives. Some users, for example, value reliability of service, while others focus on price, execution time and transportability. In this section, we present an overview of reliability modelling for WS with a special focus on dynamic reliability modelling.

3.1 Atomic WS reliability

Many reliability models assume that the failure or the error arrival times are exponentially distributed [26]. This is inadequate for web services, as error arrival times may be dependent on the operating state, including server workload. The service execution could experience long delays or even be blocked due to congestion arising from data traffic. To properly model the reliability of web services, one has to take into account software applications, the loading levels on the server running the application, and the possibility of execution blockage of a particular service due to the volume and the nature of the traffic.

In modelling error arrival times for web services, it is more appropriate to use the randomly varying environment concept. For the general reliability problem, randomly varying environments have been introduced in [27]. In our previous work, we have also developed a reliability model based on randomly varying environments [11,13]. To clarify the idea that was developed, in this manuscript, we briefly describe the basic structure of the reliability model used for atomic components. For the model shown in Fig. 1, we consider two operating environments, namely:
We associate with each of these states probabilities that the web service will remain error-free. These probabilities are assumed to be exponentially distributed and depend on the hazard rates $\lambda_1$ and $\lambda_2$ ($\lambda$ is the number of errors per time unit). The variance concerning the transitions among environment states depends on the server loading level and the types of processes being handled. These two transitions (idle to active, active to idle) have probability distributions $\alpha_1$ and $\beta_1$ called “$K$th order Erlangian Distribution” that depends on the variables $k$ (Erlangian order), $a_1$ and $b_1$ (number of events per time unit). These models can be used to determine the mean time to error (MTTE) as a function of different environment variables. As the MTTE $= 1/\lambda$ where $\lambda$ is the error rate [28,29], then having the MTTE for each time $t$, one can approximate the reliability using the exponentially distributed probabilistic density function $\lambda e^{-\lambda t}$. The time-dependent reliability is a function of server loading and can be calculated for the different web services used for the composite system reliability calculations.

It worth mentioning that Zheng and Liyang in [30] considered a dynamic reliability model based on order statistics and probability differential equations. On the other hand, dynamic web service selection for reliable web service composition is studied in [31].

### 3.2 Complex WS reliability

A main issue for the QoS analysis of service-oriented architecture (SOA) is parameter estimation. Concerning the reliability modelling for SOA [26], the architectural models are partitioned as follows:

1. **path-based models**, where the reliability of an assembly of components is calculated starting from the reliability of architectural paths;
2. **state-based models**, where the reliability is calculated starting from the reliability of system states and from the transition probabilities among states.

There are different ways that individual services can be integrated to form a business process. In the following, we represent the essential workflow for WS and their basic relationships including sequential, parallel, conditional and loop. To improve the overall reliability of composite web service, one can use redundancy or alternative services [6,7]. This kind of composite will also be included in our study in order to consider all the existing models.

For all workflows, $r_i$ represents the reliability of the $i$th component and $r$ represents the reliability of the composite scheme.

#### 3.2.1 Sequential workflow

The sequential workflow is represented in Fig. 2. The components are presented by a flow graph where the first web service must be completed before a transition to the next node can take place.

The reliability of the composite is given by:

$$r = r_0 \cdot r_1.$$  \hspace{1cm} (1)

#### 3.2.2 Parallel workflow

The parallel workflow is represented in Fig. 3. The components are presented by a flow graph where both of the web services connected to the “And” node must be completed before a transition to the next node can take place. These services are invoked at the same time (parallel).

The reliability of the composite is given by:

$$r = r_0 \cdot r_1 \cdot r_2 \cdot r_3.$$  \hspace{1cm} (2)

#### 3.2.3 Conditional workflow

The conditional workflow is represented in Fig. 4. The components are presented by a flow graph where only one of the web services connected to the XOR node must be completed.
The reliability of the composite is given by:

\[ r = r_0 \cdot (p_1 \cdot r_1 + p_2 \cdot r_2) \cdot r_3. \]  

(3)

### 3.2.4 Loop workflow

The loop workflow is represented in Fig. 5. As indicated by its name, there is certain probability that this service will be called again and this is given by the probability value \( p \). The corresponding reliability is given by:

\[ r = \frac{(1 - p) \cdot r_0}{1 - p \cdot r_0}. \]  

(4)

### 3.2.5 Alternative workflow

This kind of workflow is used to improve the overall reliability of composite web service and it is also called a redundant component. Here, by redundancy we mean that the same kind of functionality is available in the web service provided by a different service provider [32]. A redundant component in a series of atomic services is represented by a parallel scheme which is part of the family of series-parallel (SP) systems. In this case, the components are presented by a flow graph where only one of the web services must be completed before a transition to the next node can take place. The reliability calculation in the case of redundant (or alternative) components shown in Fig. 6 is as follows:

\[ r = 1 - (1 - r_0) \cdot (1 - r_1). \]  

