SErvice Reliability and Availability model with Petri Nets: a new hybrid approach for service availability

L. Palummo[a]*, R. Merigiola[a]*, E. Guidolotti*, D. De Luca[+]
*System Engineering Department
*Aster S.p.A.
*via Tiburtina 1166, 00156 Rome - Italy
[+] Telespazio Via Tiburtina 965, 00156 Rome - Italy
a - corresponding author: lucrezia.palummo@aster-te.it, rachele.merigiola@aster-te.it

***Abstract*— Most part of the existing analytical models to predict service reliability and availability assume a static behavior of the service and do not take into account the correlation between the invoked system components. In order to take into account the dynamic aspects of a service as functional chains, operational processes and logistic support, a hybrid approach is here introduced: a dynamic SERA (SErvice Reliability Availability) Model, including a service simulation model based on hybrid Petri Nets. The main goal of the proposed model is to determine the reliability/availability of a service taking into account the characteristics of the service (functional chains and operative processes), as well as the SW/HW dependability figures (MTBF, MDT). In the proposed approach the service and its invoked system components are represented through Hybrid Petri Nets where the SW/HW failures have been modeled with stochastic distribution through kinematic Monte-Carlo time simulations.**

In order to refine and validate the proposed model a case study based on a simple user registration service has been developed. The results show the feasibility of the proposed approach along with a set of metrics used to quantify service performances on a statistical basis and evaluate service quality.

***Keywords*—RAMS, Availability, Service, Simulation, Petri Nets**

I. INTRODUCTION

Service Reliability significantly affects the operational transitions, the potential users, and the degree of adherence to requirements can affect the customer satisfaction and perception of service quality. If a user request it is not completed on schedule, service is perceived as unreliable, being the requested output delivered on longer times. Service unreliability can have a great impact on the system and its users.

The increasing demand for flexibility and extensibility of the services has resulted in a wide adoption of web services and SOA (Service-Oriented Architectures) [1,2,3] applications. Even if several studies and modeling of services have been used in the past to estimate and improve reliability and availability [4,5,6,7], the evaluation of modern systems remains a challenging problem due to the increased level of complexity. One of the most commonly applied approaches is the analytical methodology, which produces accurate results. Unfortunately, it becomes not applicable due to the size and complexity of models or due to non-linear nature of the problem involved.

The Reliability Modeling and Analysis to improve a Service Reliability have been proposed using the two-state model or finite state machine, Model–Based approach and also proper algorithms [8,9,10,11], to reduce the inefficient of the approximation methods and using a simulation to predict the behavior of system/service. Various simulation methodologies such as Monte Carlo simulation, Discrete event (DE) simulation, Subset simulation, Hybrid subset simulation, Simulated annealing, Stochastic simulation, Digital simulation, and Markov System Dynamics (MSD) simulation can be used in reliability engineering [12], and also a new method as RAMSAS based on SoS (System of System) Model using suitable model-driven techniques and simulation technique to evaluate the Reliability performance of the system and possibly, compare different design alternatives and parameters settings [13].

The simulation becomes need to predict the performance of the Service and drive the design.

II. BASIC CONCEPTS

A. *Reliability Definition*

Service reliability should not be confused with network reliability, which is instead related to the overall availability of the system. For this reason the following definitions are here introduced:

Reliability - the probability that service will be continuously available over a given period of time.

Service – an available system function which can be used by a person or a machine and it is based on a sequence of operations (called *transactions*) focused on state transitions.

Service Reliability - the probability that a service infrastructure will be continuously available in a given time considering hardware failures, software faults and human errors and focuses on the state of service execution.

Transactions are instead specific instances of a service use (e.g listening to the radio station; a user connects pc to the Internet; a shopper pays for a purchase using a credit card).

On several cases the Service overall of the System is composed by different services that share HW and some functions are interlocked between them. A service is evaluated by a list of success criteria to be fulfilled in order to achieve a continuous delivery of required outputs and the execution of transactions. Such a list is defined using the attributes of the service.

A *figure of merit* is any quantitative expression, expressed by means of a probability or other statistical parameters, used to describe a specific aspect of the study target. For example, the expected downtime during one operative year is a figure of merit for the reliability of a maintained system. A *metric* is instead a quantity used to evaluate the degree of adherence to a requirement (expressed by a figure of merit).

