SERVICE AVAILABILITY
PRINCIPLES AND PRACTICE
Edited by Maria Toeroe and Francis Tam
SERVICE AVAILABILITY
SERVICE AVAILABILITY
PRINCIPLES AND PRACTICE

Editors

Maria Toeroe
_Ericsson, Canada_

Francis Tam
_Nokia Research Center, Finland_
Service availability : principles and practice / editors, Maria Toeroe, Francis Tam.

p. cm.

Includes bibliographical references and index.
ISBN 978-1-119-95408-8 (cloth)
1. Reliability (Engineering) I. Toeroe, Maria. II. Tam, Francis.
TA169.S465 2012
620'.00452--dc23

2011047219

A catalogue record for this book is available from the British Library.

Print ISBN: 9781119954088

Typeset in 9/11pt Times by Laserwords Private Limited, Chennai, India
## Contents

**List of Contributors**  
**Foreword**  
**Preface**  
**Acknowledgments**  
**List of Abbreviations**

### Part I  INTRODUCTION TO SERVICE AVAILABILITY

1  Definitions, Concepts, and Principles  
   *Francis Tam*

1.1 Introduction  
1.2 Why Service Availability?  
   1.2.1 Dossier on Unavailability of Service  
   1.2.2 Issues and Challenges  
1.3 Service Availability Fundamentals  
   1.3.1 System, Behavior, and Service  
   1.3.2 Dependable Computing Concepts  
   1.3.3 The Meaning of Availability  
1.4 Achieving Service Availability  
   1.4.1 Following the Framework of Fault Tolerance  
   1.4.2 Redundancy is a Requisite  
   1.4.3 Dealing with Failures  
   1.4.4 Upgrade Matters  
1.5 Conclusion

2  The Birth of the Service Availability Forum  
   *Francis Tam*

2.1 Introduction  
2.2 Technology Environment  
2.3 Business Environment  
   2.3.1 Ecosystem  
   2.3.2 COTS and Open Systems
<table>
<thead>
<tr>
<th>Part II</th>
<th>THE SA FORUM SYSTEM: SERVICES AND FRAMEWORKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Overview of the Service Availability Architecture</td>
</tr>
<tr>
<td>Dave Penkler</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Background and Business Context</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Goals and Requirements</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Service Availability Architecture Scope and Presentation</td>
</tr>
<tr>
<td>3.2</td>
<td>HA Concepts Applied</td>
</tr>
<tr>
<td>3.2.1</td>
<td>To Be or Not to Be High Availability Aware</td>
</tr>
<tr>
<td>3.2.2</td>
<td>HA Aware Application Perspective</td>
</tr>
<tr>
<td>3.3</td>
<td>Architecture</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Basic Architectural Model</td>
</tr>
<tr>
<td>3.3.2</td>
<td>The AIS Services and Frameworks Architecture</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Service Dependencies</td>
</tr>
<tr>
<td>3.4</td>
<td>Open Issues</td>
</tr>
<tr>
<td>3.4.1</td>
<td>The Optional Features Issue</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Integrated AIS Service API</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Common Low Level Communication Facility Interface</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Common Distributed Process Management Interface</td>
</tr>
<tr>
<td>3.4.5</td>
<td>System Trace Service</td>
</tr>
<tr>
<td>3.4.6</td>
<td>Diagnostics Framework</td>
</tr>
<tr>
<td>3.4.7</td>
<td>Overload Control Framework</td>
</tr>
<tr>
<td>3.5</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>

| 4 | The SA Forum Information Model: The Heart of Control and Monitoring |
| Maria Toeroe |
| 4.1 | Introduction |
| 4.2 | Background |
| 4.2.1 | Management Models Out There |
| 4.2.2 | The SA Forum Needs |
| 4.3 | The SA Forum Information Model |
| 4.3.1 | Overview of the SA Forum Solution |
| 4.3.2 | Administrative and Management Aspects |
| 4.3.3 | Application Information Models |
| 4.3.4 | Open Issues and Recommendations |
| 4.4 | Conclusion |

