From: *Real-Time Mission-Critical Systems: Grand Challenge Problems*, November 30, 1999; Phoenix, Arizona USA

Dependability Benchmarking & Prediction: A Grand Challenge Technology Problem

Philip Koopman
ECE Department & ICES
Carnegie Mellon University
Pittsburgh, PA, USA
koopman@cmu.edu

Henrique Madeira
Information Engineering Department
Universidade de Coimbra
Coimbra, PORTUGAL
henrique@dei.uc.pt

Abstract

We propose the grand challenge problem of dependability benchmarking and prediction for real-time mission-critical systems (RTMCSs). Evaluating dependability would quantify the degree of reliance that could justifiably be placed on a critical system, even in the face of partial failures or exceptional conditions. A comprehensive result would require an inter-disciplinary approach embracing the entire product lifecycle. While there are significant technical hurdles to both assessing the dependability of individual elements and combining resultant measures, a viable approach must be found to ensure that the computing systems our society is coming to depend upon will be reliable, available, safe, and secure. The participation of several communities, including the Real Time Computing community, is vital to successfully address this challenge.

1. The challenge: quantifying dependability

Our society is at a turning point in the history of technology adoption. Up to now, most computer applications have been non-critical, and merely provided improved convenience or efficiency. But now, computer systems are creeping into the very fabric of everyday life. We are in the process of seeing real-time mission-critical systems (RTMCSs) changing from being few, expensive, and carefully regulated to being numerous, inexpensive, and loosely regulated. The technical community may not desire such a change, because it is not known how to assure the dependability of huge numbers of systems manufactured with stringent cost controls constraints and few or no governmental certification requirements. But, eager adoption of less-than-dependable technology in effectively critical roles will happen regardless of the opinion of researchers (and, in fact, is already happening as regular readers of the comp.risks Internet newsgroup can attest).

Even though it is obvious that RTMCSs must be dependable, there is no general way to quantify or characterize the overall dependability of a newly designed system. While some elements of such an approach are available (*e.g.*, component-based hardware reliability calculations), there is no overall framework for measuring many elements of dependability, much less combining them into a system-level metric for comparison or prediction purposes.

Therefore, we propose the grand challenge of dependability benchmarking and prediction. The challenge is two-fold. The first goal is to be able to **compare the dependability of different systems**, both similar and dissimilar, to for the purpose of evaluatinge relative strengths and weaknesses. This will enable the assessment of the relative merits of different architectures, alternate design ap-

Definitions:

- **Dependability:** Trustworthiness of a computer system such that reliance can justifiably be placed on the service it delivers.
- **Reliability:** Measure of continuous correct service delivery (dependability with respect to continuity of service).
- **Availability:** Measure of correct service delivery with respect to the alternation of correct and incorrect service (dependability with respect to readiness for usage).
- **Safety:** Measure of continuous delivery of either correct service or incorrect service after benign failure (dependability with respect to the non-occurrence of catastrophic failures).
- **Security:** Dependability with respect to the prevention of unauthorized access and/or handling of information.
- **Robustness:** The degree to which a system or component can function correctly in the presence of invalid inputs or stressful environment conditions. [IEEE90]

Precise use of terminology remains an ongoing debate among different communities; the above is drawn from [Laprie92] except as noted.

proaches, and proposed methods to improve dependability. Because comparing systems might be done with relative rather than absolute measures, the second goal is to be able to **predict the field dependability of a system** in a quantified way before it is actually deployed.

2. Problem Scope

The issue of dependability of an RTMCS is multi-faceted. The attributes of dependability are generally agreed to be reliability, availability, safety, and security (see box on previous page for terminology definitions). Within these attributes, an RTMCS must ensure not only correctness, but also appropriate timeliness of its results. Beyond that, dependability further deals with whether the system delivers some level of acceptable service under adverse conditions, or at least fails safely rather than failing in a dangerous manner.