Service Reliability can be measured by several metrics, related to different aspects:

1. *End-User*: Service Accessibility, Continuity, Release

2. *Internal Metrics*: outages, duration, task interruption, failure distribution, incomplete instance and features not available.

3. *Performances* - total delay time during transaction, delivered products with or without delay.

B. *Definitions of Availability*

The concept of Availability was originally introduced for repairable systems, which are required to operate 24/7; in this case a failure could randomly occur along the operational life and a maintenance intervention is required to restore operations in a minimum time. There are several definitions of Availability on literature. A general definition is:

$$A_{op} = 1 - \frac{Downtime}{Total\_Time} = \frac{Uptime}{Total\_Time} \qquad (1)$$

where the Downtime includes a repair time (corrective and preventive maintenance time), a management time and a logistic time. In several cases it is worthwhile also to consider O&M organization, plans, procedures and tools dedicated to system management during the operational phase. In this case, *Operational Availability* can be defined:

$$A_{\mathrm{op}} = \frac{\mathrm{MTBF}}{\mathrm{MTBF} + \mathrm{MDT}} \qquad (2)$$

Where MTBF is the Mean Time Between Failure, MDT is the Mean Down Time, equal to: MDT=MTTR+LDT, where LDT is the Logistic Delay Time. For some applications, the user-oriented approach can characterize the system in a "black-box" manner and specifying availability according to the number off, for instance, delivered products, services, or mission data with respect to user demands or nominal scenario [14]. If the availability is specified by a percentage or number of successfully delivered products, the *Service Availability* (SA) shall be expressed as the ratio between the number N_c of completed requests and the number N_t of total requests:

$$SA = \frac{N_c}{N_t} \qquad (3)$$

For availability assessment, the suitable various methods can be performed:

- Analytical method
- Markov process
- Monte-Carlo simulation

This last numerical technique allows the evaluation of availability taking into account in a realistic way all aspects associated with the design, logistics and operations. The main advantage of Monte-Carlo simulation is the capability to represent complex system scenarios with deterministic or probabilistic delays.

III. HYBRID APPROACH TO SERVICE AVAILABILITY

A possibility to determine a Service availability is given by the combination of the traditional methods to evaluate Availability (Combinatorial method, Enumeration method, Simulations method) with the Service attributes. Traditionally these methods are separately used in the analysis framework, but in the proposed approach these four assessment kinds are unified with the goal to achieve a complete prediction analysis.

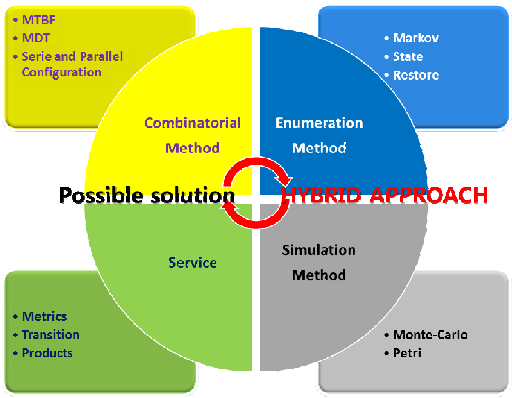

Fig. 1 – hybrid approach to service availability modeling

The steps of the process to define a hybrid approach model are listed as follow:

1. Define the estimators of a Figure of Merit of the Service (Service)

2. Define a flexible model taking into account the use case of the service and the SW/HW dependability figures (MTBF, MDT) (Combinatorial Method)

3. Use the representation of a complex system with the Hybrid Petri Nets (Simulation, Enumeration Method)

4. Model the SW/HW failures with stochastic distribution through kinematic Monte-Carlo time simulations; inputs are injected in the model and the delivered outputs are computed taking into account the functional chains and the operational processes (Simulation, Service)

To discuss service reliability, the persistence of service quality over time or the absence of service failures over time, it is necessary to know what failure means for a service. The key idea is that service failures are usually traceable to events or conditions in the infrastructure whose occurrence (or failure to occur) causes the service failure. That is, service failure mechanisms are found in the delivery infrastructure for the service. It is apparent that models for failure of transactions in a given service will depend heavily on the specific details of that service. This Section develops ideas for service reliability/availability modeling for each kind of services, using the same network topology (nodes & paths) as a way of illustrating how those details are used in creation and use of service reliability models.

The process of reliability/availability model is described as follow:

1. Define the steps of the reference Use Case and the involved HW/SW to execute the Service, according SE methodology.

2. Map the steps of the service execution; identify the nodes and find all possible paths related to the steps of service execution; the paths represent the connection between HW nodes, which involves a software function. Transform the paths into logical equations by applying "&" (AND) operators between nodes in the same path and the "||" (OR) operator between parallel paths.

