Measuring “ilities” is a Hopeless Task

Lu HAN
03-23-2006

Abstract

Our society is at a turning point in the history of technology adoption, computer systems are creeping into the very fabric of everyday life. However, there is no overall framework for measuring many elements of systems, much less combining them into a system-level metrics for comparison or prediction purposes. Therefore, researchers propose the grand challenge of quantifying and measurement the ilities. The attributes of the ilities are generally agreed to be manageability, maintainability, serviceability, dependability and diagnoseability.

We take the position that trying to quantity manageability, diagnoseability, serviceability, and maintainability in a general purpose way is a hopeless task. At best, any such quantification can be measured and applied only to a specific system with human factor studies. As arguments, we try to clarify the definition and characteristics of each “ility”, as well as the main challenges of quantifying these “ilities”. After analyzing the characteristics of the ilities as well as exploring the challenges of quantifying ‘ilities’, we draw a conclusion that it is impossible to quantity them in a general way, since all these ilities are the reflection of human effort and skill. The best we can do is using human factor studies to measure and apply these ilities to some specific systems. Even in this case, new and accurate metrics are needed to reflect human factor and measure ilities, and still many problems are inherently unsolvable. Although many works have been done to solve the existing problems about ilities, we will show in this paper that it is very difficult to solve these limitations which made measuring ‘ilities’ a hopeless task.

1. Introduction

To operate successfully, most large networks depend on software, hardware, and human operators and maintainers to function correctly. Failure of any one of these elements can disrupt or bring down an entire. “Ilities” are crucial aspects of modern systems. It has been acknowledged that the “ilities” includes manageability, maintainability, serviceability and diagnoseability, and they will play a crucial role in driving progress toward highly reliable, easily maintained system. Well-designed evaluations of these ilities will provide a yardstick for assessing the current state of the art. Most of the research has been done about ‘ilities’ today is focus on two main aspects, one is the computer system itself, which means its hardware and software, another aspects is a crucial determinant of system ilities: the human operator. Human operators play a vital role in system ilities, configuring systems at any time, detecting problems as they arise, diagnosing them, and repairing them to maintain all kinds of ilities. However, measuring the ilities and their associated metrics of availability, performance, and correctness becomes real challenges that directly affect users and the corporate bottom line of service providers. This reason is obvious, it is easy to evaluate the software and hardware functions, but there is no such a general method to evaluate human operators. To measure the ilities, the human factor study is unavoidable.

Furthermore, measuring ilities requires a radically different approach from what is used in measuring traditional performance based on human factor study. Measuring ilities demand novel methodologies and produce different categories of new metrics to reflect human behavior, since quantifying the ilities inherently involves human component, which made measuring ilities in a general purpose way a hopeless task and measuring the ilities have to base on the human factor study.

In section 2, we will discuss the characteristics of the ilities and what are the fundamental challenges on measure the ilities. Section 3 and section 4 describe our argument and the counter side argument, respectively. Section 5 gives a basic overview about the human factor study as well as the importance of considering human factor, especially the human error. Then in section 6, we talk in detail about the five significant ilities that we are studying and show why we have to consider human factor study when we trying to quantify them. Section 7 is the conclusion, in which we draw a conclusion that measuring ilities is hopeless unless we applied it in some particular system with human factor study.

2. Characteristics of 'ilities' and Challenges of Measuring ilities

The effort of measuring system behaviors can be ascend to 1860's, since then various measurement of the system are proposed. Using a single number to measure an aspect of the system is a traditional way of performance evaluation. However, unlike the traditional way of performance measurement, the ilities measurement of the existing systems is compared to what can be procured, in most cases without clear terminology or quantitative metrics for either. Performance can be measured by so many ways, like the system feed back time, feed through time, round trip time, throughput and latency, these metrics are largely depend on the system itself. The key challenge of measuring the ilities is the human factor, since all of the ilities are dealing with the human behaviors. It will be difficult to measure human behavior since there are so many variations.

The ilities of a system usually includes manageability, maintainability, diagnoseability serviceability and dependability. As described above, in order to measure ilities, we have to use other methodology instead of the general way. A good example is, when Aaron wants to benchmark the configuration complexity, he used a process-driven method instead of workload driven method.

