A Unified Performability Evaluation Framework for Computer and Communication Systems

Aad P. A. van Moorsel
Boudewijn R. Haverkort

Tele-Informatics and Open Systems
University of Twente
P.O. Box 217, 7500 AE, Enschede, The Netherlands
moorsel@cs.utwente.nl

Abstract

In this paper we discuss a unified approach towards model-based quantitative evaluation of both computer systems and communication systems. In the area of fault-tolerant computer systems, dependability evaluation has been recognized as being a topic of importance, both to judge a system on its merits and to provide trust in the actual dependability of the system. In communication systems, the need for identifying and evaluating quality of service parameters is becoming more and more apparent because of increasing demands on for instance speed and availability. In this paper we construct a framework, the so-called *performability evaluation framework*, within which the quantitative evaluation of both types of systems can be discussed. The framework closely resembles the performability framework introduced by Meyer [18]. We present a general view, a system view and a modelling view on performability evaluation, leading to a framework which naturally fits known measure definitions, modelling methods and solution techniques. We will especially discuss the importance of evaluating *useful* performability measures. In this respect we will distinguish between *system measures* and *task measures*. Task measures directly relate to evaluation of the service, and we will argue that in performability evaluation task measures should be evaluated. We relate the performability evaluation framework with known concepts in both the areas of computer and communication systems.

1. Introduction

In this paper we develop the so-called *performability evaluation framework* (PEF). The goal of the PEF is twofold. Within the PEF we motivate our approach to model-based quantitative evaluation of computer and communication systems and it will form a

framework within which modelling methods, definitions of measures and solution techniques can be discussed.

We construct the PEF in three phases. First, we identify that our first aim should be to do *meaningful* quantitative evaluation. This gives us the basic definition and motivation of performability evaluation. Secondly, we create abstractions from the application area of computer and communication systems by defining concepts such as system, service and user. In terms of these concepts we will redefine performability. The third step consists of a model definition, in which we distinguish between system and service (or task) issues. Again, we will make the definition of performability more specific by reformulating it in terms of the modelling concepts. Relevant measures and modelling and solution techniques can be discussed in the resulting framework.

One will recognize in this paper several ideas similar to earlier suggested related concepts and frameworks. In the area of fault-tolerant computer systems, the concern of evaluating the so-called *dependability* of the system has lead to a conceptual framework of Laprie [17], and to the performability framework of Meyer [18] [20]. Especially, the concept of *performability* evaluation as introduced by Meyer is closely related to the presented PEF. This explains why we have adopted the term performability for our framework. Our starting point is somewhat different, however. In [18] the need for a more expressive measure than pure reliability motivated the definition of the combined performance/reliability measure “performability” to evaluate the quality of service of degradable systems. In this paper we take a more global view and try not to identify what measure is actually of interest, but what *can* be of interest in general. However, the basic standpoint that the interest should be in doing *meaningful* quantitative evaluation, is in our opinion similar to that of Meyer [18].
In the field of communication systems, despite its much older tradition of using quantitative methods, there is a less coherent view towards quantitative system evaluation than in the field of fault-tolerant computer systems. Currently, considerable attention is paid to *Quality of Service* (QoS) aspects of communication systems, in CCITT [5] [6] [7] [8] [9], in ISO bodies [14] [15] and in various other places, such as in a project like QOSMIC [25]. Evaluation of the QoS has been recognized to be of major importance in the design and development of telecommunication systems. In particular in the area of broadband communication the quantitative aspects of a design are crucial in assuring its success. ATM technology is a case in point here, see e.g., [4] [11] [16]. Nevertheless, a solid framework as in the fault-tolerant computer systems context does not yet exist. We note here that Meyer identified in [19] the possible role performability evaluation, and especially the performability measure, can play in communication systems besides its role in degradable computer systems.

In this paper performability evaluation aspects will be discussed in an informal manner which will leave room for discussion at many places. We like to stress that we do not claim to present the one and only view at the discussed matter. We have tried to be precise, albeit informal, and want to present a framework in which model-based

performability evaluation of computer and communication systems can be fruitfully discussed.

This paper is organized as follows. First, we will develop the PEF in Section 2. In Section 3 we discuss how dependability issues in computer systems and Quality of Service issues in communication systems relate to the PEF. Section 4 states conclusions on the developed PEF.

2. Performability Evaluation Framework

In this section the performability evaluation framework is constructed in three steps. In Section 2.1 it first is defined what we in general mean with performability evaluation. This is not only a basic definition, but even more a basic motivation for doing performability evaluation. Section 2.2 then discusses the application area of computer and communication systems. We will introduce model concepts for these systems in terms of which we redefine performability. Based on this system view we develop in Section 2.3 a modelling framework within which model-based performability evaluation of computer and communication systems can be discussed.

2.1. Performability Evaluation: Basic Definition

Let us start the discussion by defining performability evaluation as follows:

Performability evaluation *is the meaningful quantitative evaluation of computer and communication systems.*

In this definition, performability (for convenience we will often leave out the suffix evaluation) is presented as a specific form of quantitative evaluation. For this reason we first specify what the scope is of the term quantitative evaluation. Quantitative evaluation comprises the evaluation of the *behaviour* of a system. We realize that it is required to define behaviour more precisely, but will postpone this until Section 2.2, in which a model for the considered systems is presented. For the moment we take this loose definition of quantitative evaluation, to indicate that we want to exclude elements as computing costs of systems.

For the preceding definition of performability evaluation we especially want to stress the implications of the adjective *meaningful*. Doing quantitative evaluation is only of use when the obtained results give the information that is desired. This seems a trivial matter, but implies first of all that the derived results should be representative for the system, thus having consequences for the freedom in modelling in case of model-based evaluation. The consequences of the term meaningful especially come forward in the choice of the measure. Often it is not at all easy to come up with measures that are really of interest. The user of a telecommunication service might for instance be interested in the quality of a video image, in which it does not want too many disturbances to occur. Defining a quantifiable measure that relates to this subjective quality demand is not trivial at all, but nevertheless is of greatest importance.

