FOUNDATIONS OF DEPENDABLE COMPUTING
Models and Frameworks for Dependable Systems

edited by

Gary M. Koob
Clifford G. Lau

FOUNDATIONS OF
DEPENDABLE COMPUTING
Models and Frameworks for Dependable Systems

edited by

Gary M. Koob
Clifford G. Lau
Office of Naval Research

FOUNDATIONS OF
DEPENDABLE COMPUTING
*Models and Frameworks for
Dependable Systems*

THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE

OFFICE OF NAVAL RESEARCH
Advanced Book Series
Consulting Editor
André M. van Tilborg

Other titles in the series:

PREFACE

Dependability has long been a central concern in the design of space-based and military systems, where survivability for the prescribed mission duration is an essential requirement, and is becoming an increasingly important attribute of government and commercial systems where reduced availability may have severe financial consequences or even lead to loss of life. Historically, research in the field of dependable computing has focused on the theory and techniques for preventing hardware and environmentally induced faults through increasing the intrinsic reliability of components and systems (fault avoidance), or surviving such faults through massive redundancy at the hardware level (fault tolerance).

Recent advances in hardware, software, and measurement technology coupled with new insights into the nature, scope, and fundamental principles of dependable computing, however, contributed to the creation of a challenging new research agenda in the late eighties aimed at dramatically increasing the power, effectiveness, and efficiency of approaches to ensuring dependability in critical systems

At the core of this new agenda was a paradigm shift spurred by the recognition that dependability is fundamentally an attribute of applications and services—not platforms. Research should therefore focus on (1) developing a scientific understanding of the manifestations of faults at the application level in terms of their ultimate impact on the correctness and survivability of the application; (2) innovative, application-sensitive approaches to detecting and mitigating this impact; and (3) hierarchical system support for these new approaches.

Such a paradigm shift necessarily entailed a concomitant shift in emphasis away from inefficient, inflexible, hardware-based approaches toward higher level, more efficient and flexible software-based solutions. Consequently, the role of hardware-based mechanisms was redefined to that of providing and implementing the abstractions required to support the higher level software-based mechanisms in an integrated, hierarchical approach to ultradependable system design. This shift was furthermore compatible with an expanded view of "dependability," which had evolved to mean "the ability of the system to deliver the specified (or expected) service." Such a definition encompasses not only survival of traditional single hardware faults and environmental disturbances but more complex and less-well understood phenomena, as well: Byzantine faults, correlated errors, timing faults, software design and process interaction errors, and—most significantly—the unique issues encountered in real-

time systems in which faults and transient overload conditions must be detected and handled under hard deadline and resource constraints.

As sources of service disruption multiplied and focus shifted to their ultimate effects, traditional frameworks for reasoning about dependability had to be rethought. The classical fault/error/failure model, in which underlying anomalies *(faults)* give rise to incorrect values *(errors)*, which may ultimately cause incorrect behavior at the output *(failures),* required extension to capture timing and performance issues. Graceful degradation, a long standing principle codifying performance/dependability trade-offs must be more carefully applied in real-time systems, where individual task requirements supercede general throughput optimization in any assessment. Indeed, *embedded* real-time systems—often characterized by interaction with physical sensors and actuators—may possess an inherent ability to tolerate brief periods of incorrect interaction, either in the values exchanged or the timing of those exchanges. Thus, a technical failure of the embedded *computer* does not necessarily imply a *system* failure. The challenge of capturing and modeling dependability for such potentially complex requirements is matched by the challenge of successfully exploiting them to devise more intelligent and efficient—as well as more complete—dependability mechanisms.

The evolution to a hierarchical, software-dominated approach would not have been possible without several enabling advances in hardware and software technology over the past decade:

(1) Advances in VLSI technology and RISC architectures have produced components with more chip real estate available for incorporation of efficient concurrent error detection mechanisms and more on-chip resources permitting software management of fine-grain redundancy;

(2) The emergence of practical parallel and distributed computing platforms possessing inherent coarse-grain redundancy of processing and communications resources—also amenable to efficient software-based management by either the system or the application;

(3) Advances in algorithms and languages for parallel and distributed computing leading to new insights in and paradigms for problem decomposition, module encapsulation, and module interaction, potentially exploitable in refining redundancy requirements and isolating faults;

(4) Advances in distributed operating systems allowing more efficient interprocess communication and more intelligent resource management;

(5) Advances in compiler technology that permit efficient, automatic instrumentation or restructuring of application code, program decomposition, and coarse and fine-grain resource management; and

(6) The emergence of fault-injection technology for conducting controlled experiments to determine the system and application-level manifestations of faults and evaluating the effectiveness or performance of fault-tolerance methods.