3. Draw the operational workflow of Service.

4. Transform the logical expression of service into a Reliability Block Diagram (RBD) taking into account possible redundancy configurations.

5. Collect the system information (e.g. architecture, the reliability figures of each equipment, and SW application, maintainability figures as mean time to restore and logistic delay time).

6. Compute the Service Reliability and Availability figures according the Prediction models on the RBD base.

7. List the traceable System events or conditions whose occurrence or failure leads to the service failure and identify the permanent o transient failures that affecting metrics (FMECA or FTA can be used).

This method can estimate objectively the Service reliability/availability starting by the functional chains and operational processes, according the chronology execution of the service and considering the architecture needs to fulfill its performances. Moreover considering the RBD and the support of techniques such as Failure Modes Effect and Criticality Analysis and the Fault Tree Analysis, it is possible to identify which failures and their impact on the Metrics and Figures of Metrics.

This flexible model above described, is the base on drive the simulation, without which time evolution and transitions on the service states can be not considered.

A. *Proposed Methodology*

Define One of the main goals of the current study was to prove the effectiveness of the proposed hybrid approach to the evaluation of Service Reliability and Availability. For this reason a feasibility study based on a simple test case of an user registration service was performed. The study allowed to refine the proposed model and to clarify several aspects of the Simulation Model. In this Section the proposed methodology along with details of its application to the study case are reported.

The purpose of the user registration service is to allow to the end-user to insert a set of initial parameters to be recorded on database in order to receive credentials for future login and access to a system. In the current example the user registration service is deployed on three different sub-systems, each of those constituted by a SW and HW component:

1. Client – in charge to provide a Graphical User Interface to the user for data entry and display

2. Server – deputed to manage the registration requests and to interrogate the database

3. Database (DB) – deputed to record the registration requests and to provide the related feedback

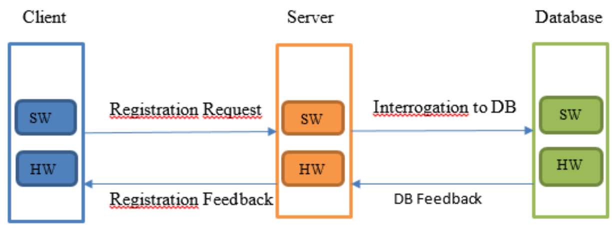

Fig. 2 – User Registration Service

The user will access to the Client (e.g. a web page via browser) and insert into the Client GUI data required for the user registration; after a data check, the Client will create a registration request and send it to the central server for request management and interrogation to database. The database will provide a feedback to the server about the correct user registration, and the server will turn it back to the Client in order to inform the end-user on the accomplished registration process. (Figure 2).

1) *Define the Use Case*

In a complex system the definition of a service can be based on the definition of the correspondent use case of the System along with the tracing of the involved System components invoked by the process. This task can be demanding for all those services invoking several configuration items or entire sub-systems, and in general it requires a preliminary analysis of the System and its components. Every logical step of the use case should be identified and correlated with the involved hardware and software components. During this phase it is also mandatory to define the level of abstraction to apply to the use case definition: for example, in the current test case HW and SW components are considered as single, independent units, characterized by its own reliability and availability, and no further level of detail is required. Fix the abstraction level is fundamental to determine the level of detail into the reliability/availability representation of each component, to be used as input into the SERA computation. Table 1 reports the use case of the service example (user registration), along with the involved HW/SW components.

TABLE I STUDY CASE

Step	Use Case of a User Registration Service		
	Description	*HW*	*SW*
1	The User access to the System GUI	Client	Client
2	The User inserts the registration parameters and submits them	Client	Client
3	The System checks the registration request	Server	Server
4	The System interrogates the database for a new user registration	Server	Server
5	The System creates an user account choosing the proper user profile.	DB	DB
6	The System provides the user with the credentials for accessing the system.	Server + Client	Server + Client

Along with the use case correspondent to the service it is required also to define the operational workflow of the analyzed service (e.g. by means of sequence diagrams or equivalent), in order to fix the sequence of logical operations that the System must perform to execute the service and in which order the System components are invoked (Figure 3).

