A Comparative Analysis of Network Dependability, Fault-tolerance, Reliability, Security, and Survivability

M. Al-Kuwaiti, *Member, IEEE,* N. Kyriakopoulos, *Senior Member, IEEE*, and S. Hussein, *Member, IEEE*

***Abstract*—A number of qualitative and quantitative terms are used to describe the performance of what has come to be known as information systems, networks or infrastructures. However, some of these terms either have overlapping meanings or contain ambiguities in their definitions presenting problems to those who attempt a rigorous evaluation of the performance of such systems. The phenomenon arises because the wide range of disciplines covered by the term information technology have developed their own distinct terminologies. This paper presents a systematic approach for determining common and complementary characteristics of five widely-used concepts, dependability, fault-tolerance, reliability, security, and survivability. The approach consists of comparing definitions, attributes, and evaluation measures for each of the five concepts and developing corresponding relations. Removing redundancies and clarifying ambiguities will help the mapping of broad user-specified requirements into objective performance parameters for analyzing and designing information infrastructures.**

***Index Terms*—Dependability, fault-tolerance, reliability, security, survivability.**

I. Introduction

THE DISRUPTIONS of the operation of various major infrastructures have highlighted the need to develop mechanisms for minimizing the effects of disruptions and improving the performance of each infrastructure. The problems start with the composition of infrastructures. They comprise systems that have been developed via different disciplines. The hardware component of the information infrastructure includes devices from all fields of electrical engineering, as the software component includes development from all disciplines in computer science to mention just two examples.

The integration of the products of diverse fields, including the human component, into complex systems has created major difficulties in the development of efficient mechanisms for analyzing and improving the performance of infrastructures. One of the problems can be traced to variety of terminologies for describing performance across different fields. A designer or user is faced with terms that may be complementary, synonymous or somewhere in between. Thus, there is a need to develop a common understanding of the meaning of the most widely used terms without reference to a specific discipline.

Manuscript received 30 September 2006; revised 23 April 2008.

The authors are with the Department of Electrical and Computer Engineering, The George Washington university, Washington, DC 20052 USA (emails: kuwaiti@gwu.edu, kyriak@gwu.edu, and drsay@gwu.edu).

Digital Object Identifier 10.1109/SURV.2009.090208.

In the evolution of engineering design approaches, the initial concern was whether a particular system would operate within a set of specifications; little consideration was given to how long the system would operate within those specifications. Military and subsequently space applications placed great emphasis on *reliability*, namely, the need for a given system to meet design requirements for specified periods of time. Similar needs arose as simple computer programs evolved into complex software leading to the requirement for *fault-tolerant* design. It did not take long to realize that although these two concepts originated in separate fields, namely, engineering and computer science, respectively, their end objectives were similar, that is to ensure the performance of a system, hardware in the first case, software in the second, for some specified time intervals. This realization led to the development of the concept of *dependability* as an all-encompassing concept subsuming reliability and fault-tolerance [1]-[6]. As computers combined with communications formed global information networks, information *security* and network *survivability* have also been introduced as significant design objectives. In addition, many other concepts such as *trustworthiness* [2], *high assurance* or *high confidence* [3], *ultra-reliability* [5] and *robustness* [2] are also being used to characterize the performance of complex systems. In [3] the observation is made, without any detailed analysis, that dependability, high confidence, survivability, and trustworthiness are all "*essentially equivalent in their goals and address similar threats*".

These terms have entered into use via different disciplines and at different times. For example, robustness has a long history of use in statistics and process control, survivability was introduced as a performance characteristic for military communications networks, and security has long been associated with law enforcement and military operations. The last two examples illustrate the specification of performance measures with discrete physical entities, while the first one relates performance to measurements. As information networks have become more complex, involving hardware, software and humans, and have assumed a prominent role, it became inevitable that performance requirements for the services provided by the entire system needed to be established.

The concepts for measuring performance evolved along with technology. For example, the reliability of electronic devices led to the reliability of circuits and, eventually, to the reliability of the entire launch operation, or, in another application, an entire nuclear power plant. Other concepts

have been "borrowed" from one technology and used in another. Survivability started as a performance characteristic of physical communications networks and has been adopted to characterize performance of information infrastructures [7]-[12].

Another concept that has evolved with technology is the *quality of service* (QoS) which is defined in the ITU-T Rec. X.902 as "a set of quality requirements on the collective behavior of one or more objects" [13]. At the dawn of digital communications, channel performance was expressed in terms of *bit error rate* to be followed by *quality of service* in describing the performance of packet-switched networks [14][15][16]. With the advent of the Internet and data streaming, the quality of service concept is also used to characterize the performance of an application [17]-[21].

The different origins and development paths of these concepts in combination with lexicological similarities raise the question whether there is some degree of overlap among the various terms, and to what extend, if any, there are redundancies in using these concepts simultaneously.

This question assumes added significance, because of the effort to develop dependability as an integrating concept that encompasses such attributes as availability, reliability, safety, integrity, and maintainability [3]. Although, theoretically, this is a highly desirable goal, some practical problems arise primarily due to multiplicity of perspectives and definitions. For example, there are more than one definitions of dependability [2][3][4][22][23]. Concepts that have been in use for a long time and in different disciplines are well entrenched in their respective fields and are viewed as end objectives rather than attributes of some other concept. One such widely used concept is reliability. Applications range from statistical data analysis to description of performance of social systems. If one uses the IEEE definition for reliability "ability of a system or component to perform its required functions under stated conditions for a specified period of time" [24], the argument could be made that it is as broad a concept as dependability. The term "required functions" is broad enough to cover, for example, safety and security which are defined as attributes of dependability.

While reliability can be specified quantitatively, safety and security are examples of qualitative attributes for which there is no objective evaluation mechanism. If these attributes are to be used as design parameters, they need to be quantified. There is no doubt that the old adage saying of Tom DeMarco: "*You can't control what you can't measure"* is relevant in describing these concepts and setting the basic performance requirements [25].

Another problem stems from differences in design perspectives. Historically, the focus of engineering design has been the device, component and system in that order. The integration of hardware and software to offer services has led to the need for specifying performance characteristics for such services. One could consider fault-tolerance in computing as the origin of the service perspective in specifying performance and a precursor to the broad concept of dependability. On the other hand, it could be argued that quality of service (QoS) is an equally valid integrating concept for specifying performance characteristics for services. The set of "quality requirements" included in the definition of quality of service could be the same attributes as those of dependability. Furthermore, in contrast to the evolution of engineering design from the device to the system, namely, a "bottom up" approach, dependability has a "top down" perspective. The result is an elaborate multi layer tree structure encompassing both quantitative and qualitative attributes. As one goes beyond a few levels in the tree structure, the multiplicity of possible paths from the root of the tree and the mixture of quantitative and qualitative attributes present major challenges in translating the top level dependability concepts into implementable design specifications for a complex system.

The work described in this paper aims toward the development of a framework for identifying a set of performance indicators for complex systems such as an information infrastructure. The objective is not to propose yet another concept, but, rather, to identify from the existing concepts the proper subset of their intersection and to develop a common and consistent understanding of the meaning of them. Dependability, fault-tolerance, reliability, security and survivability are used as representative examples for describing the proposed analytical framework.

The rapid development and wide applications of Information Technology (IT) in every aspect of human's life have made these performance indicators important challenges to system users and designers. Various critical infrastructures aim towards possessing such concepts either by embedding them in the first development design stages or as add-on features. Examples of these systems include defense systems, flight systems, communication systems, financial systems, energy systems, and transportation systems as presented in [8][9][10][12][22][23].

The paper is organized as follows: In section II the five concepts are discussed as they have been evolving. The concepts have been analyzed from three distinct perspectives, component-level, infrastructure and service. Described are also the approaches followed in evaluating the five concepts and their definitions and characteristics. This section concludes with a description of the research framework for integrating service and system performance requirements. Section III examines the concepts definitions, requirements, and attributes taxonomies. Section IV analyzes the definitions of the concepts and investigates the similarities and differences between them. Section V examines their evaluation measures. A side-by-side comparison of the concepts is done in section VI, followed by a discussion of some major observations. Finally, section VII concludes with a summary and possible avenues for future work.

II. Historical Development and Research Framework

A. Historical Development

New disciplines formed as a result of scientific and technological innovations which have developed their own terminology, partly by borrowing from existing disciplines, but also by inventing new ones. As each discipline evolves along its own path, even terms that share the same origin assume nuances influenced by the evolution of each discipline. To get a better

understanding of the nuances inherent in the five concepts, it is necessary to follow the paths they followed as they evolved. To establish a cross-discipline reference point, one can identify three distinct system view perspectives, namely, (i) component-level, (ii) infrastructure, and (iii) service.

1) The Component-Level Perspective: The design of engineering systems has progressed from the component level to very large complex systems. Regardless of complexity, system design involves planning that includes, *inter alia*, design goals and specifications, trade-offs, balances between architectures and functions, requirements and functional analysis, flow analysis, physical design, and integration along with testing. The defining characteristic of the component-level perspective has been the control exercised by the designer on the structure and operation of the system regardless of size. This reference point can be seen in transistor circuits, software packages, mainframe computers, communication networks and other bounded systems. Computer systems were mainframes or midrange systems that were bounded as well as centrally controlled and administered. This evolved from having mainly few components or devices and then progressed to a complete system over time.

System design is based on the use of quantifiable variables that embody the component's performance requirements. Thus, the concepts evolved and used in describing such systems are mainly quantifiable concepts, such as reliability and fault-tolerance. Historically, *reliability* has been one of the essential as well as the oldest concept used in designing systems. It initially arose as a consequence to the need to ensure that components performed their functions within specified performance requirements that included time. The components reliability then progressed to the overall system reliability [5]-[26]. The advent of the space program may be viewed as the defining moment when device reliability evolved into complete system reliability by linking component failures to system failure. In 1952, von Neumann, in his work on probabilistic logic and the design of reliable mechanisms from unreliable components laid out the fundamental principles of *fault-tolerance* [27]. Subsequently, in 1967 Avizienis integrated these techniques for detection, diagnosis, and recovery into the concept of fault-tolerant systems [28]. As computer systems grew in complexity and utilized in various critical applications, fault-tolerance became a fundamental design concept and a distinct field of study, although its origin is the concept of reliability.

Another concept that has its origins in the component level perspective is survivability. One of the original uses of the concept was in the design of military communications networks that would be able to operate reliably even if some nodes or links were destroyed [8][11][29][30]. The concept of survivability followed the evolution of communications networks into computer networks and information infrastructures and became applicable to large networked systems [12][31][32][33]. Nevertheless, the original underlining principles that applied to communications networks have remained unchanged.

Although both fault-tolerance and survivability trace their origins to the concept of reliability, they have emerged as distinct disciplines. Nevertheless, their shared origin in reliability gives them some degree of overlap. In other works there are neither disjoint nor hierarchical.

The component-level perspective may be viewed as associated primarily with the lower levels of the OSI Reference Model, *i.e.*, the network-dependent layers. On the other hand, dependability and security have been developed following a top-down approach [1][2][23][34][35][36]. In this respect, one may view dependability and security as associated with the upper layers of the OSI Reference Model. In the top-down approach, dependability and, to a lesser extend, survivability, are viewed as unifying concepts that subsume reliability and fault-tolerance [3][5][33][37][38].

2) The Infrastructure Perspective : The merging of computers and communications to form computer communication networks has evolved into the concept of information systems followed by the broader concept of *information infrastructure*. While information systems generally imply bounded networks with topology, location and components specified, information infrastructure comprises a collection of information systems which may be connected or disjoint. Unlike the formally planned systems that are built from the component-level perspective, information infrastructures, such as the Internet, grow *ad hoc* and unplanned. As information infrastructures are revolutionizing the conduct of human interactions, they have also given rise to a host of new issues ranging from reliability of services to protection of privacy from threats posed by governments and non-governmental actors. These threats are inherent in the open nature of the information infrastructures making their best advantage of these systems also their worst disadvantage [39].

The unplanned development of information infrastructures has not allowed the systematic application of the engineering system design methodologies. Part of the reason is the size and complexity of the infrastructures. As systems became more complex and unbounded, the number of variables increased substantially and the use of quantitative models that have been developed from the component-level perspective became impractical. As a result, new top-down concepts have emerged and previous bottom-up ones revised in an effort to describe and analyze the performance characteristics of complex systems from the perspective of the information infrastructure.