The restriction of the definition of performability evaluation to computer and communication systems is a matter of expertise and professional interest in a particular application domain. Most probably more types of systems can be put under the concepts we come up with in this section. Within the general class of man-made systems [13] for instance, other types of systems (e.g., flexible manufacturing systems) can be found for which this kind of discussions is in place. Note that the preceding definition of performability relates to the complete system. However, the analysis of components or sub-systems is intended to fall under performability evaluation as well.

Goals of performability evaluation

To motivate the evaluation itself, we discuss what the use is of performability results. In some cases the evaluation itself will provide all the desired information. For instance, when one wants to guarantee a blocking probability of messages in a communication system of less than 10^{-9}, the result of the evaluation directly gives the answer. However, in many cases an extra step is desired. One often want to be able to interpret the consequences of the outcome for the system, for instance in order to understand bottlenecks or optimize with respect to parameter settings. Then an extra step is necessary, which we have denoted 'planning' step in Figure 1. In case of for instance optimization one might want to use specific techniques such as perturbation or sensitivity analysis to optimize with respect to some system parameter. These topics are beyond the scope of this paper, our interest is basically in evaluation methods and techniques.

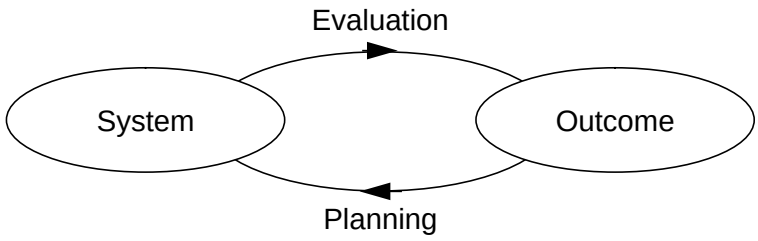

Figure 1. Evaluation in relation to planning.

Whatever is the aim of the performability evaluation, and whatever measure is considered to be meaningful, there can be different levels of accuracy be desired regarding the performability results. On one end there are 'strict' performability results, in which case an exact figure is desired as outcome. For instance, one might be interested in the exact mean waiting time in a switching node of a communication network to be able to guarantee a level of performance to the users. On the other end, 'loose' performability results can be desired, especially when the aim is to derive trends, with respect to some parameter. For instance, one might like to show some phenomena, like the existence of an optimal number of slots with regard to the mean waiting time of jobs in slotted-ring stations. In both cases, we are interested in the same measure, namely the mean waiting time, but the importance of the accuracy is different. In between the two extremes there exists all kind of different degrees of accuracy that might be desired. For instance, worst case results might be appropriate, or performability bounds might

be desired. The extent of accuracy is important for choices regarding modelling and solution techniques.

2.2. Performability Evaluation: A System View

In this section we present a more specific definition of performability evaluation, in which we incorporate a conceptual abstraction of the systems we study. Computer and communication systems will be discussed as systems of the same type, without losing aspects which are of importance for quantitative evaluation. We will first introduce the concepts of system, service, tasks and user, and then define performability evaluation in this system-oriented terminology.

Basic terminology

We assume the following definition of service, identical to the one in [17] for fault-tolerant computer systems. The **service** provided by a system is its behaviour as it is *perceived* by its users. A **system** is considered as an entity that interacts with other systems. The **user** is a system which interacts with the system that delivers the service, the service **provider**. In Figure 2 we have illustrated the concept of service provider

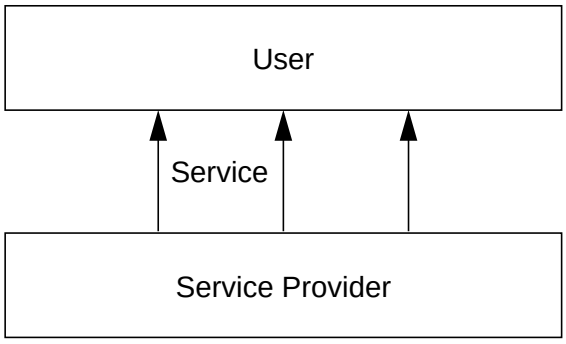

Figure 2. Basic view on system behaviour.

and user. The service consists of performing **tasks**, often in the form of messages to be delivered, or computational jobs to be carried out.

The above very general terms leave room for discussion. We do not want to elaborate on it too much, as it is merely our intention to introduce concepts that are useful for constructing the performability evaluation framework. What is important in the above definitions is that we identify an interaction between user and provider in the form of a service. This service will be evaluated from the user point of view, as we look at the service as *perceived* by the user.

The user can be of various types; it can be another physical system, to be called an **intermediate user**, but also a **human end-user**. The requirements of a human user will often decide which metrics are relevant for quantitative evaluation. The third type of user that can be defined is the provider of the service. The **provider** is the instance that provides the service, e.g., a telephone company. Although it might be confusing to name the provider a user of its own system, we do so because the provider is one of the

parties that decides how the service of a system should be. Note that we have used the term service provider in two different meanings, but in case of possible confusion we will specifically declare what is intended.

The service should fulfil the **service requirements** of the different users. The requirements of a human end-user might very well be subjective in nature, and might be difficult to quantify. For example, the interest of the human user might be to have a clear video image. What this implies in terms of quantifiable measures as bit error rates is not simple to identify. However, we stress the importance to come up with a meaningful measure. The requirements of an intermediate user, which is another physical entity, might be easier to quantify. E.g., one can think of a memory or CPU unit that is used by terminal equipment. Simple mean sojourn time requirements can be meaningful in this case. Bear in mind, however, that these requirement are usually derived from requirements of the human end-user which are harder to identify. Finally, the provider will have its own service requirements. These usually will be different from the end-user requirements and might very well be of an economical nature. When discussing the design of computer and communication systems, the requirements of a service are often given as part of a service **specification**.