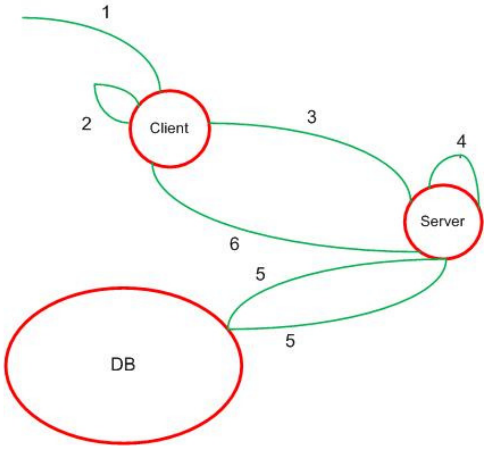

Fig. 3 Operational workflow of a service for the proposed study case

2) Compute the Reliability Block Diagram

Once the use case is defined and the operational workflow are defined, it is needed to understand how the connections between the different HW/SW components can affect the global reliability (and availability) of the functional chain. For this reason it is required to define the Reliability Block Diagram (RBD) according to the System HW/SW architecture. This analysis is generally based on system design documentation and on RAMS analysis documentation; in the proposed study case, the RBD is a simple chain (Figure 4), where no redundancy has been applied. Any occurring failure will lead to an interruption of the service, which could be permanent or temporary.

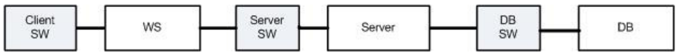

Fig. 4 RBD of the analyzed User Registration Service

3) Compute the HW/SW Availability

After the RBD definition and the detection of the single points of failure, it is needed to define as input data for the service availability analysis the availability figures of each HW/SW component invoked by the service process. Even if there are several parameters which could be used as availability metrics, in the current study is focused on the use of most common and applied parameters:

- Mean Time Between Failure - (MTBF)
- Mean Delay Time - (MDT)

These parameters will be computed according to the Prediction models on the RBD base. For the current test case, the following values have been reported on Table 2.

TABLE II COMPONENT MTBF and MDT

	Use Case of a User Registration Service			
Sub-System	*Component*	*Number of Failures (for 30 days)*	*MTBF (h)*	*MDT (h)*
Client	SW	3	240	4
	HW	1	720	4
Server	SW	2	360	4
	HW	1	720	4
Database	SW	3	240	1 or 4
	HW	1	720	4

4) Define the Service Failures

Once the model is defined, it is required to analyze and define the System events/conditions whose occurrence or failure leads to the service failure. It is possible to define service failures as Permanent (no recovery of the service) or transient (temporary failure which causes to the system to not be available within a predefined time period). In the proposed example, all failures occurred on HW have been defined as permanent, as well as failures occurring on Client and Server SW; failures occurred on Database SW have instead been defined as transient, assuming that the database is able to record the registration request and reprocess it later, keeping it as pending. This definition leads to the concept that a failure can affect both the functional aspect (the service process is interrupted) and the performance aspect of a service (the service process is not interrupted but the service is not in line within the expected performances).

TABLE III SERVICE FAILURES

Sub-System	Use Case of a User Registration Service		
	Component	***Permanent***	***Transient***
Client	SW	X	
	HW	X	
Server	SW	X	
	HW	X	
Database	SW		X
	HW	X	

5) Define the Service Metrics

In the definition of a SERA model it is fundamental to define a quantitative approach to evaluate the service availability along with the metrics to evaluate the robustness and the performance of the analyzed service. Several figures of merit can be defined to support such an analysis; here only a subset of most significant metrics have been taken into account, according to the study case.

Along with the service availability metrics, the current study introduced two distinct sets of metrics to evaluate the analyzed service: the Internal Metrics, used to monitor a specific aspect of the service related to single system components and the Performance Metrics used to quantify the service performances.

- *Internal Metrics*
  - Failure Distribution per Component
- *Performance Metrics*
  - Number of Failed Requests
  - Number of Recovered Requests with delay

6) Build the Simulation Model

After the collection of all the required inputs to describe the service and the invoked components, along with the definition of the metrics to evaluate the SERA, the final step consists into build a flexible and robust model capable to represent the service on the base of all the information provided from the hybrid approach. This process will be extensively described on Chapter 4.