The manageability of the system depends heavily on what characteristics of the services and networks need to be measured, how they are measured and how those data are used. Apparently, trying to measure the manageability needs to measure the human interaction.

The maintainability measures the effort required to modify a program once it is in production. Compared to performance, maintainability is hard to measure because it is a measurement of human's effort. There is no direct measure of maintainability. In order to better quantify the maintainability, human factor study is inevitable.

The serviceability is often mentioned with reliability and availability to achieve better system performance. Serviceability is also known as supportability, it refers to the ability of technical support personnel to debug or perform root cause analysis in pursuit of solving a problem with a product. From the definition, we can easily grab the key word – *personnel,* which clearly indicate that if we want to study evaluation the serviceability, and then the human factor study is the crucial part.

Diagnoseability is the ability of first detect the faults, isolate the faults and then recovery the faults. We cannot measure the system's diagnoseability with any of the existing methods and the metrics, because the diagnoseability includes the human faults and human recovery. The only way of trying to measure the diagnoseability is by human factor study.

As for dependability, it has many attributes, like reliability, availability, safety and security. We can easily conclude that measurement of the dependability is also depending on the human factor study.

Although all these ilities are inherently base on the human operator, unfortunately, human operators rarely perform these tasks perfectly, and often end up being the primary source of system failures and unavailability. Thus, until we can completely eliminate the human by building fully self-maintaining, self-administering systems (a goal which remains far away on the horizon), performance of human operators will remain a critical component of system dependability. We believe that measuring abilities will have to take the behavior of human operators into account, both in the design of high-dependability systems and in evaluating for ilities.

3. The Argument

It is debated in academic as whether measure ilities in a general method is a hopeless task. The reason for debate is that ilities are dealing with human behavior, and in order to measure ilities, we have to depend on human factor. If we could prove that, technically, that trying to measure ilities is exactly trying to measure the amount of human effort and skill, we could say that measuring the ilities in a general way is hopeless.

This paper takes the position that trying to quantity manageability, diagnoseability, serviceability, and maintainability in a general purpose way is hopeless. At best, any such quantification can be measured and applied only to a specific system with human factor studies. Motivated by the observation that a system's ilities are significantly influenced by the behavior of its human operators, we argue that if we want to measure the ilities we have to captures the impact of the human system operator on the system under test. What we can do now is all base on the human factor study, which requires human intervention. Some researcher gave this methodology a name, "Macro-benchmarks".

4. The Counter Argument

This argument claims that trying to quantity manageability, diagnoseability, serviceability, and maintainability in a general purpose way is not a hopeless task, although it will be a little difficult than measuring the performance metrics.

Since the method of measure the performance metrics is already usable, they just propose their method by using the existing performance measurement method, or some methods derived from existing method to qualify the ilities.

For maintainability, some scientists claim that metrics for maintainability are really metrics measuring the complexity of the system. They assume that the more complex systems the harder to maintain. Then they give some indications for the measuring system complexity; include coupling, cohesion and the inheritance depth. Coupling means that systems with high coupling are difficult to modify. Cohesion means the components with clear and singular purposes localize changes to a smaller number of components. And the Inheritance depth means the average depth of inheritance is potentially an indicator of design complexity.

The measure of the manageability of a system that results from an analysis of the collection of network elements such as computers, modems, protocols, gateways, and applications that can and are to be managed. The measure is construed so that it is proportional to management effort (that is, the more effort required in managing a network, the higher will be the value of the measure).

There are also a number of other kinds of methods, like a number of researchers proposed the soft computing techniques. Soft computing is a synergistic integration of three computing paradigms viz., neural network, fuzzy logic and probabilistic reasoning. Fuzzy models have been used to estimate effort, size, and maintainability and understand ability. Neural network models have also been used to estimate size, effort and understanding maintainability.