Requirements classification

Although requirements can be very intricate or even subjective in nature, we find it useful for our discussion on performability evaluation to distinguish between three types of requirements (see also Figure 3):

- Functional requirements;
- Performance requirements;
- Dependability requirements.

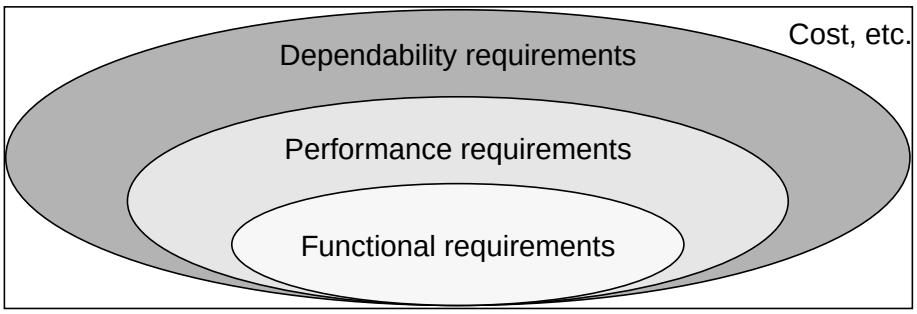

Figure 3. The relation between the different types of user requirements.

In a performability study the more intricate user requirements can be reduced to requirements falling in these three basic classes. The requirements we consider directly relate to the behaviour of the system. Other requirements exist, such as cost restrictions, legal restrictions, etc., as denoted in Figure 3 by the white box surrounding the three types of requirements. However, we will not discuss these various requirements

in more detail and assume that requirement are put in such a way that they can be met within the bounds given by these factors as cost, etc.

The **functional** requirements are any requirements on the behaviour of the tasks, excluding timing considerations. A functional requirement can for instance be that some task can be successfully processed. The **performance** requirements do consider timing of the functional requirements, and are quantifiable. A task, for instance the sending of a message, might have to be performed within some period of time. We will call the functional and performance requirements **task** requirements. The **dependability** requirements do consider reliance of the service, usually with respect to the task requirements. In Figure 3 the overlapping ellipses denote that performance requirements are defined with respect to functional requirements, and that dependability requirements are defined with respect to the task requirements. We distinguish between timed and untimed dependability requirements. **Timed** dependability requirements do consider timed aspects, **untimed** dependability requirements do not. In this way we have a similar distinction as between functional and performance requirements. As an example of a timed dependability requirement, one can think of the requirement of *continuity of service* for a specific period of time. On the other hand, the impossibility to interfere with an established telephone connection, is an untimed dependability requirement.

Performability evaluation

Before giving a system-oriented definition of performability, let us make more specific what is meant by evaluation, when put in system terminology. Qualitative and quantitative **evaluation** both validate the provided service against the service specification. Qualitative evaluation is the evaluation of the functional requirements, and of untimed dependability requirements like security. Quantitative evaluation takes time into consideration, and therefore relates to the evaluation of performance requirement as well as to the evaluation of timed dependability requirements, i.e., continuity of service. In this paper we will use a somewhat wider definition of quantitative evaluation, and also include the evaluation of service *characteristics*. For instance, evaluating the mean waiting time of a system can often be of interest, also when it is not a strict requirement posed by the users. However, always we can put the characteristics we evaluate in the form of functional, performance and dependability elements.

We adopt the following system-oriented definition of performability:

Performability evaluation *is the meaningful quantitative evaluation of the service of a system.*

Although the term quantitative evaluation has been defined in the previous paragraph, we like to stress again that we specifically look at behaviour in *time* of the system. So, the quantification stems from the time considerations. For instance, issues of waiting time, related to a performance requirement, and availability, related to a dependability requirement, are performability issues. We have chosen in this definition to use the word system instead of computer or communication system. Basically, this

implies that one can do performability evaluation for any system that allows for an abstract definition in terms of system, service and user and for which performance and timed dependability requirements can be defined, e.g., flexible manufacturing systems.

2.3. Performability Evaluation: A Modelling View

In this section we present a modelling view on performability evaluation. In order to do meaningful evaluation we advocate a modelling approach in which a system and a task model are distinguished, and the mutual influence between these two is considered. It will become clear that we can naturally distinguish between system measures and task measures, and it will be argued that the actual interest should be in task measures as they directly represent the service one wants to evaluate. We also discuss the underlying stochastic processes of the model, while this section is concluded by a definition of performability evaluation in terms of this modelling framework.

System and task model

From the definition of performability evaluation in Section 2.2 it follows that we have to model the service of a system. It therefore is natural to divide the performability model in two parts, a *system model*, which models the system, and a *task model*, which models the service. The **system model** models the changes in state, i.e., the behaviour in time, of the system configuration. With system **configuration** we mean all relevant aspects of the system, excluding tasks. For instance, the number of non-failed system components could be a relevant description of the system configuration. The **task model** models the behaviour in time of the tasks. In other words, the actual service, which consists of tasks to be carried out, is modelled by the task model. Besides knowledge such as the arrival intensity of tasks, we need to have information about the system configuration to model the task behaviour. For instance, in order to compute the mean delay of messages in a communication system, system configuration knowledge is a prerequisite for modelling the sending of messages. In other words, there is a direct influence from the system model on the task model. On the other hand, the task model can influence the system model too. E.g., a temporarily high load can increase breakdown probabilities, or a high load can trigger components to become active. We thus see that modelling mutual influence is a necessity, as has been identified before in [12].

Task behaviour assumptions

We go into some more detail concerning the possible task behaviour patterns, to relate existing modelling approaches to our modelling framework. We distinguish between:

- No assumptions;
- Partial steady-state assumption for task behaviour;
- Full steady-state assumption for task behaviour, i.e., behavioural decomposition.