IV. SERVICE AVAILABILITY SIMULATION MODEL

A. Introduction

The main advantages to develop dedicated simulations result into the capability to provide a quantitative evaluation of SERA and a rigorous representation of a specific Service and its dependency from the system components interaction. The proposed approach allows to compute and monitor the SERA evolution along time, as well as to evaluate its sensitivity to single system components. SERA simulations allow to perform feasibility studies, providing a powerful and flexible support to the system design phase. This aspect proved particularly useful in order to track and prevent unexpected service failures and/or system trends at system level.

Nevertheless, there are also some drawbacks to the choice to perform a SERA simulation w.r.t. a traditional, qualitative evaluation (e.g. FMEA/FMECA): simulations are time-consuming, require longer start-up times, and have high computational costs, which can be minimized by the adoption of proper computational facilities and by an adequate level of abstraction into the system components representation.

B. Comparative overview of Simulation Models

The availability of a service will be naturally dependent from the availability of the involved components; for this reason any service availability model will be based on the operational workflow including the involved components. A service model will have to represent a specific functionality, always related to an input request (e.g. a number of products to be released) and to an output generation (the released products). According to this view, a service availability can be modeled, among the several possibilities, by three main approaches:

- Adaptation of System simulators
- System Engineering SysML Models
- Petri Nets

System simulators are quite efficient to represent a system behavior and they can be adapted to retrieve the information related to a specific service, but at the price of an increased level of complexity in its representation, and into a limitation of the retrieved information (dedicated service modeling).

System Engineering SysML Models are also an efficient tool to represent system behavior and they can be modeled to include a service representation; this requires nevertheless the availability of a refined SysML Model at an enlarged level of detail, a prerequisite condition that often is not satisfied, for medium and complex systems.

Petri Nets are a flexible mathematical method used to represent discrete, continuous and stochastic variables [15]. Historically, Petri nets (PNs) are widely used to model discrete systems (computer systems, manufacturing systems, communication systems), but in the latter years, with the introduction of a representation for continuous and stochastic variables, their use has been enlarged to other fields (e.g. biology) [16]. Hybrid Petri Nets allow to represent stochastic and discrete behavior of system components at the same time with a good level of flexibility and scalability; this is the reason why they have been selected to determine the service availability on a statistical basis (Montecarlo simulations).

C. The Hybrid Petri Nets Model

Petri nets have been used to represent the use case workflow (see Section III). A Petri net is a mathematical modeling language used to describe distributed systems. It is not the purpose of the current paper to describe Petri Nets, extensively reported on related literature [15]; here only the basic elements required for the model understanding are briefly recalled. A Petri net is formed by the following elements:

- *Places/States* (P) – circular elements used to describe the state of a system component at a predefined time t;
- *Tokens* – black marks describing the data flowing into the system; at each time step of the simulation the tokens are added/removed from one place to the other according to the arc connections and the transition type.
- *Transitions* (T) – rectangular elements used to describe the data flow from one place to the other. At each time step of the simulation transitions can fire and change the status of the places. The Hybrid Petri Nets Model includes several transition types, the most used for Service Availability modeling are:
    - *Discrete* - tokens flow are added/removed as discrete values;
    - *Continuous* – tokens are managed as continuous (fractional) values;
    - *Stochastic* – tokens flow is managed according to a stochastic distribution along the time span of the simulation.
- *Arcs* (arrows) – connections between places and transitions, which describe the token flow conditions and directions.

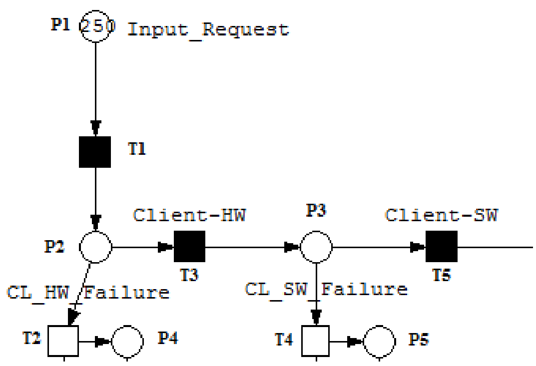

Fig. 5 – Detail of a Hybrid Petri Net

On Figure 4 a detail of an Hybrid Petri Net sample model for the user registration test case is reported. P1 represent the place of the input user registration requests (250 tokens are the input data flow). Client HW and SW components are represented by P2 and P3 places. T1, T3 and T5 are the discrete transitions that, firing at every time step, allow to the tokens (input requests) to move to the next place. The chain T1-T3-T5 represents the nominal sequence of the service: the input registration request is saved on HW and processed by Client SW.