Some other scientists try to use the existing metrics to evaluate serviceability. They use MTTR, MNTR and MTTBS to be the common metrics for evaluating serviceability. They claim that calculation of these metrics on an overall system basis as well as per failure type basis is useful towards quantitative understanding of the practical impact of each failure type. The methods they used are as following:

1. Mean Time To Repair: This calculation is intended to reflect the average amount of time it takes to recover from a failure.

$$MTTR = \frac{unscheduled\ \ downtime}{number\ \ of\ \ failures}$$

2. Mean Node hours To Repair This calculation measures the average computational ability lost per failure.

$$MNTR = \frac{unscheduled\ \ downtime\ node\ hours}{number\ \ of\ \ failures}$$

3. Mean time to Boot System Wallclock time to boot the complete system is a useful metric, whose importance increases with the number of times the system must be booted (e.g. the number of system failure events requiring a system reboot).

$$MTTBsystem = \frac{sum\ of\ \ wallclock\ \ time\ booting\ \ the\ \ system}{number\ \ of\ \ boot\ \ events}$$

The pioneering efforts and the related initiative models being developed are e.g., the IFIP WG 10.4 SIGDeB and the European IST project DBench.

However, we do not agree with the above statements.

Our point of view is that when we talk about measuring the manageability, maintainability, serviceability and diagnoseablity, we think it is a hopeless task to measure it in the general way, because first, we are talking about overall quantitative understanding of the system and accurate recording of the system's behavior. We want a design and deployment culture for measuring ilities to attain perfection, at least very close approximation, while the approach given above is just like some aspects of the ilities instead of the whole scenarios. Second, even they tried to use as simple metrics as they could and try to avoid human factor, however, what they did is still involving human factors, e.g., MTTR, it is still involving human operator to repair the system, and neural network, it is also a area based on human behavior study. The results show that trying to quantity manageability, diagnoseability, serviceability, and maintainability in a general purpose way is hopeless. At best, any such quantification can be measured and applied only to a specific system with human factor studies.

5. Human Factor Study

One of the most significant of these factors is human behavior. A system's human operators exert a substantial influence on that system's ilities: they can increase ilities via their monitoring, diagnosis, and problem-solving abilities, but they can also decrease ilities by making operational errors during system maintenance. The human error factor is particularly important to ilities.

Anecdotal data from many sources has suggested that human error on the part of system operators accounts for roughly half of all outages in production server environments. Recent survey confirms the significance of human error as a primary contributor to system failures. Human error is the largest single failure source, here are some data, in 2001, human error is the number one cause of failures in HP HA lab; in 1999, and half of the DB failures in Oracle are due to human error. Although human errors are unavoidable, the good aspect is human can self-detect error. About 75% of errors are immediately detected.

Basically, there are two kinds of human errors, one is slips/lapses error, which is the errors in execution, and other error kind is mistakes error, in planning time. The reason why we have to do human factor study also includes that automation does not cure human error, because automation shifts some error from operator errors to design errors which is much harder to detect, furthermore, automation can only address the easy tasks, leaving the complex, unfamiliar task for human, while humans are ill-suited to these tasks, especially under stress.

There are more human operators performance tasks on systems, such as system initiation, backups and restores, software upgrades, system reconfiguration, and data migration, and the ilities impact of these tasks must be measured as well. The researchers give out two choices on how to incorporate human-induced perturbations into ilities measurement. One option is to use a model of human operator behavior to perturb the system during quantifying. While this approach provides reproducibility and has the advantage of not requiring human participation, it unfortunately reduces to an unsolved problem—if we were able to accurately simulate human operator behavior, we would not need human system operators in the first place! The alternate approach is including human operators in the quantifying process. Doing this also raises several challenges, notably how to deal with human variability, how to perform valid cross-system comparisons with different operators, and how to structure benchmark trials so that the number of human operators is minimized. Defining the human operator workload requires selecting a set of maintenance tasks for the operator to perform during the benchmark; these tasks must be representative of the types of maintenance performed in real-world production installations.

The inherent variability and unpredictability of human actions make it a challenge to achieve the quantification of these ilities when we start including human in the measuring process. Thus a crucial part of the methodology to quantify the ilities is to manage the variability in the human operators, the different level of the human operators (expert, new-trained and novice), which is still having grand challenges.