If one does not make assumptions on the interactions between the system and task model, one obtains the *fully detailed* model. For complicated systems this implies that the model will be complex, and the solution method will therefore generally have to be discrete-event simulation. One can, however, assume that directly after the moment some interaction between the two models occurs, the task behaviour is considered to behave as if in steady-state for the belonging system configuration. This is called *behavioural decomposition*, and is the basic assumption in many performability analysis, see e.g., [28]. One can also assume *partial* steady-state. An interaction between the models then first introduces a transient phase in which the task behaviour cannot be considered to be in steady-state, followed by a steady-state phase. The steady-state phase always lasts until the next interaction between the models. Based on this observation the fast simulation technique injection simulation has been developed [22], which has been applied for analysis of the fault-tolerance mechanisms of the FDDI token ring [23].

Task and system measures

We now will make a distinction between *task measures* and *system measures*, which we consider to be of special importance when discussing meaningful quantitative evaluation. A **task measure** is a metric in which a task characteristic can be found, e.g., the mean waiting time of a task, the blocking probability of tasks, the throughput, etc. So, task measures are directly related to the service, as the service is built of tasks. A **system measure**, on the other hand, is a metric in which system characteristics are identified, e.g., the fraction of time a system is operational, i.e., the availability. From analysing the system model in separation, only system measures can be obtained, while analysing a task model in separation results in task measures. Analysing models that cope with both system and task behaviour gives the opportunity to derive both types of measures.

We advocate that the ultimate interest should be in *task measures* rather than in system measures, as task measures provide information about the service, and it is the *service* we want to evaluate. As an example, the fact that a system is available or not is not of primary interest, it is the fact that for this reason no tasks can be processed that matters. Another example is the mean number of tasks in a buffer. Again, it is not this system measure that is of first interest, but the consequence that the buffer occupancy has on for instance the mean waiting time or the loss probability of the tasks.
Conversion from system measure to the actually relevant task measure is therefore of great importance. For some measures there exist laws that provide this conversion. For instance, Little's formula, relates mean number of customers (a system measure) with the expected waiting time (a task measure), and the PASTA rule, relates buffer

overflow probability (a system measure) with the probability that a task will be blocked. In one special case a direct conversion from system measure to the consequences for the tasks exists. This is when one only distinguishes between **proper** and **improper** service, i.e., no grades of the quality of service are distinguished. Then, the fact that a system is in the proper state implies that tasks can be carried out successfully, while the fact that a system is in the improper state implies the impossibility of carrying out some task. However, a clear distinction between proper and improper service cannot always be established. To cope with this, Meyer has developed his performability approach [18], and for this reason we do take more into account than the system behaviour alone in the construction of the PEF.

Underlying stochastic processes and measures

The model proposed in this section needs a mathematical representation in order to derive performability metrics from it. Basically, the behaviour in time of both the system and the tasks can be described by so-called *discrete-event* stochastic processes, see e.g., contributions in [13]. These processes allow only for changes of the state of the model at discrete points in time, and the state description at these discrete points in time completely determines the stochastic process.

Performability metrics can be represented as any real-valued function of some random variable M. This random variable can take values in a range R, for instance on the interval $[0, \infty]$ when R represents waiting times. Interest can for instance be in the mean $\boldsymbol{E}M$, the m-th moment $\boldsymbol{E}M^m$, or in the quantiles $\Pr\{M < \alpha\}$, with $\alpha \in R$. We can classify measures according to three criteria:

- the 'domain' of the measure;
- transient or steady-state measure;
- interval or point measure.

The *domain of the measure* says with respect to which element the measure is computed. We can identify **task-based** measures, i.e., measures that are expressed per task or derived for some specific task, and **time-based** measures, i.e., measures that are expressed per unit of time or for some instant of time. An example of a task-based measure is the mean waiting time of a task, while the availability, i.e. the fraction of time a system is up, is a time-based measure. The distinction between these two is useful for identifying which solution method is suitable to derive the performability result, and in what way it should be applied. For instance, task-based measures cannot be derived directly by means of numerically computing distributions of (finite) Markov chains. Also, the appropriate way of setting up a simulation depends on the domain of the metric [21].
We see from the examples of task-based measures and time-based measures, that this distinction closely relates to the distinction between task measures and systems measures. Indeed, in many cases a task measure (recall that a task measure provides information for the service) will be expressed by means of a task-based function of a random

variable. However, a task measure as the throughput of tasks, does not obey this rule. The throughput, being the number of tasks processed per time unit, is directly related to tasks and thus is a task measure. However, the measure is defined per time unit, and thus time-based. Note, that the inverse of the throughput, being the time elapsed per processed task, is a task-based measure. Another example can be found in the reward measures as defined in [26], which are time-based, but are task measures by the fact that the rewards are of interest to the user.

Not always, the measure is expressed per time unit or per task. For instance, in the case of a communication system in which messages are subdivided in mini-packets, one can be interested in the fraction of corrupted messages. This measure is expressed 'per message', and thus its domain is messages. However, essentially these measures can be discussed as a special case of task-based measures.

Furthermore, we like to note that the original definition of performability with its accomplishment levels [18] naturally leads to time-based measures in the form of reward measures and to the framework of time-based measures in [26].

The second item on which measures can be distinguished is whether steady-state or transient results are desired. **Steady-state** results are results for the long term. For task-based measures it is for an average task in the long run, for time-based measures it is for time goes to infinity. **Transient** result on the other hand consider particular tasks, or results at a particular instant of time.

Both transient and steady-state, as well as task-based and time-based results can be either point measures or interval measures. **Point** measures give results for a single customer, or for a single point in time. Note that this point in time can possibly go to infinity, thus resulting in a steady-state point measure. **Interval** measures provide results for a set of tasks, or an interval of time. Note that also steady-state interval measures can exist. For instance, result for the time interval $[t, t+1]$, are steady-state measures when $t \to \infty$. We note that the motivation for distinguishing between these different measures is that they influence the applicable solution techniques.

