



HAL
open science

A framework for modeling availability of E-business systems

Mohamed Kaâniche, Karama Kanoun, Mourad Rabah

► **To cite this version:**

Mohamed Kaâniche, Karama Kanoun, Mourad Rabah. A framework for modeling availability of E-business systems. 10th International Conference on Computer Communications and Networks, 2001 (ICCCN 2001), Oct 2001, Scottsdale, AZ, United States. pp. 40-45, 10.1109/ICCCN.2001.956217 . hal-01212216

HAL Id: hal-01212216

<https://hal.science/hal-01212216>

Submitted on 6 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Framework for Modeling Availability of e-Business Systems

Mohamed Kaâniche, Karama Kanoun, and Mourad Rabah

LAAS-CNRS — 7 Avenue du Colonel Roche
31077 Toulouse Cedex 4 — France
{kaaniche, kanoun, rabah}@laas.fr

Abstract—This paper defines a multi-level modeling framework for e-business system availability evaluation. It is based on two main steps: 1) hierarchical description of the system and its interactions with the users, from the functional and structural points of view, and 2) hierarchical construction and solution of the system availability models based on information from the first step.

I. INTRODUCTION

The explosive growth of Internet users has caused a dramatic increase in electronic markets with several categories of business [1, 12, 18]: business-to-business, business-to-customer, customer-to-customer, etc. In this context, the services provided to the users result from the cooperation of several systems that are interconnected and widely distributed over the Internet. Given the critical nature of many such systems, the assessment of the quality of service as perceived by the users is a key issue for e-business service providers. Dependability and performance related measures provide useful indicators for the designers to assess the impact of component failures (hardware, software, communication links) on the quality of service provided to the users. It is important to estimate such measures early in the design process to assist the designers in making objective architectural decisions. Indeed, by comparing several architectural solutions, they can identify potential areas of improvement of the e-business infrastructure to better fulfill user expectations.

Little support is provided today for modeling and evaluating the dependability of largely distributed systems from the user perspective. The objective of this paper is to present a multi-level modeling approach for evaluating the availability of e-business systems. Although this paper focuses on availability, the proposed framework should also allow the evaluation of other dependability and performability measures.

This paper is organized as follows. Section II describes the context of the work and the main key players involved in the provision of the services delivered by an e-business system. Sections III and IV present the multi-level modeling framework that we propose for availability evaluation from the user perspective. Finally, Section V concludes the paper.

II. PROBLEM STATEMENT

E-business applications are generally based on a complex distributed infrastructure, with multiple interconnected layers of software and hardware components. Three key players are typically involved in the provision of the services delivered by such applications (see Fig. 1):

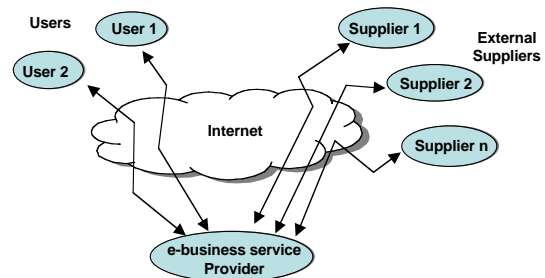


Fig. 1. E-business key players.

- 1) The *users*, (i.e., the customers) who interact with the e-business provider site(s) to invoke a set of e-business functions.
- 2) The *e-business service provider*, who implements the e-business functions that can be invoked by the users; these functions are based on a set of services and resources that are internal to the e-business service provider site(s) or are provided by external suppliers.
- 3) The *external suppliers*, who contribute to the implementation of some of the functions and services delivered by the e-business service provider to its users.

Every transaction initiated by a user is processed in several steps. In particular, it starts in the user's applications, flows through the Internet, executes through the business applications of the provider, and generally through the applications of external suppliers. For example, an e-business service provider can offer a book selling electronic service by outsourcing shipping, payment, and billing to other service providers (external suppliers). At the provider level, the user requests and the interactions with the external suppliers are supported by a set of complex distributed applications and middleware such as Web servers, application servers and database servers. Also, similar infrastructures are used at the external supplier sites.