Some scholars bring out the learning curve theory. The learning curve effect refers to the fact that operators learn something about the target system as they perform the benchmark, so that by the end of the benchmark they have more experience and better skills than when they started. Subsequent benchmark runs will show the effect of this learning, making it difficult to compare different runs directly. The learning curve effect would be irrelevant if we could assume that a fresh set of operators was available for each benchmark run, but in practice it is almost always necessary to reuse operators.

Several other approaches were also given by other scholars, like human time cost, human subjects and experiments. Some comments about them are: “the work is fundamentally flawed by its lack of consideration of the basic rules of the statistical studies involving humans...meaningful studies contain hundreds if not thousands of subjects”; “The real problem is that, at least in the research community, manageability isn't valued, not that it isn't quantifiable”. Obviously, this again proves our argument that ilities quantification can be measured and applied only to a specific system even with human factor studies.

6. Ilities

Within systems engineering, ilities are aspects or non-functional requirements. They are so – named because most of them end in “-ility”. Normally a subset of them will form another criterion to evaluate systems. For example, a subset of reliability, availability, serviceability, usability and installability are together referred to as RASUI, and reliability, availability, salability and recoverability are together to form the RASR which is an important concept of database. The “ilities” family includes more than 40 ilities. We will focus on four of them: manageability, maintainability, serviceability, diagnoseability and dependability.

A system’s overall manageability, maintainability, serviceability, diagnoseablity and dependability cannot be universally characterized with a single number as system performance (e.g., latency and throughput) because there is too much variation in capabilities, usage patterns, and administrator demands and training, etc.

In this section, after giving the definition of each ility, we will analyze their inherent characteristics which made them difficult to be measured, as well as discuss why try to quantify these ilities is a grand challenge.

6.1. Manageability

The manageability means that manages the system to ensure the continued health of a system with respect to performance, scalability, reliability, availability and security..

In order to measure the manageability, we have to consider the concurrency, load, load characteristics (e.g., latency, throughput) of future services, as well as the extent and nature of their relationships with infrastructure functions, complicate correlation functions, status reporting frequencies, performance tuning, and the proper interpretation of signal characteristics. All of these uncertainties make it especially difficult to quantify the manageability; it is also become a big barrier to development of future automated infrastructure management capabilities.

Most of these aspects are the human behavior related, especially manageability is to measure the ability of human administrator’s management to the system, and so, we can draw a conclusion that if we want to quantify the manageability, the human factor study has to been included in measuring the manageability.

6.2. Maintainability

Maintainability is the characteristic of design and installation which inherently provides for an item to be retained in, or restored to a specified condition within a given period of time, when the maintenance is performed in accordance with prescribed procedures and resources. In other words, it is the ease and speed with which any maintenance activity can be carried out on an item of system, the ability to undergo repairs and evolution. Systems should require only minimal ongoing human administration regardless of scale or complexity. There are many sub-characteristics of maintainability, like analyzability, changeability, stability and testability.

Maintainability are always discussed with reliability and availability (RAM), they are the key goals for modern systems. Both of the system maintainability, reliability and availability are pressing problems. In order to achieve these attributes, some researchers agreed with a methodology, they start by understanding the existed systems, then figure out how to measure them, evaluate existing systems and techniques, develop new approached based on what they’re learned and measure them as well.

Improved reliability, availability and maintainability metrics have been developed that account for customer perceived factors such as frequency of outage, duration of outages, business impact of outages, etc. In various realizations and exploitations, such improved metrics may be utilized for managing and/or monitoring availability of enterprise information services or suites, availability of individual computers, devices or facilities, and/or availability of particular functionality or subsystems of any of the above. In one exploitation, personnel management decisions and/or compensation levels may be based on achieved values for such improved metrics. In other exploitations, contractual commitments and/or incentive fees related to an installed system or systems maybe based on such improved metrics.

There is a simple example of the early attempt of the measurement of the maintainability. This method uses 5-person pilot study to achieve the goal, which is a prototype of measuring maintainability; it is also a typical example of human factor study when trying to evaluate the maintainability.