Performability evaluation

In terms of the modelling framework we have presented here, we can make the definition of performability evaluation again more specific.

Performability evaluation *is the derivation of a meaningful function of a random variable, for a model which takes into account the timed behaviour of the system and the tasks, as well as the mutual influence between them.*

This definition completes our performability evaluation framework. Meaningful model-based quantitative evaluation can be carried out within this framework.

Relation with other performability frameworks and concepts

We will relate the resulting use of 'performability evaluation' with the use of this terminology in literature. The concepts we have presented here are very closely related to the performability concepts of Meyer [19]. Meyer defines performability conceptual-

ly as a measure, especially tailored for fault-tolerant computer systems. In this paper performability is approached a bit different, and essentially we leave open what measure can be considered to be 'meaningful'. On the other hand, in this paper we have in more detail given an approach to modelling for performability evaluation. In this sense, one can see the modelling view in the PEF as a characterization of the general concept of model-based performability evaluation. Note that the way we have established the performability framework, this framework is in our view more or less a *consequence* of doing meaningful model-based quantitative evaluation of computer and communication systems.

We have treated performability evaluation from the viewpoint of the application area of computer and communication systems, and tried to propose a framework which is generally applicable for the area of interest, and which nevertheless is specific enough to be of practical interest. We have discussed the modelling approach in some detail, but left open what choices of the measures can be appropriate and what solution methods should be considered. With regard to the measures we refer to the performability measure of Meyer [19], reward measures of Sanders *et al.* [26]. See also discussions regarding measures in [27]. Solution methods, such as Markov reward model solutions are discussed in [28], and methods based on differential equations and 'performability-to-go' in [24].

3. The PEF in Relation to Dependability and Quality of Service

In this section we discuss the relation of the performability evaluation framework with known concepts from the area of fault-tolerant computer systems and communication systems. Within fault-tolerant computer systems issues related to the dependability of a system are considered, while within communication systems issues related to QoS receive growing attention. We will first discuss in Section 3.1 dependability issues in fault-tolerant computer systems, then in Section 3.2 Quality of Service issues in communication systems. Finally, we will relate and compare the performability, dependability and QoS concepts in Section 3.3.

3.1. Dependability in Fault-Tolerant Computer Systems

Dependability is a term used in the area of fault-tolerant computer systems, and is defined as follows [17]:

Dependability *is that property of a system which allows reliance to be justifiably placed on the service it delivers.*

In this definition the terms system and service are used as defined in Section 2.2. So, a service is defined as it is perceived by the user, and although it is not explicitly mentioned in this definition of dependability, it also is the user who places the reliance on the system. We see that although dependability is a property of a system, it is the re-

liance a user can have on the service that matters. Achieving dependability is mainly done by adding redundancy to the system, e.g., adding spare components or error correcting schemes (see [17] for a thorough discussion).

The arguments above demonstrate that the notion of dependability very much is motivated by the needs, the expectations and the perception of the user, although increasing dependability is achieved by enhancement of the system. However, the consequences of this user-oriented viewpoint have not always been accepted. Whenever the word dependability is used, it relates to the system, not to the user perceived service (see the use of the term dependability within CCITT [8], or by other authors, e.g., [29]). Also in model-based evaluation, commonly the term dependability model is used to indicate the system model as defined in Section 2.3, i.e., the model of the possible system configuration changes in time [18] [28]. We also have used dependability in this way, but we note that it is not in the spirit of the definition of dependability from [17], especially not when we discuss the evaluation of the service. Most apparent are the restrictions of the system-oriented discussion when we look at the traditional system measures, such as reliability and availability, which are used for quantitatively evaluating the dependability. As discussed in Section 2.3, especially task measures are of importance.

Comments on the use of the dependability terminology

Let us comment further on the use of the terminology associated with the dependability framework of Laprie [17], as some confusion exists. We have seen that the definition of Laprie is user-oriented, as is the meaning Anderson gives to the word dependability in the preface of [1]:

"(...) we may wish to focus our attention (...) for example on whether the system will betray our secrets, or cause its users to be killed, or produce results too late to be of any use; thus issues of *security*, *safety* and *performance* are naturally subsumed under the generic term 'dependability', as indeed are many other desirable properties of a system."

We see here that also performance is considered to be an element of dependability. In this setting dependability just means that the user can trust the system to do what it promises to do. This notion is of great importance, as it gives the basic consideration that leads to meaningful quantitative evaluation of a system. We do not just want to do quantitative analysis for analysis sake, but want to derive useful information for the system, in other words we want to achieve justified reliance.

In *system design and development* dependability is often taken to be a synonym for fault avoidance, fault tolerance or fault removal mechanisms [29]. Indeed, according to [17] this is the way dependability can be achieved, however it is slightly inaccurate to demand these mechanisms to obtain dependability. In some cases a system might be considered dependable without these elements.

In *modelling* we see that the system model, as defined in Section 2.3, is called the dependability part of the model, e.g., [28]. Typically, but not necessarily, the system model encompasses the modelling of fault-tolerance mechanisms.

In *CCITT* dependability is defined as *the collective term used to describe the availability performance and its influencing factors: reliability performance, maintainability performance and maintenance support performance* [8], see also Figure 4. This definition is followed by the note that dependability is used only for general descriptions in non-quantitative terms. The actual quantification is done under the heading of availability performance, reliability performance, maintainability performance and maintenance support performance. We see that within communication systems elements as maintainability and support come into play to decide on the dependability of the system. We will shortly discuss how this differs from the fault-tolerant computer systems point of view as presented in [17].

Within fault-tolerant computer systems there exist three *perceptible attributes* of dependability:

- *Reliability*. Dependable means reliable with respect to continuity of service;
- *Safety*. Dependable means safe with respect to the non-occurrence of catastrophic failures;
- *Security*. Dependable means secure with respect to the avoidance or tolerance of deliberate faults.