T2 and T4 are instead stochastic transitions and represent the occurrence of a failure, according to an input distribution. When a failure occurs, the data (token) are removed from the operational sequence (P2 and P3) and a failure counter (P4 and P5) is updated. This simple mechanism allows to model a permanent failure for the Client component, according to the Service Failure definition reported on 3.1.4.

This means that the data contained on P2 at the moment of a failure occurring on T2 will not be recovered and it will be considered as lost. In this very simple example, the failure is assumed to endure for a time step, but in the final model the failure time span has been easily expanded to endure for a specific time. Failure counters are fundamental, since they allow to monitor the failure trend of each component along a Montecarlo simulation, and to compute the internal metrics defined on Section III. On each Montecarlo simulation run the Service Availability is computed from the comparison between the number of tokens present at the initial and final place of the represented chain.

Fig. 6 Failure distribution per component

1) Failure Modeling

The proposed approach allows to model the failure occurrence for each HW/SW component through the Probability Density Function (PDF) of a stochastic distribution. For the specific test case a PDF of an exponential distribution was used:

$$F(t) = \lambda e^{(-\lambda t)} \tag{3}$$

where λ is the Failure Rate, computed as:

$$\lambda = \frac{1}{MTBF} \tag{4}$$

The adopted Failure rates are derived from MTBF values reported on Section 3.1.3. On Figure 5 it is possible to observe the failures distribution in a time interval of 30 days for a Montecarlo simulation of 50 runs.

2) Maintenance & Recovery

The SERA Model takes into account both permanent and transient failures; this aspect leaded to the introduction of a system chronology and a maintenance process. In the current study case the Petri Net elements have been used to build a clock to monitor system chronology and a simple maintenance process to test the effectiveness of the model on real conditions. For each component affected by transient failures (Section III) the model verifies the time of the day at which a failure occurs; if the failure occurs in the time span between the 8:00 AM and the 18:00, it is assumed an immediate intervention of the maintenance, with a MDT of 4 hours. If instead the failure occurs in the time span between the 18:00 and the 8:00 of the following day, the intervention is delayed to the 8:00, with a MDT of 1 hour (nominal process will then be restored at 9:00). On both cases during the unavailability time window the received registration requests are collected in a queue, waiting for the component restoration. The request queue will then be run out according to the simulated processing times and remaining input requests to process. The introduction of this maintenance modeling is fundamental to reproduce the real conditions of a processing queue and the related delay into the final request release. It is important underline the logistic support can also drive the system design and operation phases; in this study case the Logistic Delay Times have been choose only to highlight as the failure recovery can influence the completed requests, fundamental requirement for service success. Further evaluations, such as the different logistic scenarios, will affect Service Availability will be discussed in the next studies.

3) *Simulation Scenario*

The current study was tested with a series of dedicated Montecarlo simulations using open-source Snoopy Petri Net Tool [17]. Snoopy proved to be efficient and adapt to the feasibility study, with some limitations into the missing possibility to use timed transition, which limited the modeling possibilities for the recovery times.

The SERA Petri Net model of the User Registration Service was implemented and tested on different time windows, from 30 days up to several years, in the case of realistic MTBF values. Please notice that for the current test case the time interval of each simulation was set to 30 days (720 h) at a time step of 1 h, with low MTBF values, with the declared purpose to put in evidence the method capabilities and limits of the model. In the model each HW and SW component can be affected by failures, according to the aforementioned modeling.

V. RESULTS

The results of the SERA Model are reported on Figure 6. A specific percentage value of Service Availability was computed for each Montecarlo run, showing the statistical trend due to the interaction of stochastic failures. The most remarkable result is that the computed value of S_A (average value: 97.94%) is very far from theoretical predictions (99.99%), based on a static view of system components. It is also relevant to notice that it is possible to characterize the S_A by its related distribution, in order to define a trend and derive an expected value. On the reported study case the S_A distribution shows that a statistically expected value of S_A is between 97% and 98%, a value decreasing if the distribution is computed for the S_{Ar}. These results show how the application of a SERA model based on Montecarlo simulations can provide a reliable estimate of the service availability, to be compared to the results from operational life.