| 5 | Consistent and High Level Platform View |
| Maria Toeroe |
| 5.1 | Introduction |
| 5.2 | Hardware Platform Interface |
| 5.2.1 | Background |
5.2.2 Overview of the Hardware Platform Interface 87
5.2.3 The HPI Model 88
5.2.4 HPI Capability Discovery 93
5.2.5 Error Handling and Administrative Operations 94
5.2.6 Open Issues and Conclusions 95
5.3 Platform Management Service 96
5.3.1 The Conception of PLM 96
5.3.2 Overview of the SA Forum Platform Management 97
5.3.3 The PLM Information Model 98
5.3.4 Tracking of PLM Entities 107
5.3.5 Administrative and Management Aspects 110
5.3.6 Service Interaction 118
5.3.7 Open Issues and Conclusions 120
5.4 Cluster Membership Service 121
5.4.1 Background 121
5.4.2 Overview of the Cluster Membership Service 122
5.4.3 CLM Configuration: The Bootstrap Trap 125
5.4.4 Are You a Member? 126
5.4.5 Administrative and Management Aspects 127
5.4.6 Service Interaction 129
5.4.7 Open Issues 130
5.4.8 Recommendation 131
5.5 Conclusion 131

6 Model Based Availability Management: The Availability Management Framework 133
   Maria Toeroe
6.1 Introduction 133
6.2 Background 134
6.2.1 Error Detection and Repair 134
6.2.2 Fault Zones and Error Escalation 135
6.2.3 Separation of Services from Serving Entities 136
6.2.4 Service Provisioning Roles 136
6.2.5 Delicacies of Service State Replication 137
6.3 The Availability Management Framework 138
6.3.1 Overview of the SA Forum Solution 138
6.3.2 Components and Component Service Instances 139
6.3.3 The AMF Information Model 148
6.3.4 Redundancy Models 167
6.3.5 The AMF Administrative Interface 176
6.3.6 Interactions Between AMF and Other AIS Services 187
6.3.7 Open Issues 190
6.3.8 Recommendation 191
6.4 Conclusion 191

7 Communication and Synchronization Utilities 193
   Maria Toeroe and Sayandeb Saha
7.1 Introduction 193
7.2 Event Service 194
  7.2.1 Background: Event Service Issues, Controversies, and Problems 194
  7.2.2 Overview of the SA Forum Event Service 195
  7.2.3 Event Service Architecture and Model 196
  7.2.4 User Perspective 200
  7.2.5 Administrative and Management Aspects 201
  7.2.6 Service Interactions 201
  7.2.7 Open Issues and Recommendations 202
7.3 Message Service 202
  7.3.1 Need for Reliability and Load Distribution 202
  7.3.2 Overview of the SA Forum Message Service 203
  7.3.3 Message Service Architecture and Model 205
  7.3.4 User Perspective 207
  7.3.5 Administrative and Management Aspects 210
  7.3.6 Service Interaction 210
  7.3.7 Open Issues and Recommendations 211
7.4 Checkpoint Service 212
  7.4.1 Background: Why Checkpoints 212
  7.4.2 Overview of the SA Forum Checkpoint Service 213
  7.4.3 Checkpoint Service Model 215
  7.4.4 User Perspective 217
  7.4.5 Administrative and Management Aspects 220
  7.4.6 Service Interaction 221
  7.4.7 Open Issues 222
  7.4.8 Recommendation 222
7.5 Conclusion 223
  7.5.1 Common Issue: Entity Names 223
  7.5.2 Conclusion 223
8 Services Needed for System Management 227
  Maria Toeroe
  8.1 Introduction 227
  8.2 Log Service 228
    8.2.1 Background: Data, Data, and More Data 228
    8.2.2 Overview of the SA Forum Solution 229
    8.2.3 The LOG Information Model 231
    8.2.4 User Perspective 232
    8.2.5 Administrative and Management Aspects 233
    8.2.6 Service Interaction 233
    8.2.7 Open Issues and Recommendations 235
8.3 Notification Service 236
    8.3.1 Background: Issues, Controversies, and Problems 236
    8.3.2 Overview of the SA Forum Notification Service 237
    8.3.3 User Perspective 239
    8.3.4 Correlation of Notifications 241
    8.3.5 Administrative and Management Aspects 243
    8.3.6 Service Interaction 244
    8.3.7 Open Issues and Recommendation 246
8.4 Information Model Management Service 247
  8.4.1 Background: Issues, Controversies, and Problems 247
  8.4.2 Overview of the SA Forum IMM Solution 249
  8.4.3 The Object Manager API 251
  8.4.4 The Object Implementer API 255
  8.4.5 IMM XML File 258
  8.4.6 Administrative and Management Aspects 258
  8.4.7 Service Interaction 258
  8.4.8 Open Issues 260
  8.4.9 Recommendation 261
8.5 Conclusion 262