Dependability is the most comprehensive concept for modeling complex systems taking a top-down approach. It is evolving into a distinct discipline attempting to subsume the preceding concepts of reliability, and fault-tolerance. Although there is no universally accepted definition of dependability, the term has been accepted for use in a generic sense as an umbrella concept [3][37]. The dividing line can be traced to the perspectives from which the term is defined. The component level perspective leads to definitions contained in the ISO and CCITT standards while the development of dependability as a unifying concept follows a top-down approach [2][3][23]. The latter evolved from reliability, availability and fault-tolerance considerations [38][40][41].

In promoting dependability as a unifying concept, Avizienis *et al*. [2] have developed a comprehensive set of definitions and taxonomy. In such framework, the concepts of reliability, availability and fault-tolerance have been assigned the role of attributes, although viewed from the component level

perspective they have the characteristics of complete system performance measures.

The information infrastructure perspective has also given rise to the concept of *security*. Although it is a much older concept, its application to the information infrastructure is relatively new and closely associated with networked systems. The openness of the information infrastructure and the absence of complete system design, together with the vast quantities of stored and transmitted information have invited malicious intrusions and attempted disruptions. Thus, security has become an important consideration in the operation of the information infrastructure.

In contrast to the development of the dependability concept, security has evolved somewhat *ad hoc*, particularly as a result of lessons learnt from malicious attacks. This gave rise to the collection of theoretical and practical models, techniques and tools in both hardware and software [34][35][36][39][42]-[47]. These include qualitative techniques codified as management practices, physical protection of hardware and data protection through software. One of the pioneering groups in developing secure information systems has been the Computer Security Division of the National Institute of Standards and Technology (NIST) [43].

Although dependability is a unifying concept, security has the characteristics of a complement to dependability. In addition to sharing a number of common attributes such as availability, confidentiality, and integrity, security also has unique attributes such as accessibility, accountability, authenticity, and non-repudiation [3][40][48]. Similarly, there are links between security and fault-tolerance [49].

Following the top-down development of dependability, the concept of *survivability* has been introduced as a framework for developing requirements and strategies [31][32]. In addition, a rigorous definition of survivability has been defined as the ability of a system to comply with a six-tuple of survivability specifications [33]. In addressing the design of large scale systems the need for assessment techniques during the various design steps has led to the use of techniques developed for reliability and fault-tolerance [9][10][12][30][32]. In effect, the approach to survivability from the information infrastructure perspective has been merging with that of the component-level perspective. Although the concepts of dependability and survivability have progressed independently, they have been found to share some common characteristics [2][12]. Similarly, there is a close connection between security and survivability viewed from the component-level perspective [9].

From this brief review it is apparent that dependability, fault-tolerance, reliability, security, and survivability are related as a result of their evolution and have a number of overlapping characteristics. In this respect, they may be viewed more as parallel concepts rather than some being subordinate to the others. They are neither disjoint nor identical, but somewhere in between. What needs to be determined is the proper set of characteristics for all five concepts.

3) The Service Perspective : Although the component-level and infrastructure perspectives take different approaches, they both address issues of system performance through analysis and design. However, the infrastructure perspective has another dimension, namely, an information infrastructure as a provider of services to the users of that infrastructure. One can easily identify many such services, *e.g.* remote database searches, travel reservations, *etc*. Although users expect some reasonable level of performance for such services, the requirements could be classified as being primarily qualitative. With the advent of music downloads and streaming video, the service requirements have assumed greater importance, because the quality of service provided by the information infrastructure affects directly the quality of the sound and images. Thus, the infrastructure perspective gives rise to two sets of requirements, one for the system and another for the services.

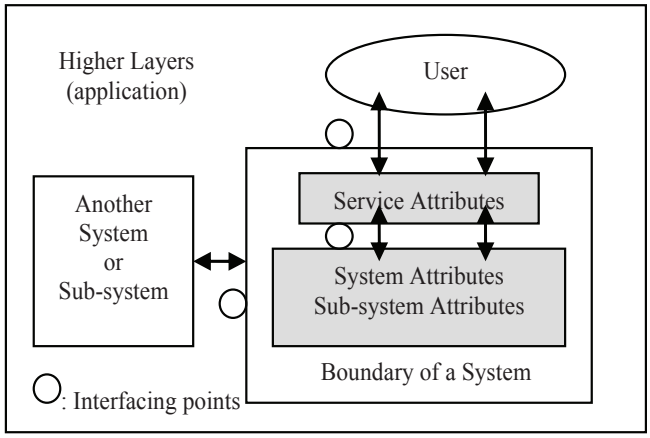

Fig. 1. Service and system attributes.

Proper system design procedures dictate that the service performance requirements must be mapped into an appropriate set of system requirements. Fig. 1 indicates the relationships among user, service and system and identifies the boundaries between these components. In order to develop proper relations between the service and the system requirements, there needs to be a clear understanding of the relations between the system performance parameters embodied in the five concepts of dependability, fault-tolerance, reliability, security, and survivability and service performance parameters.

B. An Integrated Framework

In order to find the required relations between quality of service and the parameters specifying system performance, two problems need to be solved. First, quantitative attributes need to be developed for specifying quality of service expected by the user. Second, a set of linearly independent system performance parameters needs to be identified. Since the five concepts examined in this paper appear to be overlapping to various degrees, the first step toward the development of a set of linearly independent system performance parameters is to clarify the meaning of these concepts and identify the proper union of performance parameters embodied in them.

Design methodologies need to be developed in a way that they take into consideration the advantages offered by the three perspectives. For instance:

- One advantage of the formally planned component-level design approach is derived from the ability to optimize the design using measurable factors (*e.g.*, reliability concept).

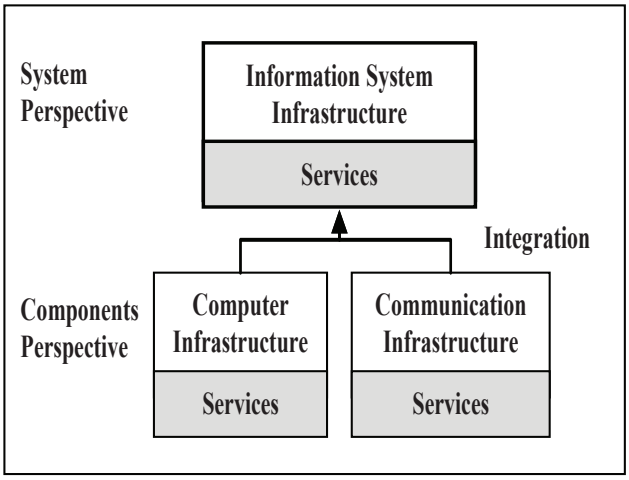

Fig. 2. System and component design perspectives.

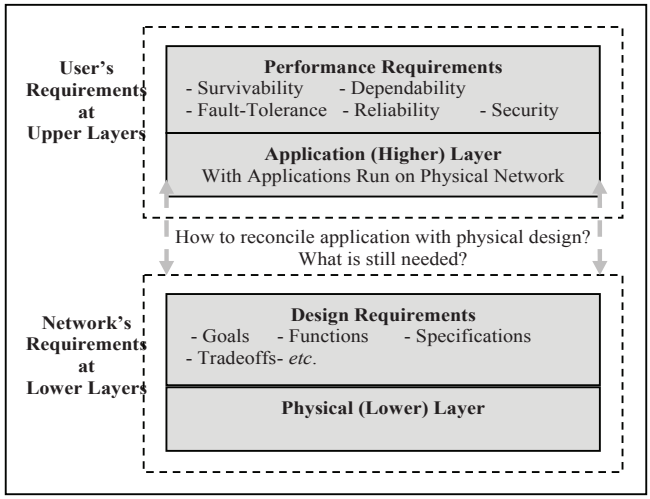

Fig. 3. Upper and lower layers design viewpoints.

- On the other hand, an advantage of the unplanned information system infrastructure is that it allows for almost instantaneous expansion of the infrastructure and the services provided by them.

Fig. 2 illustrates a system view of these design perspectives and the major constituent components of the term "information system infrastructure".

To integrate the service perspective with the component-level and infrastructure perspectives, it is necessary to analyze the user requirements specified at the application level and, ultimately, map them into those of the physical layer. The delineation between services to the users and the physical systems that provide those services is illustrated in Fig. 3 following the hierarchical structure of the OSI Reference Model.

1) Evaluation Approaches : The approaches used to analyze and evaluate the performance of systems are closely influenced by the perspective from which a system is viewed. The component level perspective leads to the use of *quantitative techniques* for analysis and evaluation. It is a natural extension of the use of quantitative models used to describe small-scale components. The quantitative analysis methods have the advantage of producing results that can be utilized to control or compare various performance characteristics of systems or services [42]. For example, in case of the IP-based networks, the application or network QoS parameters provide an assessment to the user of the performance of a specific service. Examples of the quantitative methods are: analytical, experimental, and simulation models, deterministic and probabilistic analysis, algorithms and graphic analysis, weight or impact analysis, and combinations of these techniques [25][40][50].

As the components evolved into infrastructures, the quantitative models required a large number of variables to adequately describe their performance, making the use of such models unwieldy. While the development of quantitative performance measures is necessary for designing today's information infrastructure to allow for the mapping of the service performance requirements into the appropriate system requirements, there are some challenges for achieving such measures. Examples of these challenges imposed by the complexity of implementing these systems include the dynamic changes of their behaviors and the increased number of disruptions [32][51][52].

Another source of difficulties is the evolutionary nature of the information infrastructure. As communications and computer technologies evolve, "legacy" systems have different and usually inferior performance characteristics compared to those of newer systems [8][50][53]. In addition to the problems inherent in the integration of technologies with different performance characteristics, problems also arise due to the changing environments for which the various stages of the infrastructure were built. For example, although the threat of cyber attacks did not exist in an analog communication system, the threat of electromagnetic interference did and it continues for the current systems.

Additionally, some of the concepts used at the component-level approach may no longer be sufficient to address requirements imposed on critical information infrastructures that might have additional desirable features such as *"Open" and unbounded ability* [22][52], *mobility and flexibility* [9], *integration ability and universality* [9], *and load sharing or graceful degradation ability* [5][33][54].

The infrastructure perspective leads to a holistic view of a system as exemplified by the development of the dependability concept. The holistic view has been no more successful in generating quantitative models than the component-level perspective. The dimensions of the state space of the infrastructures are so large as to make the development of quantitative models impractical. As a result, the infrastructure perspective leads to mostly *qualitative descriptions and analysis techniques*, such as professional experience, policies and standards of due cares, brainstorming and documentations, graphic analysis and label assigning, questionnaires, surveys, checklists, and interviews [39][42].

The difficulty of quantifying performance requirements developed from the infrastructure level perspective presents a challenge to use quantitative techniques when dealing with top-level requirements such as dependability, security, and survivability. Part of the reason is that these general concepts have various performance attributes that are hard to assess.

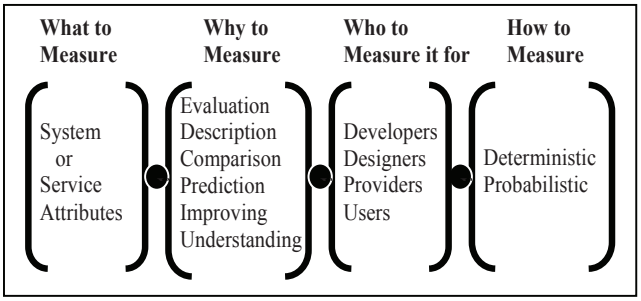

Fig. 4. Characterization of network metrics.

Nevertheless, objective analysis and evaluation requires the use of measurable metrics. At the same time, qualitative analysis techniques such as those mentioned previously also contribute to improvements in the operation of infrastructures. Consequently, *both approaches* are very important and needed in combination to provide a better understanding, designing, and controlling of any system [42].

A related characterization attempt for finding information security metrics was shown by Vaughn *et al*. in [55] where they depict a cross product involving what needs to be measured, why it needs to be measured, for whom it is measured, and how it is measured. A generalization of that is used in our quest for the concepts evaluation metrics as summarized in Fig. 4.