To model the availability of e-business systems as perceived by the users, it is necessary to identify the main functions and services provided to the users and the main resources contributing to their accomplishment. Based on this, one can build a model to assess the impact of component failures and repairs on the quality of service delivered to the users.

Different users may exhibit different behaviors and therefore may invoke the various e-business functions in different ways and with different frequencies. Some users may be heavy buyers while others may do extensive searching and browsing but very seldom buy from the

provider site(s). Thus, the types of functions invoked and the resources involved in the accomplishment of these functions are not necessarily the same. As a consequence, the availability of the e-business systems may be perceived differently by these classes of users. Indeed, the user perceived availability, is influenced by the user operational profile (i.e., workload) and the state of the components involved in the accomplishment of the functions invoked by the user. The latter may be influenced by several factors including network related failures, hardware or software component failures affecting the architecture of the provider or external suppliers.

Generally, the e-business service provider has a full control of its own architecture. Therefore, a detailed availability modeling and analysis of this architecture can be carried out to support design architectural decisions. Different variants of this architecture can be modeled and compared with respect to the availability objectives to be fulfilled.

The discussion above shows that several issues should be taken into account when modeling the availability of e-business systems as perceived by their users. Due to the complexity of the target system, a systematic and pragmatic approach is needed to support the construction of availability models. Hierarchical modeling is well suited to alleviate the problems of model specification and solution. It consists in describing the target system at different abstraction levels, with a submodel associated to each level. The submodels are composed hierarchically. Each submodel incorporates parameters that result from the processing of lower level models.

In this paper, we propose a framework for the hierarchical modeling of e-business systems availability as perceived by the users, based on two main steps: 1) hierarchical description of the e-business system and its interactions with the users, from the functional and structural point of view, and 2) hierarchical construction and solution of the e-business system availability model based on the information provided by the first step. These two steps are described in Sections III and IV, respectively.

III. E-BUSINESS SYSTEM HIERARCHICAL DESCRIPTION

The information needed to analyze the e-business behavior from the user perspective can be structured into four levels presented in Fig. 2. The first level describes how the users interact with the e-business system, and the three remaining levels detail how the user requests are implemented on the e-business system infrastructure.

- The *user level* describes the user operational profile in terms of the types of e-business functions invoked and the probability of activation of each of them.
- The *function level* describes the set of functions available at the provider site(s).
- The *service level* describes the main services needed to implement each function and the interactions among them. Two categories of services are distinguished: those provided by the e-business service provider and those provided by external suppliers.
- The *resource level* describes the architecture on which the services identified at the service level are implemented.

mented. At this level, the architecture and the fault tolerance and maintenance strategies implemented at the provider site(s) are detailed.

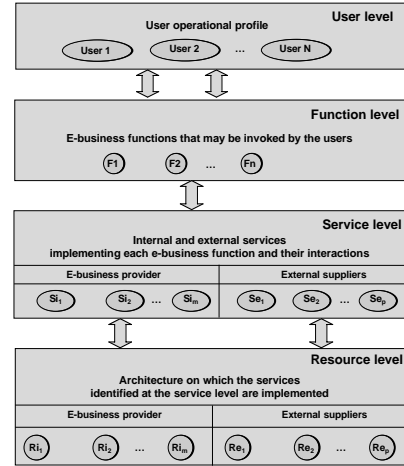


Fig. 2. E-business system hierarchical description

The above hierarchical description builds on some concepts proposed in [13] to evaluate the performance of e-business applications. However, as our framework focuses on availability modeling and evaluation from the user perspective, we have adapted these concepts and refined them to fulfill the objectives of our study.

In the following subsections, we present each of these levels and place emphasis on how this hierarchical description will help in modeling the availability of e-business systems in a hierarchical way.