9 Model-Based Software Management: The Software Management Framework 265
  Maria Toeroe
  9.1 Introduction 265
  9.2 Background 266
  9.3 Software Management a la Carte 268
    9.3.1 Overview of the SA Forum Solution 268
    9.3.2 Entity Types File: Is It Eaten or Drunk by SMF? 271
    9.3.3 The Upgrade Campaign and Its Specification 273
    9.3.4 Upgrade Campaign Execution Status and Failure Handling 279
    9.3.5 Administrative and Management Aspects 285
    9.3.6 User Perspective 288
    9.3.7 Service Interaction 289
    9.3.8 Open Issues 291
    9.3.9 Recommendation 292
  9.4 Conclusion 294

10 Combining the Services 297
  Maria Toeroe
  10.1 Introduction 297
  10.2 Application Design and Development 297
  10.3 Application Platform Design 299
  10.4 Operation and Maintenance 301

Part III SA FORUM MIDDLEWARE IN ACTION

11 SA Forum Programming Model and API Conventions 305
  Francis Tam
  11.1 Introduction 305
  11.2 Programming Model 306
    11.2.1 AIS Area Service Interfaces 306
    11.2.2 Real-Time Support 306
    11.2.3 Naming Conventions and Type Definitions 308
    11.2.4 Usage Model and Library Life Cycle 309
    11.2.5 Tracking 311
11.3 Making Sense of the API Specifications
   11.3.1 Structure of Service API Specification
   11.3.2 Administration API
11.4 Practical Topics
   11.4.1 Interacting with POSIX
   11.4.2 Allocating and Freeing Memory
   11.4.3 Handling Pointers
   11.4.4 Finding Out Implementation Limits
   11.4.5 When an Area Service is Unavailable
   11.4.6 Backward Compatibility
11.5 Concluding Remarks

12 SA Forum Java Mappings: Specifications, Usage, and Experience
   Robert Hyerle and Jens Jensen
   12.1 Introduction
   12.2 Background
   12.2.1 Early Exploration of Java Mappings in Hewlett-Packard
   12.2.2 Java in Ericsson
   12.2.3 The SA Forum Java Mapping Initiative
12.3 Understanding the Java Mappings
   12.3.1 Java Application Integration Architecture
   12.3.2 Naming
   12.3.3 Package Structure
   12.3.4 The Underlying Objects
   12.3.5 Types
   12.3.6 Parameters, Exceptions, and Method Signatures
   12.3.7 Factories, Callbacks, and Life-cycles
   12.3.8 Callbacks and the Selection Object in Java
12.4 Using the Java Mappings
   12.4.1 Integrating AIS Services with Java Applications
   12.4.2 Integrating AIS Services with Containerized Java Applications
   12.4.3 AIS Services in Mixed Language and Mixed Implementation Environments
12.5 Going Further
   12.5.1 The Java Mapping Roadmap
   12.5.2 Related Java Standards and Other References

13 SA Forum Middleware Implementations
   Mario Angelic and Ulrich Kleber
   13.1 Introduction
   13.1.1 OpenHPI
   13.1.2 OpenSAF
13.2 The OpenHPI Project
   13.2.1 Overview of the OpenHPI Solution
   13.2.2 User Perspective
   13.2.3 OpenHPI Tools
   13.2.4 Open Issues and Recommendations
13.3 The OpenSAF Project
   13.3.1 Background
13.3.2 OpenSAF Architecture 356
13.3.3 SA Forum Compliant Services 360
13.3.4 OpenSAF Infrastructure Services 364
13.3.5 Managing OpenSAF 365
13.3.6 Deploying OpenSAF 367
13.4 Conclusion 368