Then, the evaluation can be in terms of *reliability*, denoting the probability that the service has continuously been delivered in a proper way for some interval of time, and the *availability*, denoting the fraction of a time interval in which the service has been delivered properly. Note, that reliability has two different meanings; first it is the perceptible attribute denoting continuity of service, secondly it is a measure. This has lead to confusion of what attributes dependability consists of e.g., [28]; according to [17], availability is not a perceptible attribute, only a measure.

We see that within CCITT more elements decide on whether a system is dependable. We therefore introduce the following perceptible attributes of dependability when extended to communication systems:

- *Maintainability*. Dependable means maintainable with respect to the avoidance or removal of faults by means of, possibly preventive, maintenance;
- *Policability*. Dependable means policable with respect to the avoidance or tolerance of possibly non-deliberate faults.
- *Accuracy*. Dependable means accurate with respect to the avoidance of serious failures.

These attributes come across in communication systems, more than in computer systems. Maintainability is a very important property of communication networks, for example for telephony. Policability is important in the new generation of networks, such

as intelligent networks and B-ISDN. In these networks an agreement is made between supplier and end-user which the supplier want to be able to control or *police* against *both* deliberate and non-deliberate faults (compare with the meaning of safety). Maintainability and policability are both requirements posed by the service provider. Note that in Section 2.2 we have introduced the service provider as a type of user, to be able to cope with its requirements too. Accuracy has the same meaning as safety except that catastrophic is too strong a term. An example of an inaccurate service is a wrong connection in a telephone network, which is a more serious error than no connection. We do not claim to have given all possible dependability attributes. For different areas than fault-tolerant computer systems or communication systems, different perceptible dependability attributes might be thought of, compare for instance the industrial demands for dependable software.

We see that dependability is approached slightly differently by different communities. Basically, we use the word dependability in this paper as defined in [17].

3.2. Quality of Service in Communication Systems

Within the area of communication systems Quality of Service aspects are receiving growing attention (e.g., [4] [11] [16]). We will discuss QoS as it is used by CCITT, where it is defined as follows [7]:

Quality of Service *is the collective effect of service performances which determine the degree of satisfaction of a user of the service.*

In this definition we directly see the relation to the user of the service. The service performances may consist of all possible performance criteria CCITT has defined [5] [7] [8]. It comprises so-called *Grade of Service* (GoS) parameters [5] [6], such as blocking probabilities, cell or dial tone delays, etc., but also *network performance* such as the so-called serveability performance, which denotes the system's ability to provide a service when desired, and the ability to provide this service for some requested duration. See [3] for a discussion of these topics.

We note here that depending on the place where the performance is measured, within CCITT terminology serveability can be both user-perceived or 'true' network performance. In our view, see Section 2.3, this user perception is only directly related to the network performance when only proper and improper service can be distinguished (see [2] for a discussion related to these observations). Grade of Service parameters do more directly relate to the user. They comprise traffic parameters, such as delays and blocking probabilities, and are thus typical task measures. As put forward in [6] and [7] too, GoS and serveability performance both contribute to QoS.

QoS terminology

We will elaborate on QoS aspects in some more detail, as it illustrates what is important for the evaluation of a communication system. In Figure 4 the relation between QoS, dependability and measures for dependability are given, as defined within