In response to this challenging, new vision for dependable computing research, the advent of the technological opportunities for realizing it, and its potential for addressing critical dependability needs of Naval, Defense, and commercial systems, the Office of Naval Research launched a five-year basic research initiative in 1990 in *Ultradependable Multicomputers and Electronic Systems* to accelerate and integrate progress in this important discipline. The objective of the initiative is to establish the fundamental principles as well as practical approaches for efficiently incorporating dependability into critical applications running on modern platforms. More specifically, the initiative sought increased effectiveness and efficiency through (1) Intelligent exploitation of the inherent redundancy available in modern parallel and distributed computers and VLSI components; (2) More precise characterization of the sources and manifestations of errors; (3) Exploitation of application semantics at all levels—code, task, algorithm, and domain—to allow optimization of fault-tolerance mechanisms to both application requirements and resource limitations; (4) Hierarchical, integrated software/hardware approaches; and (5) Development of scientific methods for evaluating and comparing candidate approaches.

Implementation of this broad mandate as a coherent research program necessitated focusing on a small cross-section of promising application-sensitive paradigms (including language, algorithm, and coordination-based approaches), their required hardware, compiler, and system support, and a few selected modeling and evaluation projects. In scope, the initiative emphasizes dependability primarily with respect to an expanded class of hardware and environment (both physical and operational) faults. Many of the efforts furthermore explicitly address issues of dependability unique to the domain of embedded real-time systems.

The success of the initiative and the significance of the research is demonstrated by the ongoing associations that many of our principal investigators have forged with a variety of military, Government, and commercial projects whose critical needs are leading to the rapid assimilation of concepts, approaches, and expertise arising from this initiative. Activities influenced to date include the FAA's Advanced Automation System for air traffic control, the Navy's AX project and Next Generation Computing Resources standards program, the Air Force's Center for Dependable Systems, the OSF/1 project, the space station Freedom, the Strategic

Defense Initiative, and research projects at GE, DEC, Tandem, the Naval Surface Warfare Center, and MITRE Corporation.

This book series is a compendium of papers summarizing the major results and accomplishments attained under the auspices of the ONR initiative in its first three years. Rather than providing a comprehensive text on dependable computing, the series is intended to capture the breadth, depth, and impact of recent advances in the field, as reflected through the specific research efforts represented, in the context of the vision articulated here. Each chapter does, however, incorporate appropriate background material and references. In view of the increasing importance and pervasiveness of real-time concerns in critical systems that impact our daily lives—ranging from multimedia communications to manufacturing to medical instrumentation—the real-time material is woven throughout the series rather than isolated in a single section or volume.

The series is partitioned into three volumes, corresponding to the three principal avenues of research identified at the beginning of this preface. While many of the chapters actually address issues at multiple levels, reflecting the comprehensive nature of the associated research project, they have been organized into these volumes on the basis of the primary conceptual contribution of the work. Agha and Sturman, for example, describe a framework (reflective architectures), a paradigm (replicated actors), and a prototype implementation (the Screed language and Broadway runtime system). But because the salient attribute of this work is the use of reflection to dynamically adapt an application to its environment, it is included in the *Frameworks* volume.

Volume I, *Models and Frameworks for Dependable Systems*, presents two comprehensive frameworks for reasoning about system dependability, thereby establishing a context for understanding the roles played by specific approaches presented throughout the series. This volume then explores the range of models and analysis methods necessary to design, validate, and analyze dependable systems.

Volume II, *Paradigms for Dependable Applications,* presents a variety of specific approaches to achieving dependability at the application level. Driven by the higher level fault models of Volume I and buiilt on the lower level abstractions implemented in Volume III, these approaches demonstrate how dependability may be tuned to the requirements of an application, the fault environment, and the characteristics of the target platform. Three classes of paradigms are considered: protocol-based paradigms for distributed applications, algorithm-based paradigms for parallel applications, and approaches to exploiting application semantics in embedded real-time control systems.