2) A Cross-Cutting Approach: The five concepts examined in this paper follow a hierarchical approach in their application to system performance evaluation regardless how they have been developed. The component-level approach attempts to develop performance criteria from the perspective of the bottom-to-top structure; conversely, the infrastructure approach is based on the top-to-bottom structure. That process plus the fact that the five concepts have their origins in different disciplines and have appeared at different times has created some ambiguities as to the meaning and use of these concepts. In turn these ambiguities make it difficult to analyze the performance of large infrastructures using an optimal set of performance measures.

As a first step toward that goal, the precise meanings of these five concepts relative to each other need to be clarified. In effect, instead of viewing these concepts from the perspective of a hierarchical structure, the approach taken in this paper is to view them in parallel and examine whether they are independent (disjoint), subordinate, or partially overlapping. The five concepts are examined in terms of their definitions, attributes, similarities, and differences. Also performance indicators for evaluating them are identified. The approach aims to contribute toward the development of techniques for integrating the lower layers objective characteristics with the upper layers subjective characteristics to optimize the design while using the user's needs as performance requirements. Fig. 5 below shows the research framework and the next sections discuss it in more details.

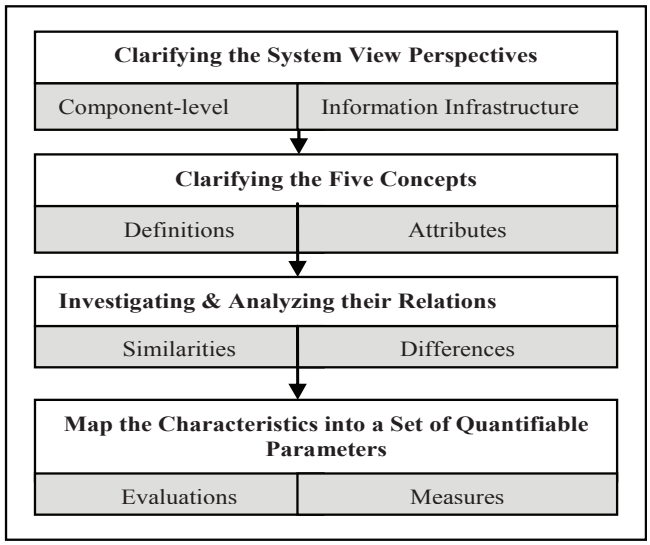

Fig. 5. The Framework of the research.

III. Definitions, Requirements and Attributes Taxonomies

This section reviews (in alphabetical order) the definitions of the five concepts and the structural taxonomy view for each concept using the respective attributes. For clarity, the terms used in the specification of the attributes are also defined. The majority of the terms are used either according to their dictionary definitions or according to definitions that can be found in computer networking glossaries such as the National Institute of Standards and Technologies (NIST) [43] or IEEE Standard Computer Dictionary (IEEE Std. 610) [24].

- *Accessibility:* Ability to limit, control, and determine the level of access that entities have to a system and how much information they can receive. [35]
- *Accountability*: The ability to track or audit what an individual or entity is doing on a system. [35]
- *Authenticity:* The property of being able to verify the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system. [43]
- *Confidentiality:* Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. [43]
- *Fault Avoidance (Prevention)*: A technique used in an attempt to prevent the occurrence of faults. [5]
- *Fault Containment (Isolation)*: The process of isolating a fault and preventing its effect from propagating. [5]
- *Fault Detection*: The process of recognizing that a fault has occurred. [5]
- *Fault Forecasting (Prediction)*: The means used to estimate the present number, the future incidence, and the likely consequence of faults. [3]
- *Fault Location*: The process of determining where a fault has occurred so a recovery can be used. [5]
- *Fault Masking*: The process of preventing faults in a system from introducing errors into the informational structure of that system. [5]

- *Fault Removal*: The means used to reduce the number and severity of faults. [3]
- *Fault Restoration (Recovery)*: The process of remaining operation or gaining operational status via reconfiguration event in the presence of faults. [5]
- *Graceful Degradation*: The ability of a system to automatically decrease its level of performance to compensate for hardware or software faults. [5]
- *Integrity*: Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity. [43]
- *Maintainability*: The ease with which a system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment. [24]
- *Non-Repudiation (Non-Repudiability)*: Assurance that the sender of information is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later deny having processed the information. [43]
- *Performability*: The degree to which a system or component accomplishes its designated functions within given constraints, such as speed, accuracy, or memory usage. [24] It is also defined as a measure of the likelihood that some subset of the functions is performed correctly. [5][56]
- *Safety*: The property that a system does not fail in a manner that causes catastrophic damage during a specified period of time. [38]
- *Testability*: The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met. [24]

These terms help bring attention to another difficulty in the effort to develop quantifiable system performance characteristics. It is best illustrated by the property of non-repudiation. Does it belong to the domain of the system or that of the users? It can be argued that a mechanism for non-repudiation among a set of users is a property of the set and not of the system. A similar argument can be made for the property of authenticity. It pertains to transactions among users and it is a mechanism for satisfying requirements on the set of users; the system merely executes the transactions. The argument whether such properties are system or user properties would be clarified with a clear definition of the interface between system and users.

A. Dependability Definition, Requirements, and Taxonomy

There is no unique definition of dependability. By one definition, it is the ability of a system to deliver the required specific services that can "justifiably be trusted" [23]. It is also defined as the ability of a system to avoid failures that are more frequent or more severe than is acceptable to the users [2]. By another definition in [4], dependability is a system property that prevents a system from failing in an unexpected or catastrophic way. Although these definitions are all similar, we will adopt the first one for our framework.

Dependability evolved from reliability and availability considerations. To remove potential confusion between the dependability and reliability concepts, dependability is used in a generic sense as an umbrella concept, whereas the reliability concept is used as a precisely defined mathematical function [3][37].

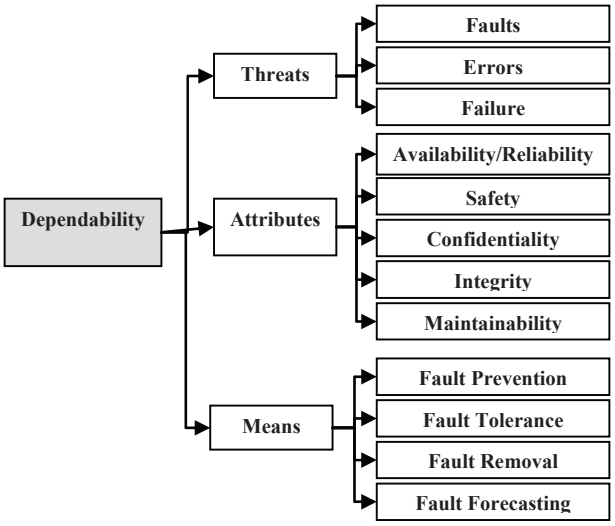

Fig. 6. Dependability Concept Taxonomy [2].

Dependability measures the degree to which a system is operable at any random time during a specific mission profile, given that its services are available at the start of the mission.

Some of the dependability requirements that need to be considered are: having no single point of failure, anticipating all faults and making system's design deal with all anticipated faults by reducing its effect to an acceptance level, and implementing fault handling methods [2][4]. The requirement of anticipating all faults is considered a big challenge in this discipline. In addition to that, dependability of a system can be achieved as described by Avizienis *et al*. by the joint and balanced deployment and operation of a set of four techniques which are: fault forecasting (or prediction), fault prevention, fault removal, and fault tolerance [2]. More details about these requirements are described in the following references [1][2][3][38].

Dependability is not a single property measure, but a collection of related measures including some attributes such as reliability, availability, and safety [1][2][4]. Different authors describe dependability of a system as a set of properties or attributes. For instance, in [1][2][3][22][57], dependability concept includes some attributes such as reliability, maintainability, safety, availability, confidentiality, and integrity where the last three are shared with the security concept. Some of these attributes are quantitative (*e.g.*, reliability and availability) while some are qualitative (*e.g.*, safety). A generalized view of dependability attributes along with its threats and the means to achieve dependability are shown in Fig. 6.

B. Fault-Tolerance Definition, Requirements, and Taxonomy

Fault-tolerance is the ability of a system or component to continue normal operation despite the presence of hardware or software faults. A fault-tolerant system is a system that has the capability to continue the correct execution of its programs and input/output functions in the presence of faults [24][27].

In a similar fashion, it aims at maintaining the delivery of correct services in the existence of active faults [1][2]. As illustrated in the dependability definition above and the reliability definition below, fault tolerance is considered as one of the important means that are used to achieve dependability and reliability [2][5][54].

There are three fundamental terms in fault-tolerant design which are fault, error, and failure, and a cause-effect relationship that exists between them [5]. Fault-tolerant design requires that faults be considered throughout the requirements definition and system design process. Since failures are directly caused by errors, the concept "fault-tolerance" and "error-tolerance" are frequently used interchangeably. Further, fault-tolerance can support performability and graceful degradation features which provide the ability to slowly and gradually eliminate the effect of hardware and software faults from a system, therefore, permitting the system to function at some reduced level.

There are various techniques for achieving fault tolerance [2][3][5][25][58]. In general, it is realized by error detection mechanisms followed by the appropriate system recovery. Fault masking is another technique to tolerate faults. Other techniques include detecting, locating, diagnosing, and confining the faults as well as reconfiguring the system to remove the faulty component. Reconfiguration is the "process of eliminating a faulty entity from a system and restoring the system to some operational condition or state" [5]. When using the reconfiguration process, then the designer must be concerned with fault detection, fault location, fault containment, and fault recovery [5].

One way to satisfy the requirement on a fault-tolerant system to operate correctly even after some failures of its elements is to introduce redundancy for some of its elements. However, it is possible for the system to attain an acceptable level of performance even without such elements.

Fault-tolerance is a property that is designed into a system to achieve some design goals. Although various authors discuss different attributes of fault-tolerance, some of the most common ones are availability, performability, maintainability, and testability [5][58][59]. Further, graceful degradation is also an important feature that is closely related to performability. The concept taxonomy for fault-tolerance is shown in Fig. 7.

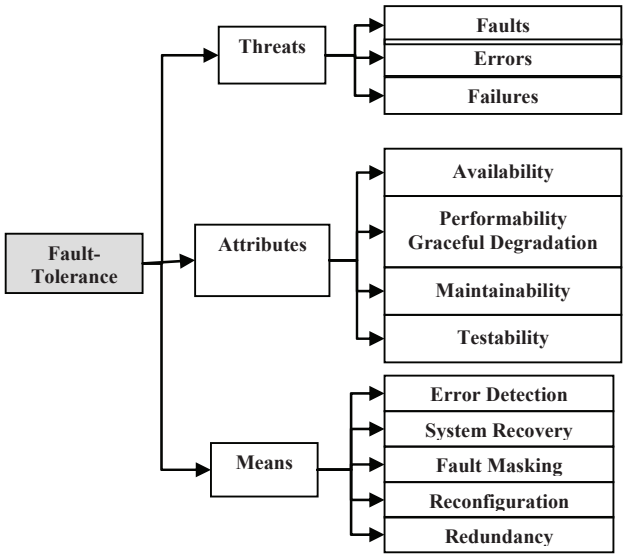

Fig. 7. Fault-Tolerance Concept Taxonomy.

C. Reliability Definition, Requirements, and Taxonomy

Unlike dependability and fault-tolerance, reliability can be used as a well-defined mathematical function, specifically, as "the ability of a system or component to perform its required functions under stated conditions for a specific period of time" [24]. A more precise definition is given as a conditional probability that the system will perform its intended function without failure at time t provided it was fully operational at time $t=0$ [5]. Still another defines reliability in a similar fashion as "a measure of the continuity of correct service" [1][3].

A related concept that needs to be addressed here is the *availability* concept. It is closely related to reliability, and is also defined as "the ability of a system 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" [24]. To clarify the difference between the reliability and availability concepts, it is important to realize that reliability refers to failure-free operation during an interval, while availability refers to failure-free operation at a given instant of time, usually the time when a device or system is first accessed to provide a required function or service [5][37]. One can make the case that availability is reliability evaluated at an instant. In analyzing these concepts, availability is always used as a priori attribute.