14 Integration of the VideoLAN Client with OpenSAF: An Example 371
Anik Mishra and Ali Kanso
14.1 Introduction 371
14.2 Going Under the Hood: The VLC Workflow 372
14.3 Integrating VLC with OpenSAF 373
14.3.1 Nonproxied-Non-SA-Aware Integration 374
14.3.2 SA-Aware VLC Integration 379
14.3.3 SA-Aware VLC with Service Continuity 384
14.4 Summary and Conclusion 387

15 Migration Paths for Legacy Applications 391
Mario Angelic
15.1 Introduction 391
15.2 Reasons for Migration 392
15.2.1 Benefits for System Owners 392
15.2.2 Benefits for ISVs 392
15.3 Integration Criteria 393
15.3.1 Main Factors 393
15.3.2 Easy Management 394
15.3.3 Streamlined Architecture 396
15.3.4 Code Quality 397
15.3.5 Integration Levels 397
15.4 How to Migrate 399
15.4.1 Availability Integration 399
15.4.2 Manageability Integration 409
15.5 Open Issues 413
15.6 Conclusion 413

16 Overcoming Complexity: Formal Modeling Techniques at the Rescue 415
Maria Toeroe and Ferhat Khendek
16.1 Introduction 415
16.2 Background 416
16.2.1 The Model-Based Approach 416
16.2.2 Starting Points in the Specifications 417
16.3 Model-Based Software Management 419
16.3.1 Configuration Model 419
16.3.2 Configuration Generation 420
16.3.3 Upgrade Campaign Generation 424
16.3.4 Analytical Models and How They Can Help 427
16.4 Conclusion 428
17 Conclusion
  17.1 Summary 431
  17.2 The Future 433

References 435

Index 443
List of Contributors

Mario Angelic, Ericsson, Stockholm, Sweden
Robert Hyerle, Hewlett-Packard, Grenoble, France
Jens Jensen, Ericsson, Stockholm, Sweden
Ali Kanso, Concordia University, Montreal, Quebec, Canada
Ferhat Khendek, Concordia University, Montreal, Quebec, Canada
Ulrich Kleber, Huawei Technologies, Munich, Germany
Anik Mishra, Ericsson, Town of Mount Royal, Quebec, Canada
Dave Penkler, Hewlett-Packard, Grenoble, France
Sayandeb Saha, RedHat Inc., Westford, Massachusetts, USA
Francis Tam, Nokia Research Center, Helsinki, Finland
Maria Toeroe, Ericsson, Town of Mount Royal, Quebec, Canada
The need to keep systems and networks running 24 hours a day, seven days a week has never been greater, as these systems form some of the essential fabric of society ranging from business to social media. Keeping these systems running in the presence of hardware and software failures is defined as service availability. In some areas of networking, such as telecommunications, it has formed an essential requirement for almost 100 years; it is part of why traditional plain old telephone service (POTS) would still be available when power went out. With the advent of the Internet, service availability requirements are increasingly being demanded in the marketplace, not necessarily due to regulatory requirements, as was the case with telephone networks, but due to business requirements and pressures from the marketplace. Of course, it’s not just communications where service availability is important, many other industries such as aerospace and defense have similar requirements. Imagine the impact of a loss of control during missile flight, for example.

After the Internet bubble of the late 1990s, and an almost global deregulation of the telecommunications market, it was increasingly recognized that the high cost of development for proprietary hardware and software systems was no longer viable. The future would increasingly be based on commercial off-the-shelf (COTS) systems, where time to market for new services, outweighs the elegance of proprietary hardware and software systems. High availability middleware, which forms a core aspect of delivering service availability, was one of these complex components. Traditionally viewed as high value and differentiating, in this new environment of time to market service emphasis, where rapid application development, adaptation, and integration are key, proprietary middleware is both time consuming to develop and costly to maintain.

The Service Availability Forum (SA Forum) was established in 2001 to help realize the vision of accelerating the implementation and deployment of service available systems, through establishing a set of open specifications which would define the boundaries between hardware and middleware and between the middleware and the application layer. At the time, concepts which are generally accepted today, such as a layered approach to building systems, the use of off-the-shelf hardware and software, and de facto standards developed through open source, were in their relative infancy.