(5)

### 4 Standard bounded set technique (two-state case)

In [33], the authors propose the bounded set technique for managing a large number of states. The bounded set method involves a systematic decomposition into mutually exclusive sets of states and the derivation of upper and lower bounds for these sets. A classification of these sets of states into operational and non-operational sets is carried out by examination of the corresponding bounds. According to the bounded set method, the general system state \( X \) can be written as a \( b \)-tuple minterm \( X = x_1, x_2, \ldots, x_b \), with \( x_j \) being a binary variable associated with the state of the component \( j \). Each of the system states \( X \) can be classified into mutually exclusive sets of system states: The \( A \) subsets consist of non-overlapping sets of working states, while the \( B \) subsets consist of non-overlapping sets of states corresponding to system failure. In order to ensure mutual exclusiveness in this division, all states belonging to a given set must have a selected number of component states set to a fixed value of 0 or 1. These selected components are not allowed to have any other value within the given set. The remaining components may be either in 0 or 1 states and they are designated by \( u \) which denotes an unselected state or a “don’t care” state. All the system states in the set \( S \) can be defined as:

\[ X^S \leq X \leq X^S \]  

(6)

where \( X^S \) and \( X^S \) are the upper and lower limiting states of the set \( S \). Limiting states set the values of a state depending on its dimension. For example, in the case of two-state models (0: failure or 1: working), the upper limit is 1 and the lower limit is 0.

Let us denote by:

\[ J^S_1 \text{ set of all indices } j \text{ where } X^S_j = X^S_j = 1 \]
\[ J^S_0 \text{ set of all indices } j \text{ where } X^S_j = X^S_j = 0 \]
\[ J_S \text{ remaining indices } j \text{ where } X^S_j = 1, \ X^S_j = 0 \]  

(7)

We consider a system of \( n \) two-state devices having:

\[ \begin{align*}
   p_j : Pr \left( j \text{ state is in an operational state i.e. } X^S_j = 1 \right) \\
   q_j : Pr \left( j \text{ state is in a failure state i.e. } X^S_j = 0 \right)
\end{align*} \]  

(8)

where \( P(\cdot) \) represents the probability function. As all the sets are statistically independent, the probability of encountering any one of the system states in set \( S \), or simply the probability of \( S \) occurring is:
where \( i \in J^1_s \) and \( j \in J^0_s \). Therefore, if the failure characteristics, \( p \) and \( q \), of every module are known, the probability of the system in any set of states can be easily evaluated using the equation given for \( Pr(S) \) Eq. (9).

For a systematic decomposition, we can start from the most significant bit of the system state, that is, we observe what happens when the first module is working \( x_1 = 1 \) and when it is failing \( x_1 = 0 \). These two decomposed sets are mutually exclusive and together contain all the original system states which are masked in the process of the decomposition. After decomposition, the two subsets \( S_1 \) and \( S_2 \) are examined. Examination of each subset \( S_1 \) involves examining the upper bounding state \( X_1^u \) with all the \( u \) states set to 1 and the lower bounding state \( X_1^l \) with all the \( u \) states set to 0. If \( X_1^l \) is a failure state, then the subset \( S_1 \) falls into the \( B \) subset. However, if \( X_1^u \) is a working state and \( X_1^l \) is a failed state, then \( S_1 \) is an unclassified subset designated by \( U \). For \( S_2 \), if \( X_2^u \) is a working state and \( X_2^l \) is a failure state, then \( S_2 \) is a \( U \) subset and it has to be decomposed further by considering the next significant bit in \( S_2 \) to give \( S_3 \) and \( S_4 \). After each decomposition, the two resulting subsets are examined as before. The decomposition process stops once all subsets have been classified as \( A \) or \( B \) subsets. Note that in the assumed decomposition, the subsets \( A_1, A_2, B_1, B_2 \) are mutually exclusive. Hence, the decomposition process results in the masking of all unclassified or \( U \) subsets as they are replaced with a mutually exclusive set of \( A \) and \( B \) subsets. By collecting all the sets that fall into the \( B \) subsets (mutually exclusive), we can determine the failure probability \( P_F \) of the systems as follows [28]:

\[
P_F = \sum_{i=1}^{N} Pr(B_i) \tag{10}
\]

where \( B_i \) are failed subsets and \( N \) is the number of the failed subsets.

To fix our ideas, let us consider the case of the composite \( SP \) system shown in Fig. 7. Each node \( WS_i \) of the graph corresponds to a web service and ALT node corresponds to the alternative composite type. Let us assume without loss of generality that all web services have the same failure probability, that is, \( q_1 = 0, 2 \) for \( i = 1 \ldots 4 \). To calculate the total failure probability, one can apply the bounded set decomposition developed in this section, where the subsets \( A \) and \( B \) are shown in Fig. 8.