It is important to note that the context for dependability is not merely a theoretical design measured against a (possibly imperfect) specification. Instead, to be useful, notions of dependability must encompass the messiness of the real world. Thus, dependability also includes robustness, including operation in situations that are unspecified, exceptional, the result of partial system failure, or even the result of malicious attacks. Even in adverse situations, a dependable system must maintain a reasonable level of correctness and timeliness.

Given the different aspects of dependability that must be considered, it is also important to realize that there are multiple different technical areas within systems that must be assessed when examining dependability. These include multiple areas within three different dimensions of system design:

- Implementation technology: hardware, software, control algorithms, user interface, mechanical safety backups
- Operational life cycle: specification, design, deployment, maintenance, operation, disposal
- Product deployment scale: capital equipment, consumer products, disposable goods

In each of these three dimensions, dependability considerations manifest in different ways, and must be measured within somewhat differing contexts.

While it would be no surprise if early attempts had carefully set, but modest, goals, the ultimate scope of our vision is to find ways to create highly dependable systems that can be deployed in huge numbers for use by everyday people at an affordable price. This is essential to supporting a safe, orderly transition from current RTMCSs that are in small-scale production to the widespread commodity RTMCSs that are inevitable in the future. This grand challenge proposes creating the most basic scientific ingredient needed for such a change – the ability to measure the desired property of dependability. Then, the challenge proposes taking the next logical step of achieving standard, repeatable, scalable, easy to use, and generally accepted ways of measuring and comparing dependability properties, in the form of benchmarks.

3. Elements of an Approach

IFIP Working Group 10.4 has created a Special Interest Group (SIG) to establish a framework for dependability benchmarking (the authors of this paper are the chair and co-chair of this SIG). The benefits of such a framework would be a clear understanding and articulation of the fundamental reasons for undependability across multiple disciplines, a perspective on available tools and techniques for measuring/predicting dependability, and an enumeration of the fundamental issues that make this a grand challenge problem area.

A preliminary vision of the Dependability Benchmarking SIG is to create a dependability benchmark, which might be defined as: **a test suite to measure the behavior of a computer system in the presence of faults (e.g., failure modes, error detection coverage, error latency, diagnosis efficiency, recovery time, recovery losses), supporting the evaluation of dependability attributes (reliability, availability, safety, security)**. This approach would increase the emphasis of using direct dependability metrics for focus on the evaluation of system designs, as a way to augment using direct dependability metrics rather than existing approaches based on retrospective field data studies andor indirect metrics based on design attributes such as complexity measures.

There would probably be many elements to a dependability benchmark suite, including:

- **Specifications** of expected system behavior in different fault situations, including in all likelihood an approach to specifying graceful degradation properties.
- **Measures** based on instrumentation that summarize and create a quantified evaluation of a system under test.
- A **workload**, used to create a reasonable operating scenario for testing.
- A **faultload**, used to inject system faults, exceptional situations, component overloads, operator mistakes, maintenance errors, component failures, and other events that could lead to undependability if not properly handled by the system.
- **Instrumentation** to record the workload, faultload, and performance of the system, including levels of degradation or failures of various operating components as well as overall system performance.

- **Procedures and rules** for benchmarking activities. It is well known that any benchmark can be "gamed" to produce optimistic results. A dependability benchmark would have to include standards for conducting measurement to ensure uniform conditions for measurement. In addition to the obvious items such as system configuration disclosures for performance metrics, dependability metrics might also include requirements or disclosures involving everything from maintenance procedures to operator training, considering all factors that affect dependability.

The results of the possible measures bear specific discussion. In general they are not likely (at least at first) to be a number so tidy as estimated Mean Time To Failure (MTTF). More likely, the results will be in the form of a set of values that characterize system behavior. Possible metrics include error detection coverage, error detection latency, error location (diagnosis) effectiveness, error location latency, recovery time, system state losses after recovery, and degree of degradation as a function of fault load. It is important to note that the resultant metrics should give a clear indication of dependability at the system level, not merely at the component level; for it is the system dependability which ultimately matters to users.