The Founders of the SA Forum, Force Computers, GoAhead Software, HP, IBM, Intel, Motorola, Nokia, and Radisys all recognized that in 2001 the world was changing. They understood that redundancy and service availability would spread downstream from the traditional high end applications, such as telecommunications and that the key to success was a robust ecosystem built around a set of open specifications for service availability. This would allow applications to run on multiple platforms, with different hardware and operating systems, and enable rapid and easy integration of multiple applications onto a single platform, realizing the vision of rapid development to meet the demands of new services in the marketplace. None of what was envisioned precluded the continued development of proprietary systems, but the concepts were clearly aimed at the increased use of COTS hardware and software with a view accelerating the interoperability between components.

Although it has changed over time, as the organization and the market has evolved, the current mission statement of the SA Forum characterizes the objectives set out in 2001.
The Service Availability Forum enables the creation and deployment of highly available, mission critical services by promoting the development of an ecosystem and publishing a comprehensive set of open specifications. A consortium of industry-leading companies, the SA Forum maintains ‘There is no Upside to Downtime.’

It is always a challenge to create an industry organization when so much investment in proprietary technology already exists. On the one hand, there needs to be a willingness to bring some of this expertise and possibly intellectual property to the table, to serve as a basis for creating the specifications. This has to be tempered with the fear that someone will contribute intellectual property and later aggressively seek to assert patent rights. To avoid issues in this area, the SA Forum was established as a not-for-profit organization and a key aspect of the bylaws was that all members agreed to license any intellectual property to any other members on fair and reasonable terms. Since the SA Forum was dealing primarily in software application programming interfaces around an underlying conceptual architecture, the assertion of patents is quite difficult, but in any event, the Forum has always operated on a cooperative model, with everyone seeking to promote the common good and to address differences within the technical working groups. To further control the objective of a common goal, the SA Forum established three levels of membership, promoters, contributors, and adopters. An academic (associate) membership level was added at later date, and the status of adopter was conferred on anyone with an implementation and use of the specifications in a product.

Promoters were the highest level, and only promoters could be on the board of directors. They were the founders of the organization, and hence the main initial contributors. To avoid predatory actions by other companies, additional promoters could be added only by a unanimous vote of all the promoters. While this may seem overly restrictive, it has worked well in practice, and companies who have demonstrated commitment and who have contributed to the Forum have been offered promoter status.

In order to participate in SA Forum work groups and contribute to the specifications, companies had to be contributor members. This proved to be the workhorse membership level for the organization and many valuable contributions came from this group of members.

The adopter members have generally been companies with interest in supporting the SA Forum’s work, or who have developed products that have incorporated some aspect of the SA Forum’s specifications.

The cooperative nature of the SA Forum has led to the development of a robust set of specifications for service availability. Indeed, that is what this book is all about, the concepts and use of the SA Forum specifications.

The first tentative steps after the formation in 2001 were white papers on the then new concepts of service availability and a layered architecture approach. These were followed by the initial specifications focused on the hardware platform interface (HPI), which has gone through a number of revisions and enhancements. The most recent release of the HPI specification includes provisions for firmware upgrades and hardware diagnostics.

Work on the more challenging application interface specification (AIS), which address the interfaces to applications, management layers, and overall control of the availability aspects of a distributed system. Early work focused on what has come to be known as the utility services, the fundamental services necessary to create a service available system, cluster concepts, checkpointing, messaging, and so on. By the 2005–2006 timeframe, the Forum was ready to address overall system concepts, such as defining the framework and policy models for managing availability. This resulted in the Availability Management Framework (AMF) and the Information Model Management (IMM). These critical services provide both the flexibility to architect a system to meet application requirements, but also a common mechanism for managing availability, with extensibility to manage applications themselves if desired. This complex work really created the core of the SA Forum AIS and it is in many
ways a remarkable piece of work. More recent developments have included the Software Management Framework (SMF) to enable seamless upgrading (and downgrade if necessary) campaigns for systems, demonstrating the true idea of service availability, and platform management (PLM), which enables a coherent abstraction of a system. This encompasses complex hardware designs with computer boards equipped with mezzanine cards, which are themselves compute engines, and enables modern virtual machine architectures to be embraced by the SA Forum system model. This in turn enables the SA Forum specifications to become an essential part of cloud computing concepts.