III.1 User Level

This level describes the user operational profile in terms of the types of e-business functions invoked and the probability of activation of each of them. The operational profile can be characterized by describing all the execution scenarios performed by the user when visiting the e-business provider site(s). Each scenario is described by a path defining the set of functions invoked, the order of execution of these functions, and the probability of activation of each function in the corresponding scenario. Each path is defined by a set of nodes and transitions. The *Start* and the *Exit* nodes correspond to the beginning and end of a user session when visiting the provider site(s). Each node F_i means that function F_i is invoked by the user. A transition from node F_i to node F_j means that function F_j is executed after execution of F_i . The associated conditional probability is p_{ij} . The output transitions from the start node, and the corresponding probabilities p_{si} , specify the first function executed by the user when entering the e-business service provider site(s). Finally, the parameters p_{ix} specify the probability of leaving the e-business site after executing function F_i . Parameters p_{ij} associated with the transitions can be estimated based on general techniques used to build operational profiles for computer systems (see e.g., [14]).

Such representation is general. The various paths may be represented in a compact form using a matrix

representation. In particular, Fig. 3 gives an example of a user operational profile described as a Markov chain with the corresponding probability transition matrix.

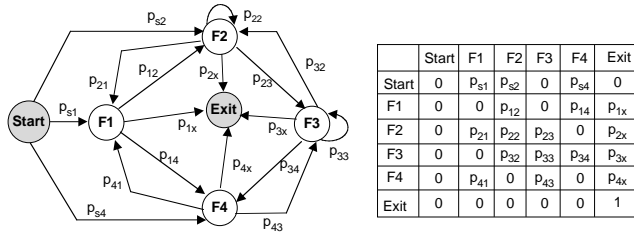


Fig. 3. User's operational profile

The e-business system availability as perceived by the users can be evaluated by considering a particular path or, by taking into account all the paths from the start node to the exit node. The availability measure will be affected by the probability of the corresponding path(s) and the availabilities of the functions involved in these paths. It is worth noting that different models can be defined to characterize different classes of users.

III.2 Function Level

This level identifies the set of functions offered to the users at the e-business service provider site(s). Table 1, extracted from [13], gives some examples of such functions. Some of these functions (e.g., Search, Login) may be found in most e-business sites, whereas others are characteristic of certain e-business sites or of specific types of e-business sites.

TABLE 1:
EXAMPLES OF FUNCTIONS PROVIDED BY E-BUSINESS SITES

Category	Function	Description
Common	Login	Login to the site
	Register	Register as a new user
	Search	Search site database
	Select	Show one of the results of a search
	Browse	Follow links within the site
	Retail	Add Item
	Remove Item	Remove item from shopping cart
	Create Registry	Create a gift registry
	Add to Registry	Add item to gift registry
	Check Status	Check status of previous order
	Pay	Pay for items in shopping cart

The identification of all functions offered by the e-business site and the classification of these functions according to their criticality require a thorough analysis of the e-business specification and the expectations of the users in terms of quality of service. Different levels of degradation of the quality of service delivered to the users can be defined based on the assessment of the impact of temporary loss or degradation of each function and the cost (e.g., loss of revenue) caused by such events. Such a classification should also take into account the impact of the loss or degradation of several functions.

III.3. Service Level

This level describes the mapping between the e-business functions and the services needed to implement them. Each e-business function identified at the function level is

decomposed and refined into a set of services implemented by various software entities (i.e., servers). Examples of servers include Web, Application, Authentication, Name, File, Database and Communication servers. Generally, the execution of one function may involve more than one server. Based on the analysis of client-server interactions, we can define a matrix specifying the mapping between the functions identified at the function level and the servers identified at the service level. Table 2 gives an example of such a mapping.

TABLE 2
EXAMPLE OF MAPPING BETWEEN FUNCTIONS AND SERVERS

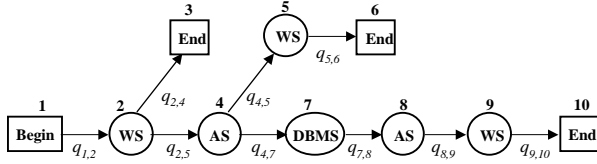
	Web Server	Authentication Server	Application Server	Database Server
Login	✓	✓		
Register	✓	✓		
Search	✓		✓	✓
Select	✓		✓	✓
Browse	✓		✓	✓

Table 2 presents a static view of the link between the functions and servers. It does not specify clearly if all servers are needed for each execution or if only subsets of them are needed for different execution scenarios. The latter are obtained from the analysis of the dynamic interactions among the servers during function executions. Several graphical notations and formalisms can be used to describe the dynamic interactions and dependencies among the servers that implement an e-business function (see, e.g., [3, 4, 17]). The graphical representation given in Fig. 4 is based on the concept of the *Interaction Diagram* defined in [13]. The interaction starts and ends with the client node ("Begin" and "End" nodes). Each path between a pair of client nodes identifies the set of servers involved in the interaction.