In order to obtain a reliable design, reliability must be implemented into each component of the system; that is, considering it from the very beginning of the system design [54]. Similar to dependability requirements, there are four important features that must be introduced and implemented at the starting phase of the design to achieve reliable and available systems, namely: fault-avoidance, fault-tolerant, fault-detection/isolation, and fault-restoration [26][54].

The attributes of maintainability and testability used in specifying fault-tolerance are also attributes of reliability. Additionally, although, as mentioned previously, availability could be viewed as a special case of reliability, it is commonly considered as an attribute of reliability. For instance, McCabe [50] shows that availability is needed to measure reliability since an unavailable system is not delivering the specified requirements at the first place. Fig. 8 shows the reliability concept taxonomy.

D. Security Definition, Requirements, and Taxonomy

Initially, security implied physical security. With expansion and wide-spread use of communications and computer networks, electronic, or more broadly, information security has also become prominent. Security has no precise definition. Broadly, it is the guarding or protection from unwanted happenings or actions. The concept of security is closely related to the confidentiality, integrity, and availability of assets [51].

According to Neumann in his technical report [51], "security must encompass dependable protection against all relevant

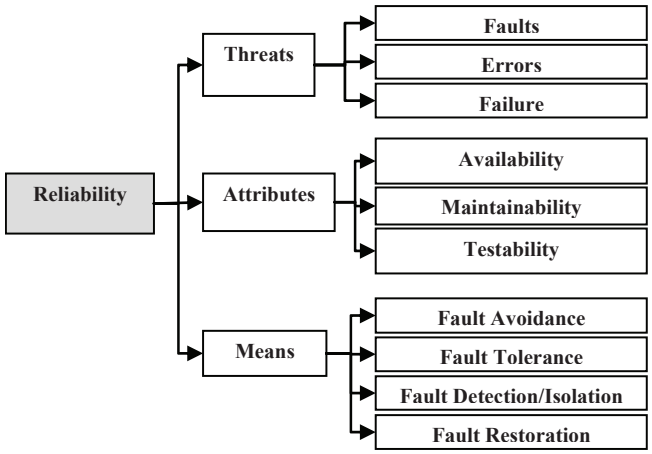

Fig. 8. Reliability Concept Taxonomy.

concerns, including confidentiality, integrity, and availability despite attempted compromises, preventing denials of service, preventing and detecting misuse, providing timely responses to threats, and reducing the consequences of unforeseen threats". Therefore, the security concept encompasses protection of systems, networks, and their components from different misappropriate actions as well as protection of information, *e.g.*, protection of data and programs from inappropriate actions. Security also entails prediction of possible threats, including insider abuses or misuses of the system, as well as outsider invasions or breaches. Consequently, there is much more to security than only providing confidentiality, integrity, and availability. Another way of expressing the concept is the resilience of a system to any type of malicious attacks [38].

It is worth mentioning that most "legacy" systems and networks are traditionally designed with security as an add-on feature as opposed to being an essential integrated property of the system design. Accordingly, security and security policies are poorly implemented, or often weak and largely neglected [9][32][46]. The required goal of security is to introduce measures and procedures that preserve confidentiality, integrity, availability, and other attributes such as authenticity and non-repudiation. Control mechanisms implement functions that help harden the system in order to prevent, detect, tolerate, and respond to security attacks. This is done using both theoretical and practical approaches such as cryptography, access controls, authentication, firewalls, risk assessments, policies, auditing, and intrusion detection and prevention systems as well as raising the human awareness and training.

Different researchers assign different attributes to security. In fact, there is no universal agreement about the expressions used in the security literature in describing its attributes. Sometimes these same attributes are referred to as security elements as in [35], or security services as in [60], or even security properties and goals as in [15]. In general, these attributes are considered as the basis for any security structures as well as the factors used to assess the security of a system. In literature, Reznik, [52], defines the following attributes for security: availability of the resources or information, confidentiality, integrity, and safety. Similarly, security has seven important

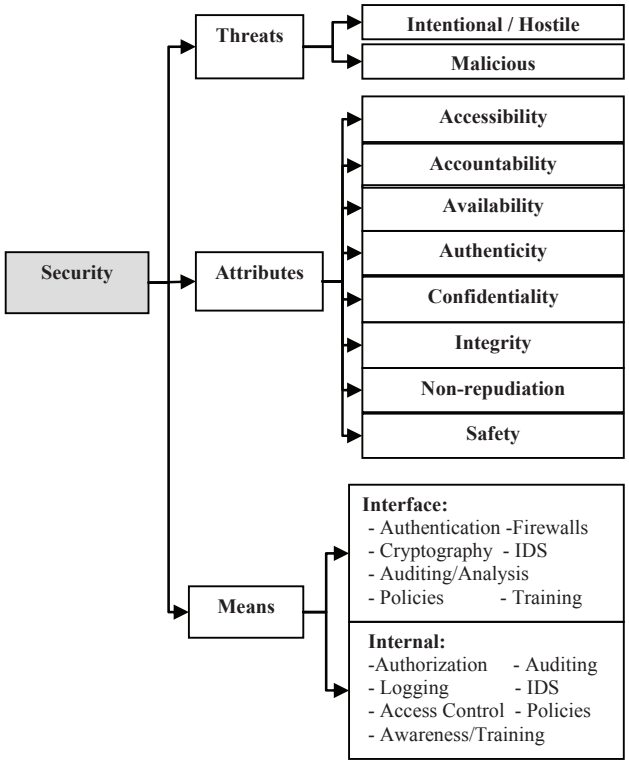

Fig. 9. Security Concept Taxonomy.

attributes as described in [35][60], namely: accountability, access control, availability, authenticity, confidentiality, integrity, and non-repudiation. For our context and in a similar fashion as the previously presented concepts' structures, we generalize a security concept structure as shown in Fig. 9 which brings together all various attributes in one comparable framework. It should be noted that when compared to other concepts, security does share some attribute with other concepts and at the same time it exclusively encloses other attributes.

E. Survivability Definition, Requirements, and Taxonomy

As a concept, survivability has its origin in a military context. It has been defined as a "property of a system, subsystem, equipment, process, or procedure that provides a defined degree of assurance that the named entity will continue to function during and after a natural or man-made disturbance; *e.g.*, nuclear burst" [12][61]. Variants to that definition have been introduced to take into account expectations about the performance of services provided by information systems as well as the element of time. It is defined as the capability of a system to "fulfill its mission in a timely manner, and in the presence of attacks, failures, or accidents" [12][31][32]. It has a single critical goal, to fulfill the mission in a timely manner. Some definitions require full service recovery, while others only specify mission fulfillment [12].

There is no clear understanding of the relationships between function, mission and service. Under survivability, services should have the capability to recognize and resist attacks, recover from them and adapt in the presence of them in order to diminish the effectiveness of future attacks. To characterize a system as survivable, it is necessary, first, that the services

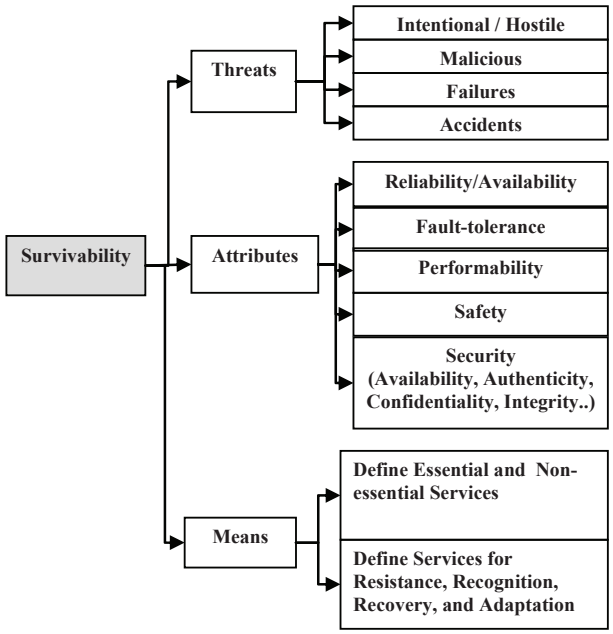

Fig. 10. Survivability Concept Taxonomy.

performed by the system in a hostile (accidentally, or deliberately) environment be categorized as essential and non-essential. Furthermore, the services expected of the system in the presence of attacks need to be prioritized and minimum operational levels specified. Park *et al*. [62] also describe a survivability strategy that can be set up in three steps: protection, detection, and response along with recovery. There are five types of requirements definitions that are relevant to survivable systems, namely: system/survivability requirements, use/intrusion requirements, development requirements, operations requirements, and evolution requirements as described by Ellison *et al*. [32].

The survivability concept applies to the entire system that offers defined services and not to any specific part or component of the system. The primary goal is fulfillment of the mission, namely, performance of essential services rather than full service recovery. By implication, a survivable system must first react and attempt to recover from a damaging effect prior to suffering complete breakdown. In other words, in a sustained hostile environment, a survivable system can either function with reduced capabilities, or function long enough to perform the specified essential services before breaking down completely.

Similar to the dependability concept, the survivability has minimum levels of quality attributes such as reliability, availability, safety, fault-tolerance, security, and performability [32][51]. Prioritization of services in a survivable system involves balancing these attributes. Following the pattern of the previous sections, the survivability concept structure is illustrated in Fig. 10.

IV. Analysis of Concept Definitions

A careful analysis of the definitions and properties of the five concepts indicates that there are interdependencies and overlapping characteristics among them. Some are explicit while others are nuanced and depend on the interpretation of the terms used to describe them. In this section we give a brief qualitative description of the relationships among the five concepts, before proceeding with the review of the quantitative measures associated with the respective attributes.

Examination of the definitions of the five terms reveals considerable overlap at the conceptual level, notwithstanding the claim of dependability [3] or survivability [32] to be an all-encompassing concept. The differences among the concepts could be attributed more to their respective connections with the disciplines within which each concept has originated rather than their perceived distinct characteristics.

With the exception of *security*, a close examination of the definitions of the remaining four concepts reveals that they could arguably be characterized as being synonymous with, perhaps, some nuanced differences. Consider the terms "performance of required functions under stated conditions for a specified period of time" and "ability of a system to deliver the required specific services that can justifiably be trusted" in the definitions of reliability and dependability, respectively. There is no logical distinction between performing "required functions under stated conditions" and delivering "required specific services". The additional qualifier "that can justifiably be trusted" does not indicate any additional distinguishing characteristic. When a system performs (delivers) functions (services) specified by the user, the system has fulfilled its mission to the satisfaction of the user.

Following the same reasoning, one can identify the similarities among dependability, reliability, survivability and fault-tolerance. Fulfillment of the mission in a timely manner (*survivability*) is no different from performance of required function under stated conditions for a specified period of time (*reliability*), or ability of the system to deliver required specific services (*dependability*), or ability of the system to continue normal operation (fault-tolerance). As in the case of dependability, the additional qualifiers "presence of attacks, failures, or accidents" (*survivability*), and "despite the presence of hardware or software faults" (*fault-tolerance*), are of explanatory nature rather than indicative of distinguishing characteristics that are unique to the specific concepts. On the contrary, the definitions of dependability and reliability contain the unequivocal requirement for the system to perform its mission, *i.e*., deliver the specified services under all system operating conditions, internal and external.

With reference to the concepts taxonomies presented in the previous section, one can realize some common characteristics among the four concepts, which include:

- Addressing similar threats (*i.e*., random or accidental errors, faults, and failures)
- Implying similar means of achievement (*i.e*., fault prevention, detection, removal, masking, forecasting, reconfiguration, restoration, and redundancy), and
- Their failures can be modeled using probabilistic distributions and random stochastic models

It should be noted that there is a subtle distinction between (dependability/reliability) and (fault-tolerance/ survivability) sets. Particularly, the former does not initially or explicitly imply the attribute of degraded service or performability but it implicitly uses it since it is embedded in its fault-tolerance attribute. That is, dependability/reliability is a measure of the

likelihood that all of the functions are performed correctly, while fault-tolerance/survivability along with its performability attribute is a measure of the likelihood that some subset of the functions is performed correctly [5]. The degraded service is vital and required for addressing the information infrastructure adequately [9][33][50].