The SA Forum itself has been responsible for the genesis of other industry organizations. It was recognized that the scope of the SA Forum was insufficient to meet the objective of the wide-spread adoption of off-the-shelf technology and the cooperation between the component layers of the solution. By its very charter, the SA Forum was focused on service availability and middleware. An outgrowth of the Forum was the creation in 2007 of the SCOPE Alliance.

The SCOPE Alliance was founded by Alcatel-Lucent, Ericsson, Motorola, NEC, Nokia, and Siemens. It is a telecom driven initiative which now includes many leading network equipment providers, hardware, and software companies, with the mission to enable and promote a vibrant carrier grade base platform (CGBP) ecosystem for use in telecom network element product development. The SCOPE members believe that a rich ecosystem of COTS and free open source software (FOSS) communities provide building blocks for the network equipment manufacturers to adopt, accelerating their time to market and better serving the service provider marketplace.

To accomplish these goals, SCOPE has created a reference architecture which has been used to publish profiles that define how off-the-shelf technologies can be adopted for various application and platform requirements. These profiles also identify where gaps exist between the various layers of CGBP technology. A core component of the CGBP is service availability middleware, based on SA Forum specifications.

Creating specifications is a complex and intellectually challenging task. This is an accomplishment in and of itself. However, the success of the SA Forum and its specifications is really measured by their adoption in the marketplace and their use in systems in the field. Over the years, there have been a number of implementations of the specifications. When the Forum was founded, and the use of open source software was in its infancy, it was foreseen that the specifications would enable multiple implementations and the portability would be accomplished at the application programming interface (API) layer. From 2006 onwards, the Forum had various initiatives aimed at demonstrating portability. Multiple companies did indeed implement some or part of the specifications to varying degrees. These implementations ranged from selected services to complete implementations of the specifications.

On the hardware side, most major hardware vendors have adopted the HPI specification. There are both proprietary, commercial implementations and an open source solution, OpenHPI, available in the marketplace. With the broad adoption of HPI, this can be very much considered a success in the marketplace.

AIS is much more complex and a range of proprietary and open source solutions have appeared in the marketplace since the mid-2000s. These have had various levels of implementation relative to the specifications discussed in this book, and they have included internal development by network equipment manufacturers, proprietary commercial products, and open source solutions. OpenAIS is an open source solution dating from around 2005 and it has been used extensively for clustering in the Linux community. The most complete implementation of the AIS is the OpenSAF project, this is a focus for many adopters of the SA Forum AIS moving forward, with rollout commitments from major equipment manufacturers and a vibrant ecosystem.

Many people, from a wide variety of companies, have contributed to the SA Forum specifications, and their effort and foresight have led to a framework that is now being implemented, adopted,
and deployed. The current focus is on expanding the use cases for the SA Forum specifications and demonstrating that they address a broad range of applications. This goes beyond the traditional five and six ‘9’s’ of the telecom world and the mission critical requirements of aerospace and defense, to the realms of the Enterprise and the emerging cloud computing environment.

Timo Jokiaho  
*Chairman of the SCOPE Alliance, 2011, President of the SA Forum, 2003*

John Fryer  
*President of the SA Forum, 2011*
Preface

How This Book Came About

Maria’s Story

I joined the Service Availability (SA) Forum in 2005 with the mandate of representing Ericsson in the efforts of the SA Forum Technical Working Group (TWG) to define the Software Management Framework. This is where I met Francis and the representatives of other companies working on the different specifications. The standardization has been going on already for several years and I had a lot to learn and catch up with. Unfortunately there was very little documentation available besides the specifications themselves, which of course were not the easiest introduction to the subject.

Throughout the discussions it became even more obvious that there was an enormous ‘tribal knowledge’ – as someone termed it – at the base of the specifications. This knowledge was not written anywhere, not documented in any form. One could pick it up gradually once he or she started to decipher the acronym ridden discussions flying high in the room and on the email reflectors. There were usually only a handful who could keep up with these conversations at the intensity that was typical at these discussions. For newcomers they were intimidating to say the least. This was an issue for the SA Forum from the beginning and for the years to come even though there was an Educational Working Group with the mandate to prepare training materials. Many TWG members felt that it would be good to write a book on the subject, but with everyone focusing on the specifications themselves there was little bandwidth to spare for such undertake.