Fig. 4 presents three possible scenarios for the execution of the "Browse" function. The nodes are numbered for the sake of clarity and each arc between two nodes i and j is labeled with the probability of occurrence of the corresponding transition (denoted $q_{i,j}$). The three scenarios are described as follows:

- 1→2→3: the user sends a request to the Web server (node 2). The Web server refuses a connection and the request fails. The user (represented by node 3) receives a reply message. This marks the end of this interaction.
- 1→2→4→5→6: the web server accepts the request from the user and sends it to the application server (node 4) which returns a dynamically generated page to the Web server that replies to the user (node 6). The database is not involved in this case.
- 1→2→4→7→8→9→10: the application server requires some specific data items, it will send a request to a database server (node 7). After the database server has answered the application server, the latter will generate an HTML page for the web server, which is then forwarded to the user.

All paths in the interaction diagram, from a "begin" to an "end" node, should be accounted for in the evaluation of the availability of the corresponding function.



WS: web server AS: application server DBMS: database management server

Fig. 4. Interaction diagram for the Browse function

III.4 Resource level

This level describes the mapping between the services defined at the server level and the resources involved in the achievement of these services. Also, it provides information on the replication of each service as well as the fault tolerance and maintenance strategies implemented at the e-business service provider site(s). A resource is a component system or an element of a component system (computer host, hardware and software components, communication link) that contributes to the implementation of e-business services. Indeed, one service may be partitioned and replicated among several resources or clusters of resources and one resource may host many services. At this level, we distinguish between internal and external services.

As the architecture on which the *external services* are implemented is not known, we associate to each of them a single resource that is considered as a black box. For example, an Internet service provider can be represented by a single resource providing connectivity service.

As regards *internal services*, a detailed analysis of the e-business provider site(s) architecture can be performed. We must define the mapping between the resources and the services, as well as the interactions among these resources, since the availability of each service will depend on the availability of the corresponding resources. Table 3 gives a simple example for illustration: three resources (computer hosts A, B and C) are involved in the provision of the web service, while the application service and the database service are implemented on host D. All these resources are connected through a Local Area Network (LAN).

TABLE 3
EXAMPLE OF MAPPING BETWEEN INTERNAL SERVICES AND RESOURCES

	Host A	Host B	Host C	Host D	LAN
Web Service	✓	✓	✓		✓
Application Service				✓	✓
Database service				✓	✓

Several alternative architectural solutions may be considered for implementing the internal services. These solutions may be defined based on:

- various organizations of the services on the hardware support (e.g., dedicated hosts for each server, vs. multiple servers on the same host);
- various fault tolerance strategies (non-redundant servers vs. replicated servers);
- various maintenance strategies adopted by the e-business service provider (e.g., immediate maintenance vs.

delayed maintenance, dedicated vs. shared repair resources).

The alternative architectures may be compared to help the designers in the selection of the most appropriate solution from the availability point of view. Each architecture should lead to the definition of the mapping between the corresponding resources and the services implemented on these resources.

Knowledge of the system architecture is required for modeling purposes. Fig. 5 presents examples of configurations of a Web server: a) a non-redundant configuration with a single server, b) a redundant configuration with geographically distributed replicas, and c) cluster-based configuration with several Web servers interconnected through a LAN and centralized at a single site with a load balancer directing incoming requests to one of the servers. Configuration (b) requires the replica states to be kept mutually consistent to ensure that clients do not get out-of-date information. This is not easy to achieve on a large-scale system [6]. Alternative solutions are proposed for instance in [2] to ensure a weak coupling between functions implemented on geographically distributed servers. Current approaches are generally based on the cluster configuration (c). However, the load balancer is a single point of failure. Therefore, a fault tolerant machine with error detection and recovery capabilities should be considered for the load balancer. Also, fault tolerance solutions should be used to ensure reliable data and reliable communication among the Web servers.