Security on the other hand has a special position among the other four concepts. While in the definition of dependability, reliability is viewed as an attribute, security is considered as an independent concept sharing the attributes of availability, confidentiality, and integrity [3]. A critical examination of security at the conceptual level provides a different perspective. In the classical sense, security implies construction of a protective shell, physical and virtual, around the system. Initially, the protective shell involved physical protection, which can be considered as a subset of hardware security. In the information infrastructure, it comprises, in addition to hardware, software and management procedures. It is not difficult to argue that "unwanted happenings or actions" is equivalent to "attacks, failures, or accidents", or "presence of hardware or software faults". Unwanted happenings is a broad enough term to include failure or degradation of service of components or of the entire system. Thus, security, fault-tolerance and survivability define similar environmental conditions within which the system performs its required functions. By implication, if the shell is effective in protecting the system, then the mission of the system would be accomplished.

Unlike the other concepts, security in many "legacy" systems is not initially considered in the design but can be added latter on to ensure the protection and resistance of the system to attacks. Some important differences between security and the other concepts include the following:

- Security mainly addresses intentional/malicious threats, whereas the other concepts address accidental random failures, except for survivability which addresses both accidental and intentional threats
- Security threats are caused by human intent, thus they cannot be modeled or analyzed using quantitative probabilistic assessments[63][64][65]
- The means used to achieve the other concepts (*e.g.*, fault prevention, detection, removal, *etc.*) are different in implementation from the common security mechanisms (*e.g.*, cryptography, access controls, authentication, *etc.*)
- Security rarely mentions how the system can recover and maintain its services during and after attacks [32][40]. Thus, unlike the other concepts security does not explicitly imply maintainability attribute

The preceding analysis of the definitions of the four concepts has led to the conclusion that they are somewhat equivalent in terms of their definitions but may differ in some of their specific characteristics. It is argued that even if security has no precise definition, the broadly accepted view of the meaning of security makes it a derived or subordinate concept. The common core of the four concepts is the requirement on the system to perform the specified services within specified time constraints. Since a system exists and operates in a given environment, the effects of the environmental conditions, including intentional interferences, must be considered in order for the system to fulfill its specified mission.

In order to analyze in depth the relationships among the five concepts, it is necessary to identify and compare the performance indicators and metrics associated with each concept. A summary of the indicators associated with each of the concepts is given in the following section in order to provide an easy reference for the comparison of the concepts. The indicators are quantitative and qualitative. An objective comparison has meaning only when it is based on quantifiable parameters. Nevertheless, for completeness, qualitative indicators are also included.

V. Performance Evaluations and Measures

Following is an overview of the major tools used in evaluating performance under the five concepts without reference to specific applications or any particular environment. In practice, performance measures are associated with specific applications and operating environments. For example, in the IP-based telecommunications networks, the performance parameters for packet transfer are given by the QoS metrics such as the delay and loss parameters which generally embody the dependability of IP packet transfer [66]. Nevertheless, the purpose of this section is not to give an exhaustive review of all potential performance indicators which could be associated with the five concepts. Rather, the focus is on generic indicators associated with each of the concepts. Further, a survey of some of the developed analysis models is explored.

Since these concepts have evolved along with technology and over different times, this section follows their development path in order to better present their performance indicators and measures. Historically, reliability has been the oldest concept among these concepts. Thus, this section starts discussing the evaluation measures of reliability. It then proceeds with the other concepts in this order: fault-tolerance, security, dependability, and finally survivability. As illustrated here, when an all-encompassing concept evolves, it usually subsumes the performance evaluations and measures of the preceding concept and adds some others to them.

A. Reliability Performance Evaluations and Measures

Some of the common quantitative performance metrics of reliability and availability that are used by various methods and models generally include:

- Reliability, $R(t)$, and Unreliability, $Q(t)$, Functions
- Availability, $A(t)$, and Unavailability, $U(t)$, Functions
- Mean Time Between Failure $(MTBF)$
- Mean Time To Failure $(MTTF)$
- Mean Time To Repair $(MTTR)$
- Failure Rate (λ), and
- Repair Rate or Maintainability Parameter (μ)

The mathematical developments of these parameters are intentionally omitted. Readers can refer to various published references that discuss the reliability calculations, models, and methods using these parameters. Examples of these include the Trivedi's book [37], Leon-Garcia's book [67], and other books and papers found in [5][15][54][68][69].

The quantitative performance measures associated with reliability have precise mathematical definitions. Reliability, $R(t)$,

is a function of time that calculates the probability, $R(\tau)$, of uninterrupted service from $t = 0$ to $t = \tau$. This leads to a measure of reliability such as "mission-time", which is the time taken for the reliability of service to drop below a certain level [1].

The reliability function, $R(t)$, is also related to the *failure rate function* or *hazard function*, $z(t)$, which is the probability that a component which has functioned up to time t will fail in the next dt seconds [5]. This relation is given by:

$$R(t) = e^{-\int z(t)d(t)} \tag{1}$$

If we assume that the system is in a useful life stage where the failure rate function has a constant value of λ, then the solution is reduced to an exponential function of the parameter λ given by [5]:

$$R(t) = e^{-\lambda t} \tag{2}$$

This function assumes that the probability that a system will fail by a time t follows an exponential distribution. Although this assumption is commonly used in many system applications, there are a number of other well-known modeling schemes and probability distributions that are used to characterize system failures (*e.g.*, Normal distribution, Weibull distribution, and Longnormal distribution) [15][39][67]. Some of the most widely used reliability analysis techniques are combinatorial models and Markov models [5][38]. Combinatorial models (*e.g.*, Reliability Block Diagrams and Fault Trees) use probabilistic techniques for enumerating the various ways in which the system can remain operational. In contrast, Markov models are state-space methods that allow explicit modeling of complex relationships and transitional structures by capturing the stochastic behaviors of sequencing of events.

B. Fault-Tolerance Performance Evaluations and Measures

Although fault-tolerance was originated in a different field than reliability, they share the same goal of ensuring the performance of a system. The fault-tolerance of a system can be specified quantitatively and qualitatively by relating it to the reliability design goals and measures [58]. The qualitative evaluation goals are usually set by specifying some reliability design characteristics that include fail-safe mode, no single point of failure, and consistency specifications. In contrast, the quantitative evaluations are usually expressed through some of the same important reliability parameters such as the maximum allowable failure rate (λ), repair rate (μ), $MTBF, MTTF$, and $MTTR$.

Another important evaluation parameter in fault-tolerance is *fault-coverage* [5]. It is defined as the "measure of a system's ability to perform fault detection, fault location, fault containment, and fault recovery". It is also defined as the conditional probability that, given an existence of a fault, the system recovers. The fundamental problem with this parameter is that it is "extremely difficult to calculate" since it requires defining all faults that can occur in a system. However, there are various approaches to estimate fault coverage parameter and the most common one involve developing a list of all faults that can occur and, from that list, form other lists of faults that can be detected, located, contained, and recovered from. Then from these lists the fault detection, location, containment, and recovery coverage factors can be calculated by finding the fraction of each type of faults [5]. Some fault tolerance models also use probabilistic methods as described in [2][3][40][69].

Another point that deserves pointing out here is that fault-tolerance requires some type of redundancy which can take many various forms such as hardware, software, or time redundancy. Redundancy, however, can have an adverse impact on the performance of a system. For example, it can increase the length of transmitted data or increase the resource consumption.

C. Security Performance Evaluations and Measures

As mentioned previously, security is evaluated by means of informal and subjective assessments. There is no system-level methodology for quantifying security or mechanisms that effectively predict the behavior of unbounded systems [38][52]. Nevertheless, lack of quantitative measures is not an impediment to securing a system. There are in fact a large number of tools and mechanism that deal with strengthen the system and help in achieving security as previously mentioned in section III.D.

There are as well some attempts to evaluate the security by comparing changes in the system strategy or configuration [70]. For instance, one qualitative way to evaluate the security of any system is to consider three important factors: requirements, policy, and mechanisms. According to Bishop, requirements usually describe the desirable security goals, the policy typically deals with the procedures that need to be taken to achieve the goals, and the mechanisms implement the policy by using all available tools [47]. Implementing a strict security policy, however, can affect the flexibility of a system in adopting some short term changes.

As in the case with dependability, some quantitative and qualitative methods and models have been developed to evaluate the system security in specific environments. The emphasis of some of these models is usually on specific attributes of security such as confidentiality (*e.g.*, Bell-LaPadula Model), integrity (*e.g.*, Biba Model), or both (*e.g.*, Chinese Wall Model) [47].

Also various risk assessment evaluation models have been developed for evaluating security by analyzing the failure frequencies, threats, vulnerabilities, severities, outage durations, and countermeasures [3][15][42][65]. The common formula for estimating risk, ρ, is:

$$\rho = T \times V \times C \tag{3}$$

where T is the threat, or frequency, of a problem expressed as a percentage, V is vulnerability or probability that a threat will impact a system's operation, and C is the expected cost or damage resulting from the threat [15]. The result, ρ, is a monetary value. Quantitative risk assessment methods are usually (i) based on probability theory, (ii) require professional experts to analyze the risk, (iii) require knowledge of all available controls, (iv) require quantification of all vulnerabilities or threats, (v) and require establishment of a value for all assets. However, these conditions are difficult if not impossible to meet; therefore it is hard to obtain adequate risk assessment evaluation for security [39].

Additionally, unlike the random accidental failures assumptions in reliability analysis, the source of security threats is different which also makes it hard to accurately model security attacks using classical stochastic models. The threat probability is not a static function and statistical probabilities assumptions do not necessary hold. Thus, the performance analysis can successfully be applied in computers and communication systems since they are predictable; however, the security analysis cannot effectively be realized because it deals with humans as well as systems where the behaviors are not predictable [3][38][63].

It is worth noting that one of the most widely used security evaluation documents produced by National Computer Security Center, NCSC, is the US Trusted Computer System Evaluation Criteria (TCSEC), known as "*The Orange Book*" [46][52]. Its main aim is to ensure that products and systems achieve a high degree of security. Further, there is also the more advanced set of security evaluation standards created by the international security community called *"The Common Criteria"* [44][46]. All of these criteria were created for the evaluation process to recognize a definition of acceptable levels of security for computer systems

D. Dependability Performance Evaluations and Measures

Dependability is the most comprehensive concept for modeling complex systems taking a top-down approach. As it was discussed in section III.A, some attributes of dependability are quantitative, while others are qualitative. The former can be measured at the interface between the user and the system providing the service and evaluated using objective criteria. On the other hand, qualitative performance parameters can only be evaluated subjectively by each user preventing the establishment of a common reference point that is necessary to comparative performance evaluation. In this respect, the inability to measure all of the parameters that affect the dependability of a design since some of them are unquantifiable and unknown especially for large or unbounded distributed systems, presents difficulties in the development of a rigorous approach for the measurement of dependability [1]. In effect, dependability, generally, is not a mathematically well-defined concept.

However, in some cases, (*e.g.*, for bounded systems and when referred to as reliability and availability), it can be evaluated using some of its attributes or combinations of these attributes such as reliability, availability, and maintainability as presented previously in the reliability performance evaluations and measures section. In other words, dependability sans the qualitative attributes can be the subject of comparative analysis.

Although some dependability attributes have no mathematical or formal definition, mechanisms exist to help reach specific goals associated with such attributes. For example, there are no metrics for security [40]. Nevertheless, there are many security models, mechanisms and policies that have been developed in order to achieve confidentiality, integrity, authenticity and other security attributes as illustrated in the security performance evaluation and measures section.

Other ways to evaluate dependability are using one of the two main approaches for probabilistic fault-forecasting. These approaches are modeling and testing as illustrated extensively in [3][38]. Dependability evaluation has been realized by the computer engineering community over the past twenty years during which many dependability evaluation tools and models were developed. Some examples are: SAVE, SHARPE, UltraSAN and MEADEP [2][71][72].

E. Survivability Performance Evaluations and Measures

There are several approaches described in the literature that define, describe, and quantify methods and metrics to achieve as well as evaluate survivability. Similar to the dependability evaluation, survivability evaluation is not mathematically well-defined since this concept does not refer to a measurable sense [68]. It can be evaluated using its qualitative and quantitative attributes especially the reliability, availability, and fault-tolerance attributes which can be statistically modeled using the parameters of $MTTF, MTTR, MTBF$, failure rate, repair rate, and fault-coverage.