Gradually I picked up most of the tribal knowledge and was able to participate in those discussions, but preparing educational materials or writing a book still did not come to my mind until Ericsson started a research collaboration with Concordia University. Suddenly I had to enlighten my students about the mysteries of the SA Forum specifications. These specifications are based on the years of experience of telecom and information technology companies in high-availability cluster computing. These systems evolved behind closed doors in those companies as highly guarded secrets and accordingly very little if any information was available about them in the public domain. This also meant that the materials were not taught at universities nor were books readily available to which I could refer my students. Soon the project meetings turned into an ad-hoc course where we went through the different details, the intricacies of the specifications and the reasoning behind the solutions proposed. These solutions were steeped in practice and brewed for production. They reflected what has worked for the industry as opposed to theoretical models and proofs more familiar to the academia. This does not mean that they lack theoretical basis. It just means that their development was driven by practice.

Understanding all these details was necessary before being able to embark on any kind of research with the students and their professors. These discussions of course helped the students but at the same time they helped me as well to distill the knowledge and find the best way to present it. Again it would have been nice to have a book, but there was none, only the specifications and the knowledge I gathered in the TWG discussions.

A few years later OpenSAF, the open source implementation of the SA Forum specifications reached the stage when people started looking at it from the perspective of deployment. They started
to look for documentation, for resources that they could use to understand the system. OpenSAF uses mostly the SA Forum specifications themselves as documentation for the services compliant to these specifications.

These people faced the same issue I had experienced coming to the world of the SA Forum. I was getting requests to give an introduction, a tutorial presentation so that colleagues can get an idea what they are dealing with, how to approach the system, where to start. After such presentations I would regularly get the comment that ‘you should really write a book on this subject.’ At this time I saw the suggestion of writing a book more realistic and also with the increasing demand for these presentations it made a lot of sense.

In a discussion with my manager I mentioned the requests I was getting to introduce the SA Forum specifications and the suggestions about the book. He immediately encouraged me to make a proposal. This turn of events transformed the idea I have toyed with for some time into a plan and the journey has begun. I have approached Francis and others I knew from the SA Forum to enroll them in the book project. This book is the realization of this plan, the end of this journey. It is a technical book with a rather complex subject that we, the authors and editors tried to present in a digestible way.

Francis’ Story

My contribution related to the SA Forum specifications in this book was based on the project titled ‘High Availability Services: Standardization and Technology Investigation’ that I worked on during 2001–2006 in Nokia Research Center. The project was funded by Strategy and Technology, the then Nokia Networks (now part of Nokia Siemens Networks), with the objective to support the company’s standardization effort in the SA Forum and contribute to a consistent carrier-grade base platform architecture for the then Nokia Networks’ business. I became one of the Nokia representatives to the SA Forum and took part in the development of the first release of the Availability Management Framework specification with other member companies’ representatives. Subsequently, I took up the role of co-chairing with Maria the Software Management specification development group. Regrettably I had to stop my participation in the SA Forum at the end of 2006 before the Software Management Framework was published.

Parallel to my full-time employments over the years, I have been giving a 12-hour seminar course on highly available systems to the fifth (final) year Master of Engineering students in Computer Science at INSA Lyon (Institut National des Sciences Appliquées de Lyon) in France almost every year since 1993. It has been widely recognized in the academic community that there is a lack of suitable books for teaching the principles and a more pragmatic approach to designing dependable computer systems. Very often such materials have to be gathered from various sources such as conference proceedings, technical reports, journal articles, and the like, and put together specifically for the courses in question. On a number of occasions, the thought of writing such a book came to my mind but it left rather quickly, probably due to my senses were warning me that such an undertaking would have been too much.

I remember it was a few years ago when Maria asked me if I could recommend a book in this area for her teaching. After explaining to her about the general situation with regard to books in this subject area, I half-jokingly suggested to her that we could write one together. She left it like that but only returned in January 2010 and asked if I would be interested in a book project. As they say, the rest is history.

The Goal of the Book

Our story of how the book came about has outlined the need that has built up and which it was time to address with a book. It was clear that the approach to the subject should not be too theoretical,
but rather an explanation of the abstractions used in the SA Forum specifications that would help practitioners in mapping those abstractions to reality; it also needed to make the knowledge tangible, to show how to build real systems with real applications using the implementations of the SA Forum specifications. The time was right as these implementations were reaching maturity fast.