Volume III, *System Implementation,* explores the system infrastructure needed to support the various paradigms of Volume II. Approaches to implementing

suppport mechanisms and to incorporating additional appropriate levels of fault detection and fault tolerance at the processor, network, and operating system level are presented. A primary concern at these levels is balancing cost and performance against coverage and overall dependability. As these chapters demonstrate, low overhead, practical solutions are attainable and not necessarily incompatible with performance considerations. The section on innovative compiler support, in particular, demonstrates how the benefits of application specificity may be obtained while reducing hardware cost and run-time overhead.

This first volume in the series covers system architectures or frameworks that serve as the foundation for dependable system design and the various models required at each layer of the system hierarchy and stage of its lifecycle to guide design decisions and evaluate their effectiveness.

Section 1 presents two frameworks for the study and design of dependable systems. Malek emphasizes the layered view of dependability advocated throughout this series and presents the concept of universal consensus for realizing dependability at each level in the context of distributed real-time systems. Agha and Sturman introduce the concepts of reflection and encapsulation as vehicles for tuning dependability to a dynamically changing fault environment and application requirements while maintaining the transparency of dependability mechanisms to the application itself. The concepts are made concrete through the example of an actor-based language and run-time system, highlighting the importance of language hooks in granting users enhanced control over detection and recovery mechanisms.

Given these frameworks, Section 2 addresses the issue of mathematically characterizing dependability requirements in a manner exploitable by them. Monaghan introduces the section by outlining the real requirements demanded by typical military systems and the difficulty of precisely translating those requirements for systems designers and verifying the results. In real-time systems, dependability encompasses timeliness as well as correctness. Shin presents an approach to deriving precise deadline constraints from the application semantics to provide dependability mechanisms with maximum flexibility based on true requirements rather than specifications of undocumented origin. Shin also extends the layered view of dependability to the larger system level: an erroneous output from an embedded computer, while technically a failure of that computer, may be still recoverable at the system level if the controlled process is robust enough.

Once a system is designed using appropriate high-level abstractions and fault models the problem remains to validate the design against the types of faults anticipated in actual operation. An emerging approach to this critical problem is fault injection, in which the response of the system to low-level injected errors is gauged. Efficiency of this process demands an intermediate model to guide injection that preserves coverage while simplifying and accelerating the testing. One such approach

xii

and the issues involved in applying it are examined by Yount and Siewiorek in Section 3. Clark and Pradhan complete the picture by describing the REACT testbed for modeling and validating dependable system designs.

Whereas the models presented thus far capture the behavior of the system in response to particular fault scenarios, a global, quanititative analysis of system dependability in terms of the probabilistic measures of reliability, availability, and performability is necessary in order to judge whether the overall requirements have met and to guide allocation of resources to the most critical system components. Iyer and Tang take an empirical approach using data from operational systems to drive their models, identify trends, and capture the shifting focus of dependability concerns as hardware and software technology evolve. Lin takes an analytical approach to developing a quantitative method for evaluating design alternatives in the context of optoelectronic interconnection networks.

Gary M. Koob
Mathematical, Computer and Information Sciences Division
Office of Naval Research

Clifford G. Lau
Electronics Division
Office of Naval Research

ACKNOWLEDGEMENTS

ERRATA

1. Due to a late editing decision, Section 2.2 was removed from this volume. The somewhat anomalous numbering scheme employed in Section 2 reflects the original organization.

2. The following notes were inadvertently omitted from Section 3.2:

- This section is based on research sponsored, in part, by the Office of Naval Research under grants N00014-91-J-1404 and N00014-92-J-1366 and conducted at the University of Massachusetts at Amherst and Texas A&M University.

- Jeffrey A. Clark is with the MITRE Corporation, Bedford, Massachusetts.

- Dhiraj K. Pradhan is with Texas A&M University, College Station, Texas.

CONTENTS

SECTION 1

FRAMEWORKS FOR
DEPENDABLE SYSTEMS