However, until now, we don’t know how to build highly-maintainable systems except at the very high-end, because there’s few tools exist to provide insight into system maintainability, neither do we have method to measure them. Most existing benchmarks ignore the human factor study. These benchmarks are mostly focus on performance and under ideal conditions. Furthermore, there is no comprehensive, well-defined metrics for maintainability.

From the above argument, it is obviously to obtain that the system’s overall maintainability cannot be universally characterized with a single number, because there is too much variation in capabilities, usage patterns, administrator demands and training, etc. General method of measuring the maintainability is by no mean reasonable, which again support our argument that at best, the evaluation of maintainability is based on human factor study.

6.3 Serviceability

Serviceability is the characteristics of a product to make it more readily serviceable. These characteristics would address features related to fault identification, diagnosis, disassembly, repair, replacement, and re-assembly. Management of a mission-critical production computing environment is all about the goals of Reliability, Availability, and Serviceability (RAS). These have been the benchmarks for evaluating system management practices since the mainframe environments of the 1960s.

In evaluating serviceability, one is often dealing with users’ perceptions, and is thus dependent to a great extent upon subjective judgments. It is therefore important to continually monitor what the users want and need from a system. Some researcher said that a good measure of serviceability is the probability that any one of the users will find the services provided by the system to be satisfactory. This parameter can only be estimated by human factor study. The quest for predictability is fought on three fronts; Standards, Processes and Technology. With Standards, Researchers are seeking to improve predictability through consistency in order to reduce downtime due to trouble-shooting and entropy. With Processes, they seek to improve predictability by ensuring that system maintenance activities follow known paths which include quality assurance steps such as peer review, impact analysis and deployment planning. With Technology, they seek to improve the manner in which we manage our environment. So, where RAS are the measures, SPT are the strategies by which we seek to improve those measures.

It is not difficult to see that both the users’ perceptions, the degree of the user satisfactory and the SPT are closely related to the human behavior, which means the serviceability is another ility that depends on human factor study. It is essential to do human factor study if people want to quantify the serviceability.

6.4. Diagnoseability

Diagnoseability is the ability to automatically read the current state of a system and controls so are to detect and diagnose the cause of defects, and subsequently correct operational defects quickly. It is the ability to identify any faults or potential faults in the operation of the product. Diagnoseability includes the ability of guessing as to what is wrong with a malfunctioning circuit, narrows the search for physical root cause, makes inferences based on observed behavior and usually based on the logical operation of the circuit.

Any good yield-oriented defect strategy must have two main components-(a) the ability to perform rapid defect diagnosis for yield learning, and (b) the ability to efficiently extract defect parameters from the manufacturing line.

As we have stated in Section 5, human error is the significant reason for the system failure. Unlike the other ilities, diagnoseability has the inherently characteristics of dealing with human faults, the biggest challenge for system performance. Of course, it is also a measure of hardware and software faults. It is impossible to measuring the diagnoseablity without taking into account of human factor study.

6.5. Dependability

Dependability is the trustworthiness of a computer system such that reliance can justifiably be placed on the service it delivers. Dependability is normally described by a set of dependability attributes. These attributes include reliability, availability, safety, security and robustness. Reliability measures continuous correct service delivery, dependability with respect to continuity of service. Availability measures correct service delivery with respect to the alternation of correct and incorrect service, dependability with respect to readiness for usage. Safety measures continuous delivery of either correct service or incorrect service after benign failure, dependability with respect to the non-occurrence of catastrophic failures. Security is dependability with respect to the prevention of unauthorized access and /or handling of information. Robustness is the degree to which a system or component can function correctly in the presence of invalid input or stressful environment conditions.

They claimed that dependability benchmarks make these tasks possible. What is dependability benchmarking? Dependability benchmarking is performing a set of tests to quantify computer dependability. This quantification is supported by the evaluation of dependability attributes such as reliability, availability and safety, through the assessment of direct measures related to the behavior of a computer in the presence of faults. Examples of these direct measures are failure modes, error detection coverage, error latency, diagnosis efficiency, recovery time, and recovery losses.

The goal of dependability benchmarking is to quantify the dependability features of a computer or a computer component in a truthful and reproducible way. Unlike functionality and performance features, which are normally available to the customers or can be certified or measured, system dependability cannot be easily assessed today. It is still because the human part.