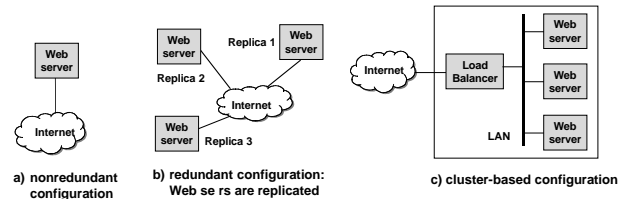


Fig. 5. Example of configurations for a Web server

For each service, different levels of service degradation may be defined depending on the state of the resources involved in its accomplishment. For instance, when a service is distributed on several computer hosts, the level of degradation can be defined as a function of the number of hosts that are operational. Architecture analysis and availability modeling at the resource level provides the information that is needed to define service degradation levels and to establish the link between the state of the resources and the corresponding degradation levels.

IV. AVAILABILITY MODELING AND EVALUATION

The hierarchical modeling approach, represented in Fig. 6, is directly related to the hierarchical description of the e-business system given in Fig. 2. It has been defined in such a way that the outputs of a given level are used in the next immediately upper level to compute the availability measures associated to this level. Accordingly, at the service level, the availability of each service is derived based on the availability of the resources involved in its accomplishment. Similarly, at the function level, the availability of each function is obtained from the availability of the services

implementing it. Finally, at the user level, the availability measures are obtained based on the availability measures of the functions invoked by the user.

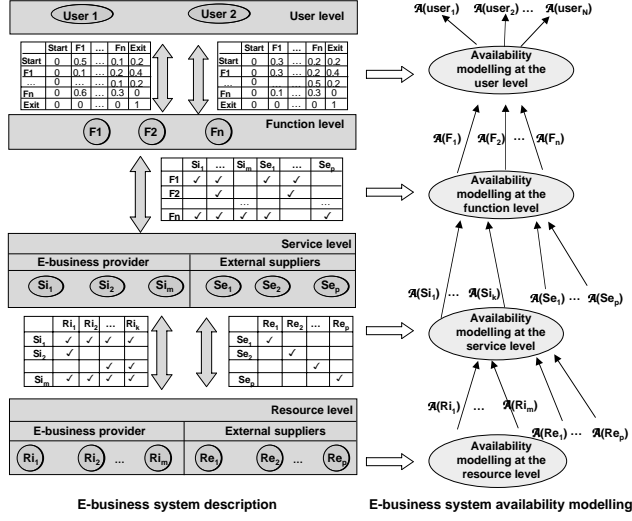


Fig. 6. Hierarchical availability modeling

Considering the service level, we made a distinction between internal and external services. The latter are delivered by providers for whom only little information is known. It is however expected that specific experiments or measurements such as those performed in [8, 10, 11] will provide the availability measures $\{\mathcal{A}(Se_j)\}$. It is assumed that external services are independent.

Internal services are supplied by the resources of the provider. Depending on the infrastructure of the provider site(s), some services may be implemented on the same resources, some of the resources may share the same repair facilities and some services or resources may be replicated or fault-tolerant. As a consequence, strong dependencies may exist between the services due to dependencies between the resources. For such infrastructures, it is more efficient to consider the service and the resource levels together, the corresponding level is simply denoted as the service/resource level.

At the **service/resource level**, one or several availability models are built based on the knowledge of the e-business infrastructure and the resources implementing the expected services, together with the fault tolerance and recovery mechanisms, and the maintenance policies at the provider site(s). Different techniques may be used to build and solve these availability models, including combinatorial techniques (e.g., fault trees, reliability block diagrams), and state-based techniques (e.g., Markov chains, Generalized Stochastic Petri Nets, GSPNs) [16, 19]. The selection of the right technique to be used mainly depends on whether i) the services are stochastically independent or not, and ii) the resources are independent or not. Markov chains and GSPNs are well suited to evaluate the service availability in the presence of strong dependencies. In particular, the modeling approaches described in [5, 9, 15] that have been successfully applied to complex real-life systems, can be used to take into account explicitly the stochastic dependencies that might exist between the

various components of the service/resource availability model (see [7] where examples of models at the resource level are given).