The following is a brief survey of some survivability evaluation models. In general, there are two broad types of survivability measures [68]. The first one is called *conditional or Given Occurrence of Failure* (GOF) models which are considered as design-oriented measures where each relevant failure scenario is first postulated to have occurred, then an assessment of survivability is made. An example of this class is maintainability measures. The other one is called *Random Occurrence of Failure* (ROF) models. These models depend on the assumption that failures can be characterized by random variables with given probability distribution functions. These are closely related to the fields of reliability and availability. However, similar to security evaluations, the intentional attacks still represent an issue for survivability evaluations.

Koroma *et al*. [29] present a mathematical definition of survivability by proposing a statistical model to determine the steady-state probability, availability, and failure frequency using Continuous Time Markov Chain (CTMC). There is also a summary in [11] for the most significant results of variety of network models that can assist in the analysis and design of survivable networks. The survivability evaluation criteria are classified as either deterministic or probabilistic, and a discussion of both analysis and synthesis of these criteria using the graph theory is demonstrated.

Another similar model described by Zolfaghari *et al*. in [8] where the authors illustrate two basic approaches for survivability analysis. The first approach uses probability of network failures and rates of repair (restore) to calculate various probabilistic measures of network availability. The second approach is a conditional approach, defining measures of a network after given failure events have occurred. Nevertheless, some limitations still exist such as not all failures can be determined and not all intentional attacks can be modeled.

Different survivability models for distributed system are also proposed. For instance, in [62], a discussion about static, dynamic, and hybrid recovery models for survivable distributed systems are presented. A comparison between these models is presented in terms of the following tradeoffs: simplicity, resource efficiency, adaptation, service downtime, immunization, and robustness. Another method for enhancing

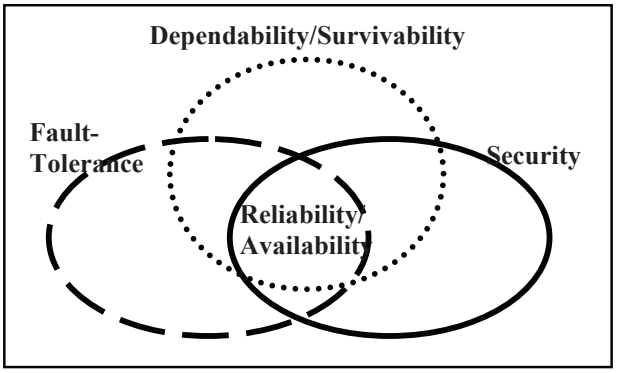

Fig. 11. Relationship between the concepts' domains.

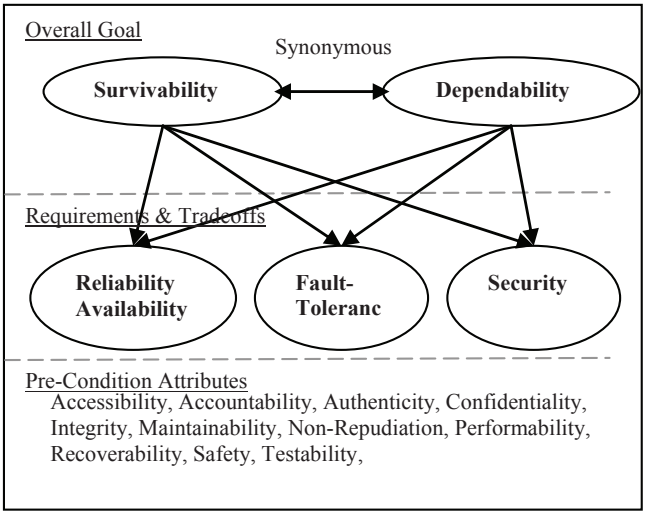

Fig. 12. Relationship between the concepts' levels.

survivability in unbounded systems is also introduced in [53] which uses practical emergent algorithms.

Additionally, an experimental method shown in [73] uses a measure of time as a prime metric to recover from component failure as the key to network survivability. In a similar fashion, [68] quantifies the network outages using three parameter which are unservability (Us), duration (D), and weight (W). The Us, D, W triple is considered the basis for measuring and quantifying network failures and their impact on the services and users.

VI. Comparison of Concepts

A side-by-side listing of the five concepts on the basis of their definitions, goals, means to achieve them, their attributes, and evaluation criteria is illustrated in Table I.

The preceding discussion of the concepts has revealed that some of these concepts had existed from the beginning in the field of system designs and they are well entrenched in their respective fields. Examples of these are reliability and fault-tolerance. Others, however, are new that have evolved with technology and used as all-encompassing umbrella terms such as dependability and survivability. The idea of decomposing the general concepts (*e.g.*, dependability and survivability) into several attributes (*e.g.*, reliability and availability) is very helpful in this area. In a sense, the decomposition of a higher level quality concept into the objective lower level factors helps reveal quantitative performance characteristics.

Following is an overview of some major observations of the concepts comparison in terms of their (i) interrelations and (ii) common performance indicators.

A. Interrelations Among Concepts

It is clear that these concepts are related as a result of their evolution. Different concepts have different meanings depending on the context in which they are applied in. A cursory look at Table I shows that the concepts have some overlapping characteristics and some redundancies in their use. In this respect, although they are somewhat conceptually equivalent, these concepts when compared to each other they are neither disjoint nor identical, but somewhere in between. Fig. 11 and Fig. 12 give an overview of the relationship among them.

With reference to Table I and Fig.11, although the all-encompassing concepts of dependability and survivability are considerably equivalent in their goals and in some of their attributes, the survivability concept places more emphasis on the intentional malicious types of threats since it was evolved from the security and military aspects.

Moreover, the component-level concepts of reliability and availability are also the most commonly used objective attributes among all five concepts discussed in this paper, because they have been in use for a long time. To be able to address issues related to the complex infrastructures such as information infrastructure, reliability and availability features needs to be supplemented together with some fault-tolerance and security features. In particular, the vital attribute of fault-tolerance that is needed here is performability, which implies the degraded performance property; whereas for security, preserving some of its qualitative features such as accessibility, authenticity, and safety are essential.

Although there is some parallelism among the five concepts, there is also some degree of hierarchy among them, as shown in Fig. 12. In particular, the all-encompassing dependability and survivability concepts can be placed at the top level and all other concepts and their corresponding qualitative and quantitative attributes can be considered as other design requirements that assist in the overall design.

B. Common Performance Indicators Among Concepts

With the various attributes associated with the five concepts, one can identify some of them as qualitative attributes and some others as quantitative ones. Table II illustrates a list of these attributes along with an indication if they are measurable or not. A closer look at this list reveals that the measurable attributes are in fact the common ones among the studied concepts. These common measurable attribute include availability, fault-tolerance, maintainability, and reliability. Undoubtedly, the more measurable attributes contained in a concept, the easier it is to evaluate the performance of that concept and balance it with all other attributes.

The measurable attributes are in fact associated with some quantifiable parameters that assist in determining the achieved degree of that attribute. The preceding analysis of the evalua-

TABLE I
SIDE-BY-SIDE COMPARISON OF CONCEPTS.

	Dependability	Fault-Tolerance	Reliability	Security	Survivability
Definition and Goal	An umbrella concept defined as the ability to deliver required services during its life cycle that can justifiably be trusted	Ability to continue the performance of its tasks in the presence of faults	A conditional probability that a system performs its intended tasks correctly throughout a complete interval of time	Ability to guard and protect from unwanted happenings or actions and preserve confidentiality, integrity, and availability	Ability to fulfill its mission in a timely manner in the presence of attacks, failures, or accidents
Means	- Fault-prevention - Fault tolerance - Fault removal - Fault forecasting	- Error detection - System recovery - Fault masking - Reconfiguration - Redundancy	- Fault avoidance - Fault tolerance - Fault detection and isolation - Fault Restoration	- Interface: IDS, cryptography, auditing, analysis, firewalls, authentication. - Internal: IDS, access control, authorization, auditing/logging, - Policies - Awareness and training	- Define essential and nonessential services - Define survivability services for attack resistance, recognition, and recovery.
Attributes	- Availability - Confidentiality - Integrity - Maintainability - Reliability - Safety - Security	- Availability - Maintainability - Performability/ Graceful Degradation - Testability	- Availability - Maintainability - Testability	- Accessibility - Accountability - Authenticity - Availability - Confidentiality - Integrity - Non-repudiation - Awareness and - Safety	- Availability - Fault-tolerance - Performability - Reliability - Safety - Security (confidentiality, integrity, availability, authenticity)
Cause of Threats and Evaluation Criteria	- Errors, faults, failures - Caused by random, accidental, and unintentional events in hardware or rare events in software, and this randomness can be quantified or modeled	- Errors, faults, failures - Caused by random, accidental, and unintentional events in hardware or rare events in software, and this randomness can be quantified or modeled	- Errors, faults, failures - Caused by random, accidental, and unintentional events in hardware or rare events in software, and this randomness can be quantified or modeled	- Intentional and hostile - Malicious - Failures are caused by human intent, resulting in security failures which are hard to model	- Intentional attacks, failure, and accidents include all potential damaging events - Randomness can be assumed for accidental faults, but not for attacks

TABLE II
MEASURABLE AND IMMEASURABLE ATTRIBUTES FOR THE CONCEPTS.

Concept Attributes		Measurability	
No.	Attributes	Measurable	Immeasurable
1	Accessibility		x
2	Accountability		x
3	Authenticity		x
4	Availability	x	
5	Confidentiality		x
6	Fault-Tolerance	x	
7	Integrity	x	
8	Maintainability	x	
9	Non-Repudiation		x
10	Performability	x	
11	Reliability	x	
12	Safety		x
13	Security		x
14	Testability		x
15	Unreliability	x	
16	Unavailability	x	

TABLE III
SET OF QUANTIFIABLE PARAMETERS FOR THE COMMON ATTRIBUTES.

Reliability / Fault Tolerance	Availability / Maintainability
MTTF	Steady State Availability, A_{ss}
Reliability Function, $R(t)$	Unavailability, U
Failure Rate Function, $z(t)$	MTBF
Unreliability Function, Q(t)	MTTR
Failure Rate (λ)	Repair Rate (μ)
Fault Coverage	

tion measures of the five concepts has led to the identification of some common quantifiable parameters associated with the measurable attributes. The set of quantifiable parameters for each of these measurable attribute is shown in Table III. It is worth noting that in some cases different terms are used to refer to similar parameters. For example, the terms $MTTR$, $MTBF$, and repair rate (μ), in the survivability terminology are called restorability while in reliability are measures of maintainability. In a similar fashion, fault-tolerance modeling uses some parameters that are related to the reliability modeling such as the $MTTF$, failure rate (λ), reliability function $R(t)$, and failure rate function $z(t)$.

In fact, reliability and availability have the most quantifiable parameters that can be considered as performance indicators when designing any network or system. They contribute sig-

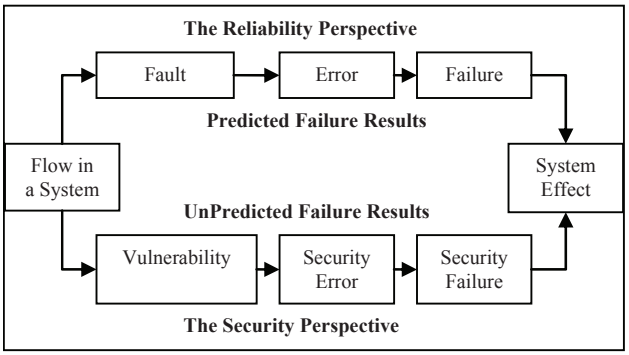

Fig. 13. Reliability vs. Security Perspective of Failure.

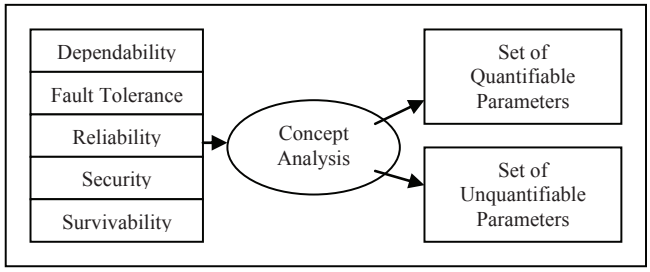

Fig. 14. Overall evaluation sets for the studied concepts.

nificantly in optimizing the design and achieving measurable performance objectives. It should be noted that without availability none of these concepts could exist since an unavailable system is the same as a non-existing one. Further, since all networks and systems are subject to failure, methods for their reliability and availability modeling and measures are needed. Consequently, great emphasis has been placed on the development of tools and procedures for performing reliability and availability analysis; most have their origins in probability theory [37].