5 Extended bounded set technique: three-state case

It is straightforward to apply the classic bounded set technique (two-state) to the sequential, parallel or loop cases. These cases can be modelled by two-state. The conditional case, however, offers some difficulties as we need to deal with special cases where some components are in the failure states while the process is working. Let us take the example given in Fig. 9 where we represent a simple conditional XOR case with two different web service choices, that is, \( WS1 \) or \( WS2 \), where only one of these services must be chosen according to certain probabilities denoted by \( p_1 \) and \( p_2 \) in order to continue the execution. Here,
we have a special modelling case where the system can be working despite a total failure in one of its components.

As an example, let us take the case of failure for service WS1 in Fig. 9. If the user chooses the service WS2, then the process will still continue. This case of working state while having failure will be represented by a third state called covered state. Each one of these states will have a probability:

- State 0, probability \( q \)
- State 1, probability \( p \)
- State 2, probability \( r \)

In Table 1, we show the different probabilities and their corresponding definitions. We have a three-state case (0, 1 and 2) for the conditional XOR where:

\[
p + q + r = 1
\]

For the case of the XOR structure under consideration and shown in Fig. 9, there are two service choices with \( a \) and \( 1 - a \) probabilities.

Consider the three events presented in Fig. 9.

- \( A \) : Activated WS1
- \( B \) : WS1 Working
- \( C \) : WS2 Working

Let us take \( a \) and \( 1 - a \) as the probabilities of choosing WS1 or WS2. Let \( F \) be the failure event, then:

\[
F = B \cdot \overline{C} \cdot \overline{A} + B \cdot C \cdot A + \overline{B} \cdot \overline{C}
\]

The failure probability of the whole XOR structure \( q \) will be given by:

\[
q = P(F) = P(B \cdot \overline{C} \cdot \overline{A} + B \cdot C \cdot A + \overline{B} \cdot \overline{C})
\]

\[
= P(B \cdot \overline{C} \cdot \overline{A}) + P(B \cdot C \cdot A) + P(\overline{B} \cdot \overline{C})
\]

(14)

where \( P() \) represents the probability function. When the events \( A \) and \( B \) are independent, we can write \( P(A \cdot B) = P(A) \cdot P(B) \), so \( q \) in Eq. (14) can be written as:

\[
q = R_B(1 - R_C) \cdot (1 - a) + (1 - R_B)
\cdot R_C \cdot a + (1 - R_B) \cdot (1 - R_C)
\]

(15)

\[
q = 1 - a \cdot R_B - (1 - a) \cdot R_C
\]

where \( R \) stands for the reliability of a component. If we take \( W \) as the “working with no failure” event, then we will have:

\[
W = B \cdot C
\]

(16)

where \( B \) and \( C \) are the 1 “working” events for WS1 and WS2 respectively. The probability \( p \) that both components WS1 and WS2 are working is:

\[
p = P(B \cdot C) = P(B) \cdot P(C)
\]

\[
= R_B \cdot R_C
\]

(17)

where \( R \) stands for the reliability of a component. If we take \( W_f \) as the “working with failure” event, then we will have:

\[
W_f = A \cdot B \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C
\]

(18)

The last probability \( r \), which is the probability that the XOR structure is working even with failure components, is given by:

\[
r = P(A \cdot B \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C)
\]

\[
= P(A \cdot B \cdot \overline{C}) + P(\overline{A} \cdot \overline{B} \cdot C)
\]

\[
= a \cdot R_B \cdot (1 - R_C) + (1 - a) \cdot (1 - R_B) \cdot R_C
\]

(19)

The sum of all probabilities should be equal to 1, as could be easily verified from the three Eqs. (15), (17) and (19):

\[
q + p + r = 1 - a \cdot R_B - (1 - a) \cdot R_C + R_B \cdot R_C
\]

\[
+ a \cdot R_B \cdot (1 - R_C) + (1 - a)
\cdot (1 - R_B) \cdot R_C = 1
\]

(20)

For each state \([0, 1, 2]\) in Table 1, we can get the corresponding reliabilities \([q, p, r]\).

As the bounded set model previously developed is valid only for two-state cases, in the following section, we will develop a new model for the bounded set in order to include the conditional cases and to have a general model that reflects any business process case. The limiting states introduced in Sect. 4 will be 0 and 2. The upper limit is 2 and the lower limit is 0. In the three-state cases, the probability definitions
given in Eq. (8) for the two-state cases should be transformed as follows:

\[
\begin{align*}
  p_j & : Pr \left( j \text{ in operational state with no failure } X_j^S = 1 \right) \\
  q_j & : Pr \left( j \text{ in failure state} \right) \\
  r_j & : Pr \left( j \text{ in operational state with failed components} \right) \quad \text{i.e. } X_j^S = 2
\end{align*}
\]

Then, the probability of encountering any one of the system states in set \( S \), or simply the probability of \( S \) occurring is:

\[
Pr(S) = \prod_i p_i \prod_j q_j \prod_k r_k 
\]  

(22)

where \( i \in J_1^S, j \in J_2^S \) and \( k \in J_3^S \). Therefore, if the failure characteristics \( p, q \) and \( r \) of every module are known, the probability of the system in any state can easily be evaluated using Eq. (22). By collecting all the sets that fall into the \( B \) subsets (failure subsets), we can determine the failure probability \( P_F \) of the systems as follows:

\[
P_F = \sum_{i=1}^{N} Pr(B_i) 
\]  

(23)

where \( B_i \) are the failure subsets and \( N \) is the number of the failure subsets.