The outputs of this modeling step are the availability of the various internal services $\{\mathcal{A}(Si_j)\}$.

The availability model at the **function level** is based on the knowledge of the availability of all services involved in function accomplishment, along with the matrix giving the mapping between Functions and Services and the path probabilities derived from the interaction diagram as defined in Section III.3. The outputs of this level are the availability of the various functions $\{\mathcal{A}(F_i)\}$ that can be defined as follows.

$$\mathcal{A}(F_i) = \sum_{j=1}^M \pi_j \mathcal{A}(\sigma_j(F_i))$$

where:

- M is the number of execution paths for function F_i in the interaction diagram
- π_j is the probability of activation of execution path j
- $\sigma_j(F_i)$ is the set of servers involved in execution path j
- $\mathcal{A}(\sigma_j(F_i))$ is the availability of the servers involved in execution path j

This formula is general and can be applied whether the services are independent or not. When the services are independent, $\mathcal{A}(F_i)$ can be expressed as:

$$\mathcal{A}(F_i) = \sum_{j=1}^M \pi_j \prod_{p \in \sigma_j(F_i)} \mathcal{A}(S_p)$$

where $\mathcal{A}(S_p)$ is the availability of server S_p involved in execution path j .

At the **user level**, the availability model for a given user class is based on the knowledge of the execution paths followed by the user when visiting the e-business provider site(s) (described in the model characterizing the user operational profile, as defined in Section III.1) and the availability of the functions invoked by the user in each path. The outputs of this level are the availability as seen by the various classes of users $\{\mathcal{A}(user_k)\}$.

Similarly to the function level, $\mathcal{A}(user_k)$ can be obtained by the following formula:

$$\mathcal{A}(user_k) = \sum_{i=1}^N \beta_i \mathcal{A}(L_i)$$

where:

- N is the number of paths in the Markov chain describing the user operational profile
- β_i is the probability of activation of path i
- L_i is the set of functions involved in path i
- $\mathcal{A}(L_i)$ is the availability of functions involved in path i

In particular, when the functions are independent:

$$\mathcal{A}(user_k) = \sum_{i=1}^N \beta_i \prod_{q \in L_i} \mathcal{A}(F_q)$$

where $\mathcal{A}(F_q)$ is the availability of function F_q executed in path i .

VI. CONCLUSION

The quest for the construction of a dependable electronic business starts with the definition of the e-business provider architecture and the identification of the internal services as well as the possible external services complementing them. We have defined a hierarchical framework for an e-business availability modeling taking into account the infrastructure of the provider site(s), the availability of the external services and the user's profile. We have shown how to describe and model progressively an e-business system.

The primary problem when building availability models for complex systems is related to state explosion. Techniques addressing this problem are of two categories: "largeness avoidance" and "largeness tolerance". Largeness avoidance techniques try to circumvent the generation of very large models by constructing sub-models that can be processed in isolation; they are efficient when the sub-models are loosely coupled. Largeness tolerance techniques are more appropriate for systems with multiple and complex interactions between components. From a practical point of view, our multi-level modeling approach combines the two categories of models. We advocate a largeness avoidance technique at the levels where independence or weak dependency assumptions hold, and we recommend largeness tolerance techniques for constructing the sub-models that exhibit strong dependencies (in particular at the resource and service levels). Indeed, even with the proposed approach, large and complex availability models cannot be totally avoided due to the large number of elements and the complexity inherent to e-business systems. The aim of our approach is to develop independent models as far as possible and combine the results. Both analytical and simulation based techniques can be used to solve the availability models defined within the multi-level modeling approach. In particular, simulation techniques should be used when the models include particular assumptions that cannot be handled with traditional analytical techniques.

Certainly, we have not covered all topics that need to be addressed when designing an e-business service provider site, but we put emphasis on how to decompose the main problems to solve them in an easier manner than considering all levels at the same time. We have shown the kind of calculations and modeling that are required for the various steps.