One should realize that analysis of a design concerning reliability, availability, and fault-tolerance usually focuses on *the accidental or random faults* which are based on concerns such as component aging or benign failure rates. Therefore, modeling these attributes can be performed easily; hence, the ability of optimizing the design can be achieved. However, there is another failure cause that should be addressed, namely, *the malicious or intentional threats*. These threats are mainly associated with the security concerns. There is no accurate statistical modeling technique available for these types of threats due to the fact that malicious attacks do not usually follow predictable patterns. Because the root causes of system failure in reliability or generally in dependability context (*e.g.*, random accidental failures) are fundamentally different from the root causes of security violations (*e.g.*, intentional attacks), then it is difficult to accurately represent security events using classical stochastic models [38][39]. Fig. 13 shows the pathology of these two different perspectives.

Although they are different, an attempt to develop an architecture with a comprehensive set of mechanisms for tolerating both accidental faults and malicious attacks in complex systems is presented by Verssimo *et al*. [74]. The European project MAFTIA (Malicious-and Accidental-Fault Tolerance for Internet Applications) uses ideas from fault tolerance to automatically detect, contain, and recover from attacks [74].

When a failure occurs, the network user may not be able to differentiate whether the cause of the failure is resulted from an accidental fault of an intentional attack. Nevertheless, identifying the causes of these disruptions is sometimes essential in some cases especially when dealing with the security concerns. For instance, it is important to determine the sources and motivations of system threats in order to identify and apply the appropriate avoidance, deterrence, and recovery controls. Generally, it is difficult to differentiate and implement a specific countermeasure if no clarification is performed on specifying the differences between threats.

Although the lack of quantification and modeling in security have created some issues in design optimization and cost effectiveness, various qualitative theoretical and practical criteria (*e.g.*, cryptography, access control, and policies) have been developed to preserve the intended security attributes.

The methodology presented in this paper for comparing concepts leads to the conclusion that evaluation of performance of complex infrastructure requires the use of two complementary evaluation methods, qualitative and quantitative. Fig. 14 illustrates the models for this. Incorporating and balancing both approaches is an essential task for better design optimization, especially since pure quantitative or qualitative evaluation measures may not be sufficient to exhaust all different aspects of these concepts. The evaluation approaches are categorized as two sets, namely:

- *The Set of Unquantifiable Parameters:* These can be considered as input design characteristics or safeguards that can be implemented into the system to increase its ability to resist attacks [40]. These applied mainly for security concerns since no real evaluation metric is available. Examples of these are risk analysis methods, intrusion detection systems, security mechanisms, management policies, *etc*.
- *Set of Quantifiable Parameters:* These can be considered as initial system design characteristics and requirements that can be incorporated and controlled into the system. They deal with system failures such as probability of failure, or reliability, availability, and performability as illustrated in Table III.

VII. Conclusion

In this paper, we have developed a conceptual framework for integrating the component-level design approaches used at the physical level of the OSI Reference Model with the information infrastructure design and its performance requirements as stated at the application level. The approach involves the integration of the concepts of dependability, fault-tolerance, reliability, security and survivability on the basis of their definitions, attributes, relationships, and performance evaluation measures.

Although quantifying qualitative concepts is still considered an open issue, nevertheless, lack of quantitative measures is not an impediment to achieving them. The comparison methodology presented in this paper led to the conclusion that

evaluation of performance of complex infrastructure requires the use of two complementary evaluation methods, qualitative and quantitative.

Regardless of how ambiguous any concept is, if a designer wants to achieve the ultimate goal of designing better systems with the desired services then it is important to clearly specify the following:

- *The System Requirements:* that is by looking at the system environment that we are trying to make dependable, fault-tolerance, reliable, secure or survivable. This includes the technical configurations, functions, specifications, services and physical surroundings of the system.
- *The Goals and Objectives:* that is understanding what you want the system to do for you is helpful when determining what features you need. The set of quantifiable parameters are useful here.
- *The System Policy:* that includes the goals of policy in terms of the five studied concepts as well as generic management goals and descriptions for all users. The set of unquantifiable parameters can be helpful here.
- *Other Requirements and Constraints:* this will articulate common requirements and constraints that might affect the selection of right tools and mechanisms to achieve the intended concepts. This includes external requirements and resource constraints.

To achieve system design optimization, these points need to be correctly defined, properly specified and balanced.

The work described in this paper is only the first step of a process to develop a unified design model for information infrastructures. The approach takes into account all of the characteristics of the five concepts and establishes a common baseline and standardization for their definitions and attributes. It also helps in the clarification of ambiguous terms and their relationship as well as the mapping of subjective user-oriented requirements into objective performance parameters.

The methodology developed in this paper can be applied to specific systems such as telecommunication networks. In telecommunication network engineering, these concepts are related and closely tied to the Quality of Service (QoS). Generally, QoS is defined as a set of qualitative and quantitative characteristics of a network that are necessary for obtaining the required functionality of an application [66]. The term QoS is used to measure the performance of IP networks with respect to the transport of data.

The idea is that, within a single network, the common attributes found in Table III are translated into the QoS network design parameters (*e.g.*, packet loss, delay, or jitter). In a sense, the concepts of dependability, fault-tolerance, reliability, security, and survivability are comparable to that of quality of service. The term QoS encompasses many aspects including reliability, availability, and fault tolerance. A system level design can be optimized by mapping QoS performance requirements at the application level with those at all levels of the OSI model. The future work will mainly aim at mapping the IP network QoS performance parameters (*e.g.*, packet loss and delay) into the application performance characteristics.

ACKNOWLEDGMENT

The authors gratefully acknowledge the manuscript reviewers who provided them with their valuable feedbacks and comments.

REFERENCES

[1] N. Edwards, "Building dependable distributed systems", ANSA, Feb., 1994, APM Ltd., Cambridge, U.K.

[2] A. Avizienis, J.-C. Laprie, and B. Randell, "Fundamental concepts of dependability", Research Report No. 1145, LAAS-CNRS, Apr. 2001.

[3] A. Avizienis, J-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing", *IEEE Trans. Dependable and Secure Computing (TDSC)*, vol. 1, no. 1, Jan.-Mar. 2004, pp. 11-33.

[4] B. Melhart, and S. White, "Issues in defining, analyzing, refining, and specifying system dependability requirements", *Proc. 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2000)*, Edinburgh, Scotland, UK., Apr. 3-7, 2000, pp. 334-311.

[5] D. Pradhan, *Fault-Tolerant Computer System Design*, 1st Ed, Prentice Hall, Inc., Upper Saddle River, NJ, 1996, pp. 5-14.

[6] D. Siewiorek, R. Chillarege, and Z. Kalbarczyk, "Reflections on industry trends and experimental research in dependability", *IEEE Trans. Dependable and Secure Computing (TDSC)*, vol. 1, no. 2, Apr.-Jun. 2004, pp. 109-127.

[7] W.S. Harrison, A.W. Krings, N. Hanebutte, and M. McQueen, "On the performance of a survivability architecture for networked computing systems", *Proc. 35th Hawaii International Conference on System Sciences*, Jan. 2002.

[8] A. Zolfaghari and F.J. Kaudel, "Framework for network survivability performance", *IEEE J. Sel. Areas Commun.*, vol. 12, no. 1, Jan. 1994, pp. 46-51.

[9] K. Kyamakya, K. Jobmann, and M. Meincke, "Security and survivability of distributed systems: an overview", *Proc. 21st Century Military Communications Conference (MILCOM 2000)*, vol. 1, Oct. 2000, pp. 449-454.

[10] R. Ellison, D. Fisher, R. Linger, H. Lipson, T. Longstaff, and N. Mead, "Survivability: protecting your critical systems", *IEEE Internet Computing Magn.*, vol. 3, no. 6, Nov.-Dec. 1999, pp. 55-63.

[11] H. Frank, and I. T. Frisch, "Analysis and design of survivable networks", *IEEE Trans. Commun. Technol.*, vol. 18, no. 5, Oct. 1970, pp. 501-519.

[12] P. Tarvainen, "Survey of the survivability of IT systems", *Proc. 9th Nordic Workshop on Secure IT Systems (NORDSEC 2004)*, Nov. 4-5, 2004, pp.15 - 22.

[13] Object Services and Consultant, Inc. *Quality of Service (QoS)*, DARPA Contract DAAL01-95-C-01121997, Jan. 1997; http://www.objs.com/Survivability/QoS and Survivability.htm

[14] F. Halsall, *Data Communications, Computer Networks and Open Systems*, 4th Ed, Addison-Wesley Publishing Company, NY, 1996, pp. 13-18.

[15] M. Liotine, *Mission-Critical Network Planning*, Artech House Publishing, Inc., Norwood, MA., 2003, pp. 31-59.

[16] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 3rd Edition, Addison-Wesley, 2005.

[17] M. Rudack, K. Jobmann, A. Pajares, and M. Esteve, "Policy-Based Quality of Service Mapping in Distributed Systems", *Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS2002)*, Apr. 15-19, 2002, pp. 947 - 949.

[18] L. DaSilva, "QoS Mapping along the Protocol Stack: Discussion and Preliminary Results", *Proc. IEEE International Conference on Communications (ICC'00)*, vol. 2, Jun. 18-22, 2000, pp. 713-717.

[19] J.-F. Huard, and A. A. Lazar, "On QOS Mapping in Multimedia Networks", *Proc. The Twenty-First Annual International Computer Software and Applications Conference (COMPSAC '97)*, Aug. 13-15, 1997, pp. 312 - 317.

[20] N. K. and J. Smith, "The QoS Broker", *IEEE Multimedia*, vol. 2, no. 1, 1995, pp. 53-67.

[21] J. Jingwen, and K. Nahrstedt, "QoS Specification Languages for Distributed Multimedia Applications: A Survey and Taxonomy", *IEEE Multimedia*, vol. 11, no. 3, July-Sept. 2004, pp. 74- 87.

[22] N. Kyriakopoulos and M. Wilikens, *Dependability and complexity: exploring ideas for studying open systems*, in Report EUR 19797 EN. Brussels, Belgium, EC Joint Research Centre, 2001.

[23] J. C. Lapie (Ed.), *Dependability: Basic Concepts and Terminology*, Springer-Verlag, New York, NY, Wien, 1992.

[24] IEEE Std 610 - IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries, 1990.

[25] P. Koopman, "Toward a scalable method for quantifying aspects of fault tolerance, software assurance, and computer security", *Proc. Computer Security, Dependability, and Assurance: From Needs to Solutions (CSDA'98)*, Nov. 1998, pp. 103-131.

[26] G. Trouessin, "Quantitative evaluation of confidentiality by entropy calculation", *Proc. 4th IEEE Computer Security Foundations Workshop (CSFW'91)*, Franconia, New Hampshire, Jun. 18-20, 1991, IEEE Comp. Soc., 1999, pp.12-21.

[27] J. von Neumann, *Probabilistic logics and the synthesis of reliable organisms from unreliable components*, in C. E. Shannon and J. McCarthy, Eds., Automata Studies, Annals of Math Studies, No. 34, Princeton: Princeton University Press, NJ, 1956, pp. 43-98.

[28] A. Avizienis, "Design of Fault-Tolerant Computers", *Proc. of 1967 Fall Joint Computer Conf., AFIPS Conf*, vol. 31, Thompson Books, Washington, D.C., 1967, pp. 733-743.

[29] J. Koroma, W. Li, and D. Kazakos, "A generalized model for network survivability", *Proc. Richard Tapia Celebration of Diversity in Computing Conference 2003 (TAPIA'03)*, Atlanta, Georgia, Oct. 15-18, 2003, ACM 2003, pp. 47-51.

[30] T1A1.2 Working Group on Network Survivability Performance, Technical Report on Enhanced Network Survivability Performance, Feb. 2001.