The flowchart corresponding to the bounded set method is presented in Fig. 10. It starts with one unclassified set, divides into three subsets and recursively checks the obtained sets looking for the failure sets \( B \) sets. The \( B \) sets contain all the failed subsets \( B_i \) in Eq. (23). The calculation stops when no more failure sets are generated.

**Three-state bounded set models**

Let us consider the following application of a series scheme with one XOR case and one atomic case as shown at the top of Fig. 11.

The components \( B, C \) and \( D \) are atomic services. Each of these components \( X \) have their own reliability \( p_X \) and probability of failure \( q_X \). The values \( a \) and \( 1 - a \) are the probabilities of choosing either service \( B \) or \( C \). In order to apply the bounded set, the XOR structure should be transformed into a single component with three states \( [p_{BC}, q_{BC}, r_{BC}] \) that depend on the probabilities of the single component of the XOR structure. Table 1 gives the \( [p, q] \) for each component and the probability \( a \) of choosing the XOR structure. Before applying the bounded set divisions, let us transform the XOR structure into a single component (named \( BC \)) with three states as shown in Fig. 11. The corresponding \( [p, q, r] \) are obtained as follows:

\[
p_{BC} = p_B \cdot p_C = 0.63
\]  

(24)

\[
q_{BC} = 1 - (a \cdot p_B + (1 - a) \cdot p_C) = 0.16
\]  

(25)

\[
r_{BC} = 1 - p_{BC} - q_{BC} = 0.21
\]  

(26)

For the bounded set, we start from initial unclassified states \( [u, u] \) corresponding to \( BC \) and \( D \) components. The bounded set of this initial set is given in Fig. 12.

The first unclassified \( u \) of the set \([u, u]\) corresponds to the three-state XOR structure, so we go from 2 to 0 and analyse the corresponding sets. The second unclassified \( u \) of the set \([u, u]\) corresponds to the two-state component, so we go from 1 to 0 and analyse the corresponding sets. We obtain three failure subsets \( B_1, B_2, B_3 \) and their corresponding failure probabilities (using Table 2 and Eq. 22) as follows:

- \( B_1 = [2, 0] \), \( Pr(B_1) = r_{BC} \cdot q_D = 0.042 \)
- \( B_2 = [1, 0] \), \( Pr(B_2) = p_{BC} \cdot q_D = 0.126 \)
- \( B_3 = [0, u] \), \( Pr(B_3) = q_{BC} = 0.16 \)

Using these values in Eq. (23), we obtain the probability of failure of the whole system as follows:

\[
P_F = \sum_{i=1,2,3} Pr(B_i) = 0.328
\]  

(27)

The application of the bounded set in the case of composite services therefore resides in transforming all the main XOR structures to obtain a single component with three states and then apply the bounded set to calculate the probability of failure for the composite services.

In a future work, we plan to extend the model to become a four-state model by introducing the concept of *error latency*. Error latency describes the fact that in digital circuits there is typically a delay between the occurrence of a fault and the first error in the output. In this case, the four-state should be as follows:

1. **Total failure**
2. **Partial failure** where errors occurred and the system will fail within a certain time due to error latency
3. **Partial working state** where the system is working even with some failure components
4. **Perfect working state** where all the components are working

This presentation should be important for the systems that are characterized by error latency. This error latency may be comparable with the mean time between failures [34] indicating that output sequences are reliable² long after a system failure has occurred.

² Probability that the output is correct regardless of whether or not there are faults in the components.
SOCA

6 Equivalent failure probability based on a probabilistic model

In this section, we will develop a new probabilistic approach to model the failure probability for web services. Let us consider the example given in Fig. 7 which could be extended to general cases. Each node $W_{S_i}$ of the graph corresponds to a web service and the ALT node corresponds to the alternative composite type. The failure probability of the whole system will be calculated by knowing the probability of each web service.