In this section, we give the definition as well as state the characteristics of the ilities one by one. We conclude that, all the ilities are not easy to quantify because they all inherently involve human performance, which is a grand challenge to measure even to benchmark these ilities.

7. Conclusion

The question whether measuring ilities is a hopeless task has been bought out in this position paper. We take the argument that trying to quantify the manageability, maintainability, serviceability and diagnoseability is the hopeless task using a general method. At best, any such quantification can be measured and applied only to a specific system with human factor studies. In section 6, we have proved that if we want to quantify the ilities we have to consider human factor study, by analyzing the inherent characteristics of manageability, maintainability, serviceability, diagnoseability, and dependability, as well as the exploring the challenges of evaluating them. Also, in section 5, we argue that human factor study is currently immature technique, which can only be applied on specific system. Thus, we proved our argument that trying to evaluate the ilities is a hopeless task.

We would argue that both ilities areas are still far from mature compared to performance benchmarks, there are lots of work and new methodologies and metrics must be developed. Furthermore, better human studies technology has to be employed.

All the above discussion should convince the reader of the fact that trying to quantify the ilities is a hopeless task currently.

Acknowledgement

I am grateful to Professor Richard P. Martin for his valuable advice on this paper.

References

[1] A. Brown. Towards Availability and

Maintainability Benchmarks: A Case Study of Software RAID Systems. *UC Berkeley Computer Science Division Technical Report UCB//CSD-01-1132*, Berkeley, CA, January 2001.

[2] A. Brown. and David A. Patterson. To Error is Human.

[3] T.C. Nicholas Graham. Quality Attribute-Based Architectural Analysis. *CISC 322: Software Architecture*

[4] Aaron B. Brown, Leonard C. Chung and David A. Patterson. Capturing the Human Component of Dependability in a Dependability Benchmark. *2000 DNS Workshop on Dependability Benchmarking*

[5] James C. Huy, Sumedh Mungee, and Douglas C. Schmidt. Techniques for Developing and Measuring High-Performance Web Servers over ATM Networks", *INFOCOM '98*

[6] K.K. Aggarwal, Yogesh Sigh, Pravin Chandra, and Manimala Puri. Sensitivity Analysis of Fuzzy and Neural Network Models. *ACM SIGSOFT Software Engineering Notes, July 2005, Volume 30 Number 4*

[7] Jon Stearley. Defining and Measuring Supercomputer Reliability, Availability, and Serviceability (RAS). *Proceeding of the 6th LCI International Conference on Linux Cluster 2005*

[8] Jens Gustavsson and Magnus Osterlund. Requirements on Maintainability of Software Systems – An Investigation of the State of the Practice.

[9] Fred Moavenzadeh. Large Scale Infrastructure Systems. *Working Paper Series ESD-WP-2003-01.24-ESD Internal Symposium*

[10] Madeira, H. and P. Koopman. "Dependability Benchmarking: making choices in an n-dimensional problem space." *Proceedings of the First Workshop on Evaluating and Architecting System dependabilitY (EASY '01), Göteborg, Sweden, July 2001.*

[11] Oppenheimer, D. and D. A. Patterson. "Architecture, operation, and dependability of large-scale Internet services." *IEEE Internet Computing, February, 2002.*

[12] Saito, Y., B. Bershad, and H. Levy. "Manageability, Availability, and Performance in Porcupine: A Highly Scalable Internet Mail Service." *Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP'99),*

[13] Adrian Wong, Leonid Oliker, William Dramer, Teresa Kaltz, and David Bailey. SEP: A system utilization benchmark. *In Proceedings of Supercomputing 2000 Conference, 2000*

[14] Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Ilities

[15] Kiran Nagaraja, Xiaoyan Li, Ricardo Bianchini, Richard P. Martin and Thu D. Nguyen. Using Fault Injection and Modeling to Evaluate the Performability of Cluster – Based Services.

[16] A. Avizienis, J. –C. Laprie and B. Randell: Fundamental Concept of Dependability. *Research Report No 1145, LAAS-CNRS, April 2001*