Fig. 7 – Global & restricted Service Availability

The second remarkable result from simulations is the relevant difference occurring between the S_A and the S_{Ar}, especially in two worst cases where the random distribution of failures had a dramatic impact on the S_{Ar} value (<65% and <25% respectively). This result led to the conclusion that in a real condition of stochastic distribution of failures there is a significant probability (>1/50) to have at least a service availability value far below the commonly accepted standards (80% or higher). Most of all, this is not due to a failure of a specific component, but to the way the failures occur and at which time. On the S_{Ar} < 25% case, one failure occurred on DB-SW at 19:00, generating a large request queue; another failure occurred at 10:00, giving rise to a unavailability window of 16 h on a total of 17 h (from 9:00 to 10:00 the component was available). In other terms, the results show how, in a real stochastic distribution of failures, the maintenance policy can have a significant weight on the final service availability performance. Such a result is fundamental especially in a system design phase, where the selection and the RAMS analysis of the single components is not the only element to take into account for the implementation of reliable services. The proposed method provided a refined picture of the reliability and availability of a service at a very reduced cost (one single Petri Nets), proving its flexibility and its effectiveness.

CONCLUSIONS

The traditional Prediction Model gives an Availability figure on steady-state (asymptotic condition) without taking into account time evolution and transitions on the state of the service. The proposed SERA Model allows to determine the failure distribution and the impact on the service outputs, taking into account the logistic support and the operative process. By Monte-Carlo runs it is possible to predict the Availability mean value and its distribution on a statistical basis, replicating the operational conditions. The proposed approach proved to be suitable especially as support method into system design, allowing to detect possible criticalities and to predict on a statistical basis a reliable value of Service availability.

ACKNOWLEDGMENTS

The authors thank L. Tirone, S. Sorge and System Engineering team from ASTER for the provided support and the constructive comments. The authors are grateful for support to A. Di Bona from Telespazio for his support into define the logistic support aspect according the O&M organization

REFERENCES

[1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

[2] V. Cortellessa, V. Grassi, Reliability modeling and analysis of service-oriented architectures, in Test and Analysis of Web Services. -2007

[3] M. del Mar Gallardo, J. M. P. Merino, G. Rodrıguez, Integration of Reliability and Performance Analyses for Active Network Services. Electronic Notes in Theoretical Computer Science - 2005

[4] M. Rout, P. Bhuyan, A Survey Report on Reliability Models and Frameworks in SOA, IJARCSSE-2014

[5] C. Xie, B. Li, X. Wang, A Web Service Reliability Model Based on Birth-Death Process. SEKE, 2011K. Elissa, "Title of paper if known," unpublished.

[6] Y. Dai , B. Yang , J. Dongarra , G. Zhang. Cloud Service Reliability: Modeling and Analysis -2010

[7] Y. Bai, H.Zang, F.Yangzhen, Reliability modeling and analysis of cloud service based on complex network. Prognostics and System Health Management Conference- 2016

[8] S. M. Iyer, M. K. Nakayama, and A. V. Gerbessiotis, A markovian dependability model with cascading failures," IEEE Transactions on Computers, vol. 58, pp. 1238–1249, September 2009.

[9] C.A.Ardagna, E. Damiani, R. Jhawar, V.Piuri A Model-Based Approach to Reliability Certification of Services Digital Ecosystems Technologies (DEST), 2012 6th IEEE International Conference on

[10] L.Jereb, Efficient Reliability Modeling and Analysis of Telecommunication Networks (1998)

[11] Thirumaran, M., et al. "Finite State Machine Based Evaluation Model for Web Service Reliability Analysis." *International Journal of Web & Semantic Technology* 2.4 (2011): 125.

[12] M. Srinivasa Rao, V. N Anaikan, Review of Simulation Approaches in Reliability and Availability Modeling. International Journal of Performability Engineering, Vol. 12 No. 4, July 2016

[13] A.Garro, A.Tundis. On the Reliability Analysis of Systems and Systems of Systems: the RAMSAS method and related extensions, IEEE Systems Journal, 9(1):232-241, 2015, ISSN. 1932-8184, IEEE Systems Council

[14] ECCS-Q-ST-30-09C Availability analysis. 31 July 2008

[15] David, R., & Alla, H. (2010). *Discrete, continuous, and hybrid Petri nets*. Springer Science & Business Media.

[16] Goss, P. J., & Peccoud, J. Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. *Proceedings of the National Academy of Sciences*, *95*(12), 6750-6755. (1998).

[17] Heiner, M., Herajy, M., Liu, F., Rohr, C., & Schwarick, M. (2012). Snoopy–a unifying Petri net tool. Application and Theory of Petri Nets 398-407.

.