First, let us denote by:
The system presented in Fig. 7 is working, that is, $Y$ if $\{ \text{WS1 is working, i.e. } X_1 \} \text{ and } \{ \text{WS2 is working, that is, } X_2 \text{ is working or } \{ \text{both WS3 and WS4 are working, i.e. } X_3 \cdot X_4 \} \}$. This can be written as:

$$P_r(Y) = P_r(X_1X_2X_3X_4)$$  \hspace{1cm} (29)  

Then, the probability that the system is not working $\overline{Y}$ is developed as follows:

$$P_r(\overline{Y}) = P_r(\overline{X_1} + X_2 + X_3X_4)$$

This probability can be written as [28]:

$$P_r(\overline{Y}) = P_r(\overline{X_1}) + P_r(X_2 + X_3X_4) - P_r(X_1X_2 + X_3X_4)$$

In order to continue our development, we will use these well-known probability relations:

$$P_r(A + B) = P_r(A) + P_r(B) - P_r(A \cdot B)$$

$$P_r(A \cdot B) = P_r(A) \cdot P_r(B)$$ if independence

Based on these relations, the development of $P_r(\overline{Y})$ will give the following expression:

$$P_r(\overline{Y}) = q_1 + p_1q_2q_3 + p_1q_2p_3q_4$$

where $q_1 = P_r(\overline{X_1})$, $q_2 = P_r(\overline{X_2})$, $q_3 = P_r(\overline{X_3})$, and $q_4 = P_r(\overline{X_4})$. While the proposed probabilistic approach can be generalized to handle composite systems, its formulation for complex systems can be challenging.

In order to conduct simulations using this probabilistic model, we choose the Monte Carlo method [35] using random numbers and probabilities to solve problems. It is an iterative method to evaluate a deterministic model using sets of random numbers as inputs. The probability that the component $X$ is working $P_r(X = 1)$ is equal to $P_r(u > \alpha)$ where $\alpha$ is the failure probability for the different components of the study system.

However, this method has some drawbacks such as the sensitivity of the standard error to the number of samples. The standard error is equal to the standard deviation $\sigma$ devised by the square-root of the number of observations $N$ i.e. $\frac{\sigma}{\sqrt{N}}$.

The familiar law of random walk applies: to reduce the error by a factor of 10 requires a 100-fold increase in the number of sample points, and the complexity increases sharply in the case of a large system.

### 7 Simulation results

Let us consider the following application using basic structures of workflow for web services. In our application, web services are used to book package holidays. In our package, one could rent a car and have multiple payment options (Visa card or cheque). The different choices are assumed to be binary and are modelled using XOR structure. It is worth mentioning here that the assumption about binary choice could be straight-forwardly generated to multiple choice by using a set of XOR.
Figure 13 shows the main structure of our web server which contains different structures such as ALT, AND and XOR. Each component will be modelled using the doubly stochastic models and the corresponding reliability will be calculated using the bounded set technique for two- and three-state cases.

In the case of the ALT and AND structures, we can directly apply the two-state bounded set technique. But concerning the XOR structures, the mother XOR should be transformed to a single component named “11”. Then, we can get the corresponding reliability value (with no failure) \( p_{11} \) and failure probability value \( q_{11} \) as in Table 1. The sum of \( q_{11} \) and \( p_{11} \) will be less than 1 because of the covered state probability \( r_{11} \). So the sum of the three probabilities should be one as in Eq. (12).

In order to obtain the final probabilities of the XOR structure, we go through the following calculations:

\[
\begin{align*}
    r_{p23} &= 0, 3 \cdot r_{p2} + 0, 7 \cdot r_{p3} \\
    p_{23} &= p_2 \cdot p_3 \\
    q_{23} &= 1 - r_{p23} \\
    r_{p123} &= r_{p1} \cdot r_{p23} \\
    p_{123} &= p_1 \cdot p_{23} \\
    q_{123} &= 1 - r_{p123} \\
    r_{p45} &= p_{45} \\
    p_{45} &= p_4 \cdot p_5 \\
    q_{45} &= 1 - r_{p45} \\
    r_{p12345} &= 0, 6 \cdot r_{p123} + 0, 4 \cdot r_{p45} \\
    p_{12345} &= p_{123} \cdot p_{45} \\
    q_{12345} &= 1 - r_{p12345}
\end{align*}
\]

where \( p_j, q_j, r_j \) are state probabilities for the \( j \)th component as described in Table 1 and \( r_{p_j} \) is the sum of working probabilities \( r_j + p_j \). For a single component, the parameter \( r_j \) is equal to 0 because it is a two-state case. The subscripts in these equations could be a combination of numbers as for \( q_{123} \). In this case, it is equivalent to the failure probability of the group of nodes 1, 2, and 3. Our transformed component “11” is characterized by the following probabilities:

\[
\begin{align*}
    p_{11} &= p_{12345} \\
    q_{11} &= q_{12345} \\
    r_{11} &= 1 - p_{11} - q_{11}
\end{align*}
\]  

(30)

where \( p_{12345} \) is the covered working probability and \( q_{12345} \) is the failure probability. Using the three-state probabilities \( p_j, q_j, r_j \) with the probabilities for each single component \( p_i \) and \( q_i \), the bounded set is applied in order to calculate the failure probability of the whole system.