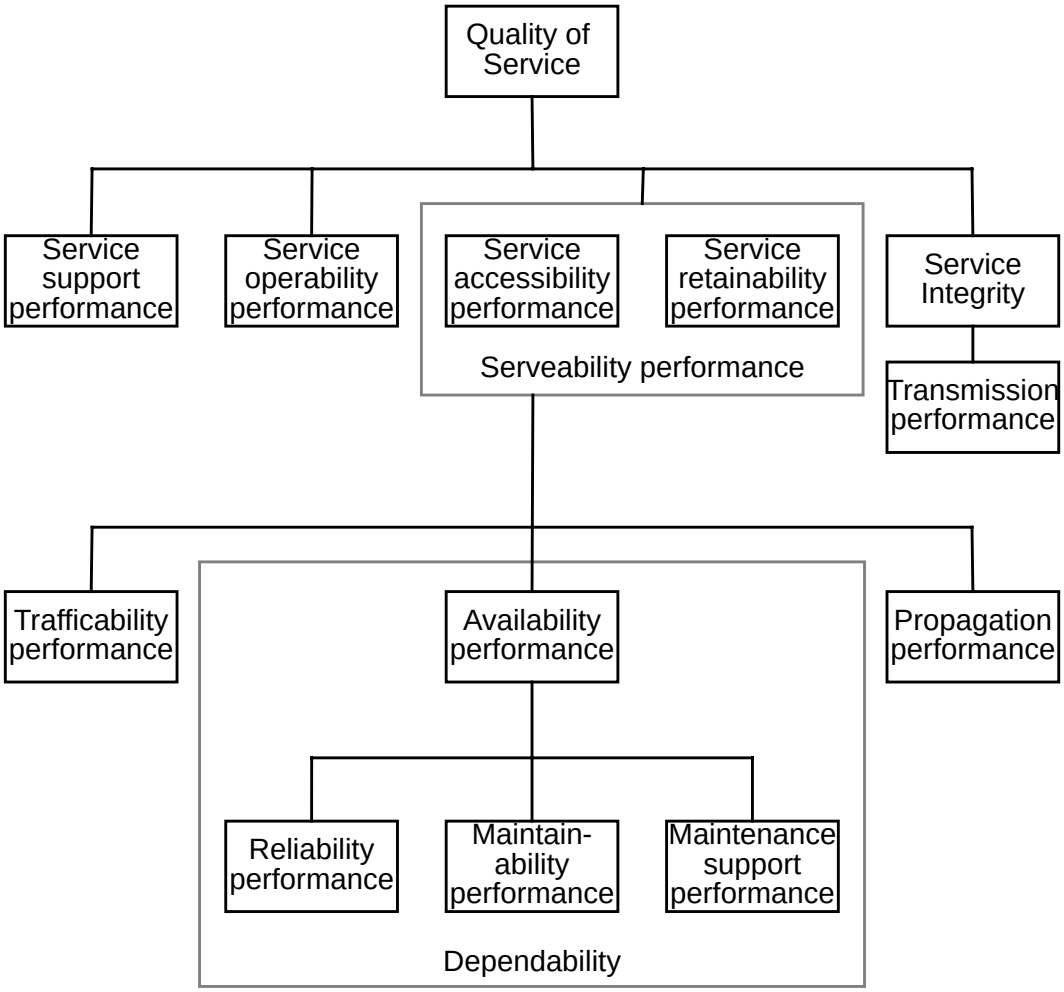

Figure 4. Quality of Service concepts within CCITT [7].

CCITT [7]. We will discuss the main items somewhat informally, more formal definitions of all items can be found in [7].

The *user-perceived* QoS, which forms the top layer of Figure 4, incorporates five different service types, of which we discuss serveability. The lower layer forms the *item-related* performances which deliver the service. We see that dependability in CCITT terminology belongs to these system-oriented performance elements.

The two aspects of **serveability performance** denote the ability of a service to be obtained when requested by the user and continue to be provided for a requested duration. **Service accessibility performance** then is the attribute that deals with the possibility to obtain a service when requested, while **service retainability performance** deals with the possibility to retain a service once obtained. Serveability performance is user perceived, but is delivered by the item-related performances, an **item** being any part of a system that can be individually considered. The large box denotes the **dependability** of the items that deliver the service. **Dependability** is the collective term used to describe the availability performance and its influencing factors: reliability performance, maintainability performance and maintenance support performance

[8]. **Availability performance** denotes the ability of an item to be in a state to perform a required function at a given instant of time, or at any instant of time within a given time interval, assuming that the external resources, if required, are provided. This availability can be partitioned into three aspects. First, **reliability performance**, denoting the ability of an item to perform a required function for a given time interval. Secondly, **maintainability performance**, denoting the ability to restore or retain an item into a state in which it can perform a required function. Thirdly, **maintenance support performance**, denoting the ability of the service supplier to provide upon demand the resources for maintenance.

Within ISO bodies, the meaning of QoS is different, for instance when it is embedded in the OSI reference model. QoS parameters typically are delay of a connection set up, etc. In different fields of communication systems the term Quality of Service has different meanings, even within ISO [10] [14] [15]. We will use the term QoS as defined by CCITT, which has as main distinguishing element the user-oriented view at QoS.

3.3. Relation between PEF, Dependability and QoS

In this section we relate the performability evaluation concepts as defined in Section 2, with the concepts of dependability and Quality of Service as used in fault-tolerant computer systems and communication systems, respectively.

Both dependability and QoS essentially discuss *requirements* that can be put on a service. So, assuming the collecting of requirements is carried out well, the dependability and QoS objectives define what is *meaningful* evaluation. Furthermore, dependability relates to computer systems, and QoS to communication systems, the types of systems which are considered in the PEF. Restricting our attention to requirements on the behaviour *in time* of the system, we can state that performability evaluation evaluates whether, or to what extent, dependability and QoS requirements are met.

We thus see that there is a direct link between posing requirements and evaluating them. The discussion on meaningful measures can be held in a similar manner for meaningful dependability and QoS requirements, and our arguments that interest should be in task measures, more than in system measures, can be repeated for requirements. These considerations can face practical difficulties, however. One main difficulty lies in translating the possibly subjective requirements of the human end-user to quantifiable measures. Furthermore, one might want to be able to evaluate whether indeed the requirements are met, for instance, requirements testing has to be carried out for railway systems. In this case, requirements that are meaningful but cannot be evaluated are useless.

In describing QoS in CCITT many levels of abstraction are used, sometimes model--based, sometimes not. As examples, there are considerations about where to measure the performance, there exist models for calls and there are dependability planning models. It will be of interest to relate these different approaches within CCITT from a performability evaluation point of view, and thus also to relate it with the PEF. In

fault-tolerant computer systems, the frameworks of Laprie [17] and Meyer [18] form the link between system considerations and performability evaluation considerations. In the area of QoS evaluation of communication systems, such a conceptual framework cannot yet be found.

4. Conclusion

We have constructed the performability evaluation framework in this paper. The PEF is constructed such that it should naturally fit the different measures, modelling methods and solution techniques that arise when doing meaningful model-based quantitative evaluation of computer and communication systems. Furthermore, we have discussed dependability issues in computer systems as well as Quality of Service issues in communication systems. We have commented on the existence of perceptible dependability attributes, other than the ones in the dependability framework, which arise especially in communication systems. The PEF should naturally support evaluation approaches which come from the dependability and QoS considerations. In this respect, the need for user-oriented performability evaluation in the form of task measures, has been the focus of our attention. With this in mind, meaningful quantitative evaluation can be carried out.

Acknowledgments

We like to thank Ignas G. Niemegeers and Victor F. Nicola of the University of Twente, as well as the anonymous referee for providing us with insightful comments.

References

[1] T. Anderson (Ed.), *Dependability of Resilient Computers*, Blackwell Scientific Publications Ltd., Oxford, 1989.

[2] C. Asgersen, “Grade of Service as Basis for Network Planning”, *Teletraffic and Datatraffic in a Period of Change,* International Teletraffic Conference 13, A. Jensen, V.B. Iversen (Eds.), pp. 79-86, Elsevier Science Publishers, Amsterdam 1991.

[3] N. Bjorkman, M. Goldstein, L. Hedman, A. Latour-Henner, P. Tholin, L. Gil, “Network Performance (NP) and its relationship with Quality Of Service (QOS) in an Experimental Broadband Network”, *Broadband Communications*, A. Casaca (Ed.), Elsevier Science Publishers, pp. 157-168, 1992.