[31] R. J. Ellison, R. C. Linger, T. Longstaff, and N. R. Mead, *Case study in survivable network system analysis*, (CMU/SEI-98-TR-014, ADA355070). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1998.

[32] R. Ellison, D. Fischer, R. Linger, H. Lipson, T. Longstaff, and N. Mead, *Survivable network systems: an emerging discipline*, (Report CMU/SEI-2001-TN-001), Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, Mar. 2001.

[33] J.C. Knight, E. A. Strunk, and K J. Sullivan, "Towards a rigorous definition of information system survivability", *Proc. DARPA Information Survivability Conference and Exposition (DISCEX 2003)*, Washington, DC, Vol. 1, Apr. 2003, pp. 78-89.

[34] J. Wylder, *Strategic Information Security*, Auerbach Publications, CRC Press LLC, NY, 2004, pp. 47-61.

[35] J. E. Canavan, *Fundamental of Network Security*, Artech House, Inc., MA, 2001, pp.1-49.

[36] T. R. Peltier, *Information Security Risk Analysis*, Auerbach Publications, CRC Press LLC, FL, 2001, pp.1-21.

[37] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Application*, 2nd Ed, Wiley-Interscience Publications, John Wiley and Sons, Inc., NY, 2002, pp. 1-4.

[38] D. Nicol, W. Sanders, and K. Trivedi, "Model-based evaluation: from dependability to security", *IEEE Trans. Dependable and Secure Computing (TDSC)*, vol. 1, no. 1, Jan.-Mar., 2004, pp. 48-65.

[39] D. Parker, *Fighting Computer Crime: A New Framework for Protecting Information*, Wiley Computer Publishing, John Wiley and Sons, Inc., NY, 1998, pp. 364-370.

[40] E. Jonsson, "An integrated framework for security and dependability", *Proc. of the 1998 New Security Paradigm Workshop (NSPW '98)*, Sep. 22-25, 1998, Charlottesville, Virginia, ACM Press, 1999.

[41] J. Lala, "DARPA's path to self-regenerative systems", *Proc. 42nd IFIP WG Meeting, Workshop on Dependability and Survivability*, Hilton Head Island, SC, Jun. 27-July 1, 2002.

[42] E. Orlandi, "Computer security: a consequence of information technology quality", *Proc. 1990 IEEE Int. Carnahan Conference on Crime Countermeasures*, Security Technology, Oct. 1990, pp.109-112.

[43] National Institute of Standards and Technology, Special Publication 800-37, Guide for Security Certification and Accreditation of Federal Information Systems, May 2004.

[44] Syntegra, The Common Criteria: An Introduction, Jan. 2004.

[45] M. J. Attallah, E. D. Bryant, and M. R. Stytz, "A survey of anti-tamper technologies", CrossTalk - *Journal of Defense Software Engineering*, vol. 17, no.11, Nov. 2004, pp. 12-16.

[46] G. B. White, E. A. Fisch, and U. W. Pooch, *Computer System and Network Security*, Boca Raton, FL, CRC Press, 1996.

[47] M. Bishop, *Computer Security: Art and Science*, Addison Wesley, Pearson Education, Inc., 2003, pp. 95-184.

[48] J.-C. Fabre, Y. Deswarte, and B. Randell, "Designing Secure and Reliable Applications Using Fragmentation-Redundancy-Scattering: An Object-Oriented Approach", *Proc. 1st European Dependable Computing Conf. (EDCC-1)*, Berlin, Germany, Oct. 1994, Springer-Verlag, LNCS 852, pp.21-38.

[49] A. Avizienis, "Toward Systematic Design of Fault-Tolerant Systems", *IEEE Computer*, vol. 30, no. 4, 1997, pp.51-58.

[50] James McCabe, *Practical Computer Network Analysis and Design*, Morgan Kaufmann Publishers, Inc., CA, 1998, pp. 1-9.

[51] P. Neumann, *Practical architectures for survivable systems and networks*, Technical report, Final Report, Phase Two, Project 1688, SRI International, Menlo Park, California, Jun. 2000.

[52] L. Reznik, *Which models should be applied to measure computer security and information assurance?*, *Proc. 12th IEEE International Conference on Fuzzy Systems (FUZZ'03)*, vol. 2, 25-28 May, 2003, pp. 1243-1248.

[53] D. Fisher, and H. Lipson, "Emergent algorithms-A new method for enhancing survivability in unbounded systems", *Proc. 32nd Annual Hawaii International Conference on System Sciences (HICSS-32)*, Maui, HI, 5-8 Jan., IEEE Computer Society, vol. Track7, 1999, pp. 10.

[54] F. E. Oliveto, "The four steps to achieve a reliable design", *Proc. 1997 National Aerospace and Electronics Conference, (NAECON)*, vol. 1, Dayton, OH., Jul. 14th, 1997, pp. 446-453.

[55] R. B. Vaughn Jr., A. Sira, and D. A. Dampier, "Information security system rating and ranking", Software Technology Support Center (STSC), CrossTalk - *Journal of Defense Software Engineering*, vol. 15, no. 5, May 2002, pp. 30-32.

[56] J. F. Meyer, "Closed-Form Solutions of Performability", in IEEE Trans. Comput., vol. C-31, no. 7, July 1982, pp.648-657.

[57] N. Kyriakopoulos and M. Wilikens, "Dependability of complex open systems: a unifying concept for understanding internet-related issues", *Proc. 3rd Information Survivability Workshop (ISW2000), IEEE Comp. Soc. Press*, Oct. 2000.

[58] W. Heimerdinger, and C. Weinstock, *A conceptual framework for system fault tolerance*, Technical Report CMU/SEI-92-TR33. ESC-TR-92-033. SEI., Oct. 1992.

[59] A. Avizienis, "Framework for a Taxonomy of Fault-Tolerance Attributes in. Computer Systems", *Proc. of the IEEE 10th Annual Int. Symp. on Computer Architecture archive*, 1983, Stockholm, Sweden, pp.16-21.

[60] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Prentice Hall, Inc., NJ, 1999.

[61] Telecommunications: Glossary of Telecommunication Terms, Federal Standard 1037C, August 7, 1996

[62] J. Park, and P. Chandramohan, "Static vs. dynamic recovery models for survivable distributed systems", *Proc. 37th Annual Hawaii International Conference on System Science*, Maui, HI, 5-8 Jan. 2004, IEEE Comp. Soc. Press, 2004, pp. 55-63.

[63] K. H. Kim, "Incorporation of security and fault tolerance mechanisms into real-time component-based distributed computing systems", *Proc. 20th Symposium on Reliable Distributed Systems (SRDS 2001)*, Oct. 28-31, New Orleans, LA, IEEE Comp. Soc., 2001, pp. 74-75.

[64] P. G. Neumann, *Principled assuredly trustworthy composable architectures*, (Emerging Draft of the) Final Report, SRI International, Menlo Park, California, Mar. 15, 2004.

[65] S. A. Butler, "Security attribute evaluation method: a cost-benefit approach", *Proc. of the 22nd International Conference on Software Engineering (ICSE2002)*, May 19-25, 2002, Orlando, Florida, ACM 2002, pp. 232-240.

[66] N. Seitz, "ITU-T QoS standards for IP-based networks", *IEEE Communications Magazine*, vol. 41, no. 6, Jun. 2003, pp. 82-89.

[67] A. Leon-Garcia, *Probability and Random Process for Electrical Engineering*, 2nd Ed, Addison-Wesley Publishing Company, NY, 1994, pp. 150-155.

[68] W. D. Grover, *Mesh-Based Survivable Transport Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*, New Jersey, Prentice-Hall PTR, 2004, pp. 103-172.

[69] S. F. Andler, B. Lindstrom, and M.R. Barbacci, *Distributed real-time systems: dependability in critical systems, software reliability, and security in critical systems*, University of Skvde, Distributed Real-Time Systems Course, Fall 2002.

[70] R. Ortalo, Y. Deswarte and M. Kaniche, "Experimenting with Quantitative Evaluation Tools for Monitoring Operational Security", *IEEE Trans. Software Eng.*, vol. 25, no. 5, Sep- Oct. 1999, pp. 633-650.

[71] Dong Tang, Myron Hecht, Xuegao An, and Robert Brill "MEADEP and its application in dependability analysis for a nuclear power plant safety system", *Proc. 1997 Symposium on Nuclear Power Systems (SNPS'97)*, Nov. 11-14, 1997, pp. 916-920.

[72] D. Tang, M. Hecht, J. Miller, and J. Handal, "MEADEP - a dependability evaluation tool for engineers", *IEEE Trans. Reliability*, Dec. 1998, IEEE Comp. Soc. Press, 1998, pp.443-450.

[73] K. Burbeck, S. G. Andres, S. Nadjm-Tehrani, M. Semling, and T. Dagonnier, "Time as a metric for defence in survivable networks", *Proc. Work in Progress session of 24th IEEE Real-Time Systems Symposium (RTSS 2003)*, Dec. 2003.

[74] P. Verssimo, N. Neves, C. Cachin, J. Poritz, Y. Deswarte, D. Powell, R. Stroud, and I. Welch, "Intrusion-Tolerant Middleware: The Road to Automatic Security", *IEEE Security and Privacy*, vol. 4, no. 4, July-August 2006, pp.54-62.

Mohamed Al-Kuwaiti received his B.Sc. degree in Computer Engineering, and M.Sc. degree in Telecommunications and Computer Networks from The George Washington University, Washington D.C. in 1999, and 2001, respectively. He also received his D.Sc degree in Electrical Engineering majoring in Computer Networks and Network Reliability/Security from The George Washington University, Washington D.C. in May 2008. Since 2001, Dr. Al-Kuwaiti worked for The Embassy of The United Arab Emirates-The Military Attach Office in Washington, D.C., as an Administrative Attach where he was engaged in several computer and communication network design projects. In 2006, he became the Assistance of The UAE Military Attach at The UAE Embassy where he has been involved in several administrative and technical tasks. Dr. Al- Kuwaiti has also been extensively associated with The George Washington University, Washington, D.C., since 2000, where he has been a part of the Department of Electrical and Computer Engineering (ECE). He has been working on information networked systems dependability and survivability as part of his graduate thesis. His main research interests and activities include Reliable Networks and QoS where his focus is on finding measurable parameters for evaluating telecommunication network performance. He is also interested in Network Security problems in general.

Nicholas Kyriakopoulos received his B.E.E., M.S., and D.Sc. degrees in Electrical Engineering from The George Washington University, Washington, DC, in 1960, 1963, and 1968, respectively. In 1960, he became an Electronics Engineer at the Harry Diamond Laboratories, U. S. Department of the Army, Washington, DC, where he was engaged in the development of test circuits, and procedures for the determination of semiconductor device parameters. In 1964, he became an Instructor in the School of Engineering and Applied Science, at The George Washington University, Washington, DC, where he has been a Professor since 1980. Since 1979, he has also been affiliated with the U. S. Arms Control and Disarmament Agency, and subsequently with the U. S. Department of State as an Expert on monitoring systems for verifying arms control agreements, and on critical infrastructure protection. Dr. Kyriakopoulos has been teaching, conducting research, and publishing in the areas of systems and signal processing. His current interests and activities are in monitoring systems, digital signal processing, information systems dependability, and the application of technology to arms control. His latest publication as co-editor is Verifying Treaty Compliance: Limiting Weapons of Mass Destruction and Monitoring Kyoto Protocol Provisions (Heidelberg: Springer, 2006).

Sayed Hussein received his B.Sc, and M.Sc degrees in Electrical Engineering from Cairo, Egypt in 1976, and 1981 respectively; and the Ph.D. degree in Broadband Communication, Congestion Control in Broadband Multi-Source Packet Switching, from The George Washington University, Washington DC, in 1989. In 1990-1996, he joined The MTC in Cairo, Egypt teaching and researching in the area of LAN hybrid protocols, VOIP, and network survivability. In 1996-2004, he joined NTI research institute in Cairo, Egypt, conducting research in network performance, and network security. During 1990-2004, he worked in multiple national projects in computer information networks within the governmental sector. Since 2004, he joined the George Washington University, Washington, DC, as professional lecturer teaching network and security, research in QOS, high speed switching, bi-criteria routing techniques, and secure routing in Ad hoc network.