To better serve our comparison purpose, three different simulations have been conducted using some internal parameters. By considering static and dynamic reliabilities, failure probabilities were elaborated using bounded set and stochastic models. The main parameters used are inspired by the doubly stochastic model presented in Sect. 3.1. In the following simulations, the time of execution is \( t = 10^4 \), the error rates are \( \lambda_1 = 10^{-6} \) and \( \lambda_2 = 10^{-4} \), and the Lagrangian order is \( k = 2 \). Concerning the probabilistic model that was developed in Sect. 6, we know that \( Y \) is the event that the system in Fig. 13 is working and it is given by:
$Y$: the system is working

\[
Y = X_1 \cdot X_2 \cdot (X_3 + X_4 \cdot (X_5 + X_6)) \cdot X_7 \cdot X_8 \cdot X_9 \cdot X_{10} \cdot X_{11} \cdot X_{12}
\]  \hspace{1cm} (31)

where $X_i$ is the event that the ith component is working.

This model is used to determine the reliability of the system which turns out to be the probability $P_r(Y)$. The simulations are done with Matlab software using normal distribution functions and the values of the reliabilities obtained with the different simulations. The Matlab code is presented in the “Appendix A”.

### 7.1 Static reliability

By neglecting the time factor, we could consider that our system has static reliabilities.

#### 7.1.1 Bounded set under static reliability assumption

First of all, we should transform the XOR structure to a single node (node 11) following the calculations in Eq. (30) for the three-state case. Table 3 shows the obtained parameters of the XOR node 11.

For the remaining nodes, the static reliabilities are given in Table 4. Using the bounded set technique with the static reliability values, the failure probability of the system is equal to $0 \cdot 705$.

Before continuing the simulations, we would like to show the failure states (Bsets) which were obtained with the bounded set technique.

In total, we obtained 81 failure states and we show the Bset number 37 in a continuous line as presented in Fig. 14. As we have the same WS combination, we will have the same Bsets for all the simulations. One can clearly see the values of the 37th failure set which varies between $-2$ (unclassified) and 1 (working). These values indicate one of the 81 failure cases where the system is unable to work.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Parameters of the obtained component 11 with three-state $p, q, r = 1 - p - q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>0.4406 0.2579 0.3015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Reliabilities of the non XOR structure obtained using constant values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>N.1</td>
</tr>
<tr>
<td>r</td>
<td>0.80</td>
</tr>
</tbody>
</table>

### 7.1.2 Probabilistic method Eq. (31) under static reliability assumptions

The probabilistic method in Eq. (31) used with the static reliability values in Table 4 gives a reliability of the system equal to 0.2951.

#### 7.1.3 Comparison between bounded set and probabilistic method using static reliabilities

We can see that the reliability values coming from the bounded set and from the probabilistic method ($1 - 0.705$ and 0.2951) are almost the same. This is very useful in the case of small systems.

### 7.2 Dynamic reliability for the whole system

Again, the simulations are performed considering a system with 12 nodes including node 11 that results from the XOR structure whose parameters are given in Table 6. The reliability values are obtained using the dynamic stochastic model. Three different loading levels are considered: high, low and heterogeneous levels. Two different simulation methods are applied using the bounded set and the probabilistic method.

#### 7.2.1 High loading level

1. Bounded set using high loading level:
   
   The specific parameters for the XOR components are shown in Table 5. This XOR structure is transformed to a single node (11) and the corresponding parameters are shown in Table 6.
   
   For the remaining components from 1 to 12, the parameters are presented in Table 7. In these tables, we give the different loading value of $a_1$ for different components.
Table 5 Variable loading \( (a_1) \) for the doubly stochastic models of the XOR structure used in simulation 1

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1(\text{Sim1}) )</td>
<td>30</td>
<td>25</td>
<td>40</td>
<td>50</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 6 Parameters of the obtained component 11 with three states \( p, q, r = 1 - p - q \)

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
<th>N.6</th>
<th>N.7</th>
<th>N.8</th>
<th>N.9</th>
<th>N.10</th>
<th>N.12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{11} )</td>
<td>0.2347</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( q_{11} )</td>
<td>0.4440</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_{11} )</td>
<td>0.3213</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7 Variable loading \( (a_1) \) for the doubly stochastic models of the non XOR structure used in simulation 1

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
<th>N.6</th>
<th>N.7</th>
<th>N.8</th>
<th>N.9</th>
<th>N.10</th>
<th>N.12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_{1}(\text{Sim1}) )</td>
<td>100</td>
<td>90</td>
<td>95</td>
<td>80</td>
<td>87</td>
<td>80</td>
<td>95</td>
<td>97</td>
<td>80</td>
<td>98</td>
<td>88</td>
</tr>
</tbody>
</table>

Table 8 Reliabilities of the non XOR structure obtained using high loading levels