[4] V.A. Bolotin, J.G. Kappel, P.J. Kuehn (Eds), *IEEE Journal on Selected Areas in Communications*, special issue on Teletraffic Analysis of ATM Systems, Vol. 9, No. 3, April 1991

[5] CCITT, *Blue Book*, Vol. II, Fascicle II.3, Recommendation E.600, Terms and definitions of traffic engineering, International Telecommunication Union, Melbourne, Australia, 1988.

[6] CCITT, *Blue Book*, Vol. II, Fascicle II.3, Recommendation E.720, ISDN Grade of Service concept, International Telecommunication Union, Melbourne, Australia, 1988.

[7] CCITT, *Blue Book*, Vol. II, Fascicle II.3, Recommendation E.800, Quality of Service and dependability vocabulary, International Telecommunication Union, Melbourne, Australia, 1988.

[8] CCITT, *Blue Book*, Vol. II, Fascicle II.3, Supplement 6 to the series E recommendations relating to telephone network management and traffic engineering, International Telecommunication Union, Melbourne, Australia, 1988.

[9] CCITT, *Blue Book*, Vol. III, Fascicle III.8, Recommendation I.350, General aspects of Quality of Service and Network Performance in digital networks, including ISDN, International Telecommunication Union, Geneva, Switzerland, 1989.

[10] C. Chabernaud, S. Goerlinger, "Requirements on IN nodes to meet QoS Objectives", *Intelligent Networks*, P.W. Bayliss (Ed.), pp. 51-62, IOS Press, Washington, 1992.

[11] J.W. Cohen, C.D. Pack (Eds.), *Queueing. Performance and Control in ATM, Proceedings of the Thirteenth International Teletraffic Congress,* ITC-13, Copenhagen, Denmark, June 19-26, 1991.

[12] B.R. Haverkort, I.G. Niemegeers, "On the mutual performance-dependability influence in dynamic queueing networks," *Proceedings of the First International Workshop on Performability Modelling of Computer and Communication Systems*, Twente, The Netherlands, pp. 33-40, 1991.

[13] Y. C. Ho, "Dynamics of Discrete Event Systems", *Proceedings of the IEEE*, Vol. 77, No. 1, pp. 3-6, 1989.

[14] ISO, "A Suggested QoS Architecture for Multimedia Communications," United Kingdom contribution, ISO/IEC JTC1/SC21/WG1, project JTC 1.21.57, September 1992.

[15] ISO, "Reference Model for Open Distributed Processing," ISO/IEC JTC1/SC21/WG7, project JTC 1.21.43, Recommendation X.902: Basic Reference Model of Open Distributed Processing - Part 2: Descriptive Model, concept 11.2.3.

[16] J. Kurose, "Open Issues and Challenges in Providing Quality of Service Guarantees in High-Speed Networks", *Computer communication review*, Vol. 23, No. 1, pp. 6-15, January 1993.

[17] J.C. Laprie (Ed.), Dependability: Basic Concepts and Terminology, Springer Verlag, Wien, 1992.

[18] J.F. Meyer, "On Evaluating the Performability of Degradable Computing Systems", *IEEE Transactions on Computers,* 29, pp. 720-731, 1980.

[19] J.F. Meyer, "Performability evaluation of telecommunication networks," *Teletraffic Science*, M. Bonatti (Ed.), North-Holland, Amsterdam, pp. 1163-1172, 1989.

[20] J.F. Meyer, "Performability: A Retrospective and Some Pointers to the Future", *Performance Evaluation* 14, pp. 139-156, 1992.

[21] I. Mitrani, *Simulation Techniques for Discrete Event Systems*, Cambridge University Press, 1982.

[22] A.P.A. van Moorsel, B.R. Haverkort, I.G. Niemegeers, "Fault Injection Simulation: A variance reduction technique for systems with rare events," *Dependable Computing for Critical Applications, Vol. 6 of Dependable Computing and Fault-Tolerant Systems*, J.F. Meyer, R.D. Schlichting (Eds.), pp. 115-134, Springer Verlag, Wien, 1992.

[23] A.P.A. van Moorsel, B.R. Haverkort, I.G. Niemegeers, "A Method for Analyzing the Performance Aspects of the Fault Tolerance Mechanisms in FDDI," *Proceedings of IEEE INFOCOM'92*, Firenze, Italy, pp. 372-381, May 1992.

[24] K.R. Pattipatti, Y. Li, H.A.P. Blom, "A Unified Framework for the Performability Evaluation of Fault-Tolerant Computer Systems," *IEEE Transactions on Computers*, Vol. 42, No. 3, March 1993, pp. 312-326.

[25] QOSMIC, "QoS and NP, relationship between related terms," *QOSMIC Consortium*, QOSMIC / STG 1.4 5/C,contribution to ETSI, 1991.

[26] W.H. Sanders, J.F. Meyer, "A Unified Approach for Specifying Measures of Performance, Dependability, and Performability", *Dependable Computing for Critical Applications, Vol. 4 of Dependable Computing and Fault-Tolerant Systems*, A. Avizienis, J.C. Laprie (Eds.), Springer Verlag, Wien, 1992.

[27] E. de Souza e Silva, H.R. Gail, "Performability Analysis of Computer Systems: From Model Specification to Solution", *Performance Evaluation* 14, pp. 157-196, 1992.

[28] K.S. Trivedi, J.K. Muppala, S.P. Woolet, B.R. Haverkort, "Composite Performance and Dependability Analysis", *Performance Evaluation* 14, pp. 197-215, 1992.

[29] C.B. Weinstock, W.L. Heimerdinger (moderators), "Panel: The State of the Practice in Fault Tolerant Systems", *Proceedings of FTCS-22, Symposium on Fault-Tolerant Computing*, IEEE Computer Society Press, pp. 2-5, 1992.