ACKNOWLEDGMENT

The work presented in this paper has benefited from fruitful discussions with Jean-Claude Laprie and David Powell. This work is partially supported by the European Community (Project IST-1999-11585: DSoS — Dependable Systems of Systems).

REFERENCES

- [1] Y. Bakos, "The Emerging Role of Electronic Marketplaces on the Internet", *Communications of the ACM*, vol. 41, pp. 35-42, 1998.
- [2] N. Bowen, D. Sturman, T. Liu, "Towards Continuous Availability of Internet Services Through Availability Domains", *The International Conference on dependable systems and Networks*, (New York), pp. 559-566, IEEE Computer Society, 2000.
- [3] D. Caswell, S. Ramanathan, "Using Service Models for Management of Internet Services", HP Laboratories Palo Alto HPL-1999-43, 1999.
- [4] C. Ensel, "Automated Generation of Dependency Models for Service Management", *Workshop of the OpenView University Association (OVUA 1999)*, (Bologna, Italy), 1999.
- [5] N. Fota, M. Kââniche, K. Kanoun, "Dependability Evaluation of an Air Traffic Control Computing System", *Performance Evaluation*, vol. 35 (3-4), pp. 553-573, 1999.
- [6] D. B. Ingham, S. K. Shrivastava, F. Panzneri, "Constructing Dependable Web Services", *IEEE Internet Computing*, pp. 25-33, 2000.
- [7] K. Kaâniche, K. Kanoun, M. Rabah, "A Preliminary Framework for SoS Dependability Modelling and Evaluation", DSoS Project, IST-1999-11585, LAAS-CNRS Report n°01157, 2001.
- [8] M. Kalyanakrishnan, R. K. Iyer, J. U. Patel, "Reliability of Internet Hosts: a Case Study from the End User's Perspective", *Computer Networks*, vol. 31, pp. 47-57, 1999.
- [9] K. Kanoun, M. Borrel, T. Morteveille, A. Peytavin, "Availability of CAUTRA, a Subset of the French Air Traffic Control System", *IEEE Transactions on Computers*, vol. 48, pp. 528-35, IEEE Computer Society, 1999.
- [10] D. Long, A. Muir, R. Golding, "A Longitudinal Survey of Internet Host Reliability", *14th Symposium on Reliable Distributed Systems (SRDS-95)*, (Bad Neuenahr, Germany), pp. 2-9, 1995.
- [11] V. Machiraju, M. Dekhil, M. Griss, K. Wurster, "E-services Management Requirements", HP Laboratories Palo Alto, CA, USA HPL-2000-60, 2000.
- [12] P. Maes, R. H. Guttman, A. G. Moukas, "Agents That Buy and Sell", *Communications of the ACM*, vol. 42, pp. 81-91, 1999.
- [13] D. A. Menascé, V. A. F. Almeida, *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.
- [14] J. Musa, *Software Reliability Engineering*: McGraw-Hill, 1998.
- [15] M. Rabah, K. Kanoun, "Dependability Evaluation of a Distributed Shared Memory Multiprocessor System", *3rd European Dependable Computing Conference (EDCC-3)*, (Prague, Czech Republic), pp. 42-59, Springer, 1999.
- [16] A. Reibman, M. Veeraraghavan, "Reliability Modeling: An Overview for System Designers", *IEEE Computer*, vol. April, pp. 49-57, 1991.
- [17] G. D. Rodosek, T. Kaiser, "Determining the Availability of Distributed Applications", *IFIP/IEEE International Symposium on Integrated Network Management*, (San Diego, CA, USA), pp. 207-218, Chapman & Hall, 1997.
- [18] S. S. Y. Shim, V. S. Pendyala, M. Sundaram, J. Z. Gao, "Business-to-Business E-Commerce Frameworks", *Computer*, pp. 40-47, 2000.
- [19] K. S. Trivedi, B. R. Haverkort, A. Rindos, V. Mainkar, "Techniques and Tools for Reliability and Performance Evaluation: Problems and Perspectives", *7th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, pp. 1-24, Springer-Verlag, 1994.