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
<th>N.6</th>
<th>N.7</th>
<th>N.8</th>
<th>N.9</th>
<th>N.10</th>
<th>N.12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r )</td>
<td>0.57</td>
<td>0.91</td>
<td>0.68</td>
<td>0.60</td>
<td>0.89</td>
<td>0.93</td>
<td>0.60</td>
<td>0.58</td>
<td>0.92</td>
<td>0.63</td>
<td>0.59</td>
</tr>
</tbody>
</table>

Table 9 Variable loading \( (a_1) \) for the Doubly Stochastic models of the XOR structure used in simulation 2

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1(\text{Sim2}) )</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 10 Parameters of the obtained component 11 with three-state \( p, q, r = 1 - p - q \)

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
<th>N.6</th>
<th>N.7</th>
<th>N.8</th>
<th>N.9</th>
<th>N.10</th>
<th>N.12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{11} )</td>
<td>0.7105</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( q_{11} )</td>
<td>0.1227</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_{11} )</td>
<td>0.1668</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 11 Variable loading \( (a_1) \) for the Doubly Stochastic models of the non XOR structure used in simulation 2

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
<th>N.6</th>
<th>N.7</th>
<th>N.8</th>
<th>N.9</th>
<th>N.10</th>
<th>N.12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_{1}(\text{Sim2}) )</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>11</td>
<td>6</td>
<td>4</td>
<td>10</td>
</tr>
</tbody>
</table>

3. Comparison between the bounded set and the probabilistic method using high loading levels:
   We can see that the reliability values coming from the bounded set and from the probabilistic method (1 – 0.9894 and 0.0106) are almost the same.

7.2.2 Low loading level

1. Bounded set using low loading levels:
   The specific parameters for the XOR components are shown in Table 9.
   This XOR structure is transformed to a single node named 11 and the corresponding parameters are shown in Table 10.
   For the remaining components from 1 to 12, the parameters are presented in Table 11. In these tables, we give the different loading values of \( a_1 \) for different components.
   A close look at Fig. 16 gives us an idea about the reliabilities obtained by the constant values or by the dynamic model.
   Using the bounded set technique with the low loading levels, the failure probability of the system using the bounded set technique is equal to 0.5221.
The tables and figures in the document are as follows:

**Table 12** Reliabilities of the non XOR structure obtained using low loading levels

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
<th>N.6</th>
<th>N.7</th>
<th>N.8</th>
<th>N.9</th>
<th>N.10</th>
<th>N.12</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>0.92</td>
<td>0.92</td>
<td>0.93</td>
<td>0.89</td>
<td>0.96</td>
<td>0.93</td>
<td>0.91</td>
<td>0.88</td>
<td>0.92</td>
<td>0.94</td>
<td>0.89</td>
</tr>
</tbody>
</table>

**Table 13** Variable loading \(a_1\) for the doubly stochastic models of the XOR structure used in simulation 3

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_1)(Sim3)</td>
<td>9</td>
<td>30</td>
<td>5</td>
<td>35</td>
<td>15</td>
</tr>
</tbody>
</table>

**Table 14** Parameters of the obtained component 11 with three-state \(p, q, r = 1 - p - q\)

<table>
<thead>
<tr>
<th>Sim3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p_{11})</td>
</tr>
<tr>
<td>(q_{11})</td>
</tr>
<tr>
<td>(r_{11})</td>
</tr>
</tbody>
</table>

**Table 15** Variable loading \(a_1\) for the doubly stochastic models of the non XOR structure used in simulation 3

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
<th>N.6</th>
<th>N.7</th>
<th>N.8</th>
<th>N.9</th>
<th>N.10</th>
<th>N.12</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_1)(Sim3)</td>
<td>100</td>
<td>8</td>
<td>50</td>
<td>80</td>
<td>10</td>
<td>5</td>
<td>80</td>
<td>95</td>
<td>7</td>
<td>70</td>
<td>90</td>
</tr>
</tbody>
</table>

2. Probabilistic method Eq. (31) using low loading levels (Table 12). Using the probabilistic method with low loading level values, the reliability of the system is equal to 0.4778.

3. Comparison between the bounded set and the probabilistic method using low loading levels:
   We can see that the reliability values coming from the bounded set and from the probabilistic method (1 − 0.5221 and 0.4778) are almost the same.

7.2.3 Heterogeneous

1. Bounded set using heterogeneous loading levels:
   The specific parameters for the XOR components are shown in Table 13.

   This XOR structure is transformed to a single node named 11 and the corresponding parameters are shown in Table 14. For the remaining components from 1 to 12, the parameters are presented in Table 15. In these tables, we give the different loading values of \(a_1\) for different components.

   A close look at Fig. 17 gives us an idea about the reliabilities obtained by the constant values or by the dynamic model.

   Using the bounded set method with heterogeneous loading values, the failure probability of the system is equal to 0.9587.

2. Probabilistic method Eq. (31) using heterogeneous loading levels (Table 16). Using the probabilistic method with heterogeneous loading values, the reliability of the system is equal to 0.0412.