4. Technical Hurdles

The previous design and deployment culture for RTMCSs has been to attain perfection (or a very close approximation thereof), and thereby avoid the need to quantify dependability in advance of deployment beyond traditional component-failure-based reliability and availability calculations. This thinking is common in both military and commercial systems. However, the approach of expending a huge amount of resources to ensure near-perfection doesn't scale from current systems to commodity product RTMCS domains. First, no real system is free of design defects, nor will one ever be in the foreseeable future. Thus, when multiplied by a large number of deployed units, any RTMCS can be expected to fail in usage even when designed and manufactured with best-known practices. Second, the reality of limited resources and limited development/testing time results in causes there to be defects remaining even in even carefully designed, low-volume mission critical systems. Third, consumer product development time and resource budgets cannot support even the limited techniques used on traditional RTMCSs, boding poorly for the future. Thus, creating new development approaches and training designers hanging development team mindset to deal with the realities of new domains rather than traditional RTMCS domains might well be the first (non-technical) hurdle to achieving highly dependable designs.

There are, of course, technical hurdles as well. The state of the art in dependability measurement varies considerably across different technology areas. Hardware reliability measurement is mature for failures stemming from physical defects. However, both hardware and software design dependability are essentially unquantifiable at the present time (techniques exist for estimating defect rates of released software, but these are primarily aimed at establishing correctness rather than dependability, despite the term “software reliability” being used for this area). User interface dependability has been studied in the areas of human factors (now known as Human/Computer Interfaces), but is not yet a mature science. And, while mechanical safety dependability is understood from a hardware reliability point of view, issues such as mechanical/software safety tradeoffs are largely an unexplored area. Finally, the sub-issue of assuring security is extremely difficult (and is arguably a grand challenge problem in its own right). In terms of tool support, there are in fact many different measurement tools, and in particular a variety of fault injection tools that are the product of a decade of research. However, the tools are for the most part niche-oriented, and there is no generally agreed upon framework for fault injection experimentation and interpretation of measurements.

It is possible that setting a goal of an outright dependability benchmark in the usual sense is overly ambitious. However, attempting to make progress in this direction could also bring to light viable alternatives for the nearer term. Possible alternatives include documenting best practices, measuring the effectivness of design processes, certification-based approaches of either processes or systems, and using measurements of low-level system properties to estimate overall system dependability.

5. Potential Impact

What if instead of vainly hoping that a system was perfect, one could instead know just how dependable (or undependable) it was going to be before it was deployed? While this prospect might raise some interesting (and critical) legal and ethical issues, consumer-oriented RTMCSs are unlikely to be so dependable that they can be considered essentially perfect. And, even if they were as near-perfect as the best current RTMCSs, the huge number of deployed units would mean that, for example, what were “improbable” events in an aircraft fleet would be everyday events in an automotive fleet orders of magnitude larger. Quite likely, many systems will be far less dependable than that. Thus, quantifying dependability is not simply a nicety for future RTMCSs – it is a vital necessity. Performing this feat will require close cooperation among the various technical groups that deal with RTMCSs, including the fault tolerance community, the real time system community, and the mission-critical software community.

Any RTMCS that is developed on a tight budget, that does not admit to best-possible software quality assurance practices, that cannot afford stringently screened components, that uses off-the-shelf technology instead of special safety-critical components, that is deployed in huge volume (hundreds of millions of units instead of hundreds of units), or that is used by consumers who are not professionally trained operators is a potential beneficiary of dependability benchmarking. Those who have much to gain include not only end-users of systems, but also developers who wish to cost-justify increasing the dependability of components they produce, and system integrators who wish to understand the dependability of off-the-shelf components they acquire as well as the dependability of the finished systems they create.

6. Acknowledgements

The authors wish to thank Jean Arlat for his insightful comments. Funding for this paper was provided, in part, by DARPA under contract DABT 63-96-C-0064, and by the Portuguese Government/European Union through R&D Unit 326/94 (CISUC).

7. References

[IEEE 90] Institute of Electrical and Electronics Engineers. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*, 1990.

[Laprie92] J.-C. Laprie (Ed.) *Dependability: Basic Concepts and Terminology in English, French, German, Italian and Japanese*, Springer-Verlag, 1992.