3. Comparison between the bounded set and the probabilistic model using heterogeneous loading levels:
   We can see that the reliability values coming from the bounded set and from the probabilistic method (1 − 0.9587 and 0.0412) are almost the same.
### Table 16  
Reliabilities of the non XOR structure obtained using heterogeneous loading levels

<table>
<thead>
<tr>
<th>Node</th>
<th>N.1</th>
<th>N.2</th>
<th>N.3</th>
<th>N.4</th>
<th>N.5</th>
<th>N.6</th>
<th>N.7</th>
<th>N.8</th>
<th>N.9</th>
<th>N.10</th>
<th>N.12</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>0.57</td>
<td>0.91</td>
<td>0.68</td>
<td>0.60</td>
<td>0.89</td>
<td>0.93</td>
<td>0.60</td>
<td>0.58</td>
<td>0.92</td>
<td>0.63</td>
<td>0.59</td>
</tr>
</tbody>
</table>

### Table 17  
Parameters for the simulations using bounded set technique

<table>
<thead>
<tr>
<th></th>
<th>Constant</th>
<th>Sim1</th>
<th>Sim2</th>
<th>Sim3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_f$</td>
<td>0.7050</td>
<td>0.9894</td>
<td>0.5221</td>
<td>0.9587</td>
</tr>
</tbody>
</table>

### Table 18  
Parameters for the simulations using a probabilistic method

<table>
<thead>
<tr>
<th>Monte Carlo</th>
<th>Constant</th>
<th>Sim1</th>
<th>Sim2</th>
<th>Sim3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>0.2951</td>
<td>0.0106</td>
<td>0.4778</td>
<td>0.0412</td>
</tr>
</tbody>
</table>

### 7.3 Failure probability comparison

Using the constant and the variable reliabilities, the failure probabilities $P_f$ of the composite scheme using the bounded set are shown in Table 17.

We can see how the probability of failure obtained using the different loading values varies around the constant reliability provided by the web service. This is reasonable since the reliability should depend on different parameters including the server loading. For the first data set (Sim1), the failure probability is higher than the constant one because we took high loading values for the simulations. For the second data set (Sim2), the failure probability is lower than the constant one because we took low loading values for the simulations. The third one is done with variable loading levels (high, low and middle values) and it gives a value between the lower and the higher loading levels.

In order to verify the efficiency of the bounded set scheme, we applied the Monte Carlo method previously discussed. Table 18 shows us how the reliabilities obtained using the probabilistic model (Monte Carlo) are complementary to the values obtained with the bounded set shown in Table 17 (the sum is close to 1 with a difference of the order 0.0001).

### 8 Conclusion

In this article, we developed a model for composite web services real-time reliability based on the bounded set techniques to limit the computational burden in large scale systems. The real-time reliabilities are calculated using our hybrid path and state-based models [11]. In order to apply the bounded set for the conditional cases which cannot be handled by two-state, we developed a new model for the bounded set where we transform the XOR structures into single components represented by three-state. We reached the same results with the Monte Carlo calculation method, we were able to limit the number of failed states in the Bsets thus limiting the computational burden in the case of large systems.

### Acknowledgments
The authors would like to express their thanks to Prof Hadi Aggoun for his support.

### Appendix A: Matlab code for the probabilistic model

Supposing that the reliabilities of the different nodes are known and represented by the vector $R$, the Matlab code that generates the reliability of the whole system is:

```matlab
function Rel = montecarlosReliab(R)
    % R contain the values of the reliabilities for the different nodes
    % Generate n samples from a normal distribution
    r = (randn(n,1) * sd) + mu
    % Generate n samples from a uniform distribution
    r = a + rand(n,1) * (b-a), a : minimum, b : maximum
    N = 20;
    z = zeros(1,N);
    for i=1:N
        q1 = 1 - R(1);
        q2 = 1 - R(2);
        q3 = 1 - R(3);
        q4 = 1 - R(4);
        q5 = 1 - R(5);
        q6 = 1 - R(6);
        q7 = 1 - R(7);
        q8 = 1 - R(8);
        q9 = 1 - R(9);
        q10 = 1 - R(10);
        q11 = 1 - R(11);
        q12 = 1 - R(12);
        y = (X(1,:) > q1) .* (X(2,:) > q2) .* ((X(3,:) > q3) && ((X(4,:) > q4) && ((X(5,:) > q5) && (X(6,:) > q6))));
        z(i) = sum(y) / n;
    end
    Rel = mean(z);
end
```

### References

11. Dillon T, Mansour HE (2009) Dependability and error manifestation for a web service based on the doubly stochastic model and renewal processes. In: IEEE 15th Pacific Rim international symposium on dependable computing (PRDC’09), Shanghai, China
34. Boon C (1994) The latency problem in fault-tolerant and secure computer systems: Models, design, analysis, and validation. Ph.D. dissertation. Faculty of science and Technology, La Trobe University, Melbourne