At the same time we did not want to write a programmers’ guide. First of all a significant portion of the specifications themselves is devoted to the description of the different application programming interface (API) functions. But there is so much reasoning in these systems and the beauty of their logic cannot be delivered just by discussing the APIs, which are like the scrambled puzzle pieces do not reflect the complete picture, the interconnection and interdependencies until they are put together piece by piece. They give little information on the reasoning which animates the picture and fills in even missing puzzle pieces.

The specifications may not be perfect at this time yet but they bring to the light this technology that has been used and proved itself in practice to provide the magic five-nine figures of in service performance, but has been hidden from the public eye. At this time they already come with open source implementations meaning that they are available for anyone to experiment with or to use for deployment, and also to evolve and improve.

The concepts used in these specifications teach a lot about how to think about systems that need to provide their services continuously 24/7 in the presence of failures. Moreover they are designed to evolve respecting these same conditions, that is, these systems and their services develop without being taken out for planned maintenance, they evolve causing minimal service outage. They are ideal for anyone who needs to meet stringent service level agreements or SLAs.

The concepts presented in this book remain valid whether they are used in the framework of the SA Forum specifications or transpired to cloud computing or any other paradigm that may come. The SA Forum specifications provide an excellent basis to elaborate and present the concepts and the reasoning. They also set the terminology allowing for a common language of discussion, which was missing for the area.

We set out to explain these concepts and their manifestation in the specifications and demonstrate their application through use cases.

So who would benefit from this book? The obvious answer is that applications and systems designers who intend to use the SA Forum middleware. However since we look at the specifications more as one possible manifestation of the concepts, ultimately the book benefits anyone who needs to design systems and applications for guaranteed service availability, or who would like to learn about such systems and applications. We see this book as a basis for an advanced course on high service availability systems in graduate studies or in continuous education.

**The Structure of the Book**

The book is divided into three main parts:

Part One introduces the area of service availability, its basic concepts, definitions, and principles that set the stage for the subsequent discussions. It also delivers the basic premise that makes the subject timely. Namely that in our society the demand for continuous services is increasing in terms of the number and variety of services as well as the number of customers. To meet this demand it is essential to make the enabling technologies widely available by standardizing the service APIs so that commercial off the shelf components can be developed. Enabling such an ecosystem was the mission of the SA Forum, whose coming about is presented also in this part.

Part Two of the book focuses on the specifications produced by the SA Forum to achieve its mission. The intention was to provide an alternative view of the specifications, a view that incorporates that ‘tribal knowledge’ not documented anywhere else and which provides some insight to the specifications, to the choices that were made at their design.
We start out with the architectural overview of the SA Forum middleware and its information model. The subsequent chapters elaborate on the different services defined by the SA Forum Architecture. Among them the Availability Management Framework and the Software Management Framework each has their own dedicated chapter while the other services are presented as functional groups: the Platform services, the Utility services, and the Management Infrastructure services.

Rather than discussing all the SA Forum services at a high level we selected a subset on which we go into deeper discussions so that the principles become clear. We do not cover the Security service in our discussions as it is a subject crosscutting all the services and easily filling a book on its own.

The presentation of the different services and frameworks follow more or less the same pattern:

First the goals and the challenges addressed by the particular service are discussed, which are followed by an overview of the service including the service model and architecture supporting the proposed solution.

Rather than presenting the gory details of each of the API functions like it would be in a programmer’s guide we decided to explain the usage through the functionality that can be achieved by using the APIs. This approach reveals better the complete picture behind the puzzle pieces of the API functions. We mention the actual API functions only occasionally when it makes it easier to clarify the overall functionality.

Whenever it is applicable we also present the administrative perspective of the different services. The goal of these sections is to outline what a system administrator may expect to observe in a running system and what control he or she can obtain through configuration and administrative operations according to the specification. Sometimes these details could be overwhelming, so the anticipation is that different implementations of the standard services may restrict this access while other vendors may build management applications that enhance the experience by assisting the administrator in different ways.

Subsequently the service interactions are presented inserting the service discussed thus far in isolation into the environment it is expected to operate. Since the specifications themselves are written in a somewhat isolated way, these sections collect information that are not readily available, which require the understanding of the overall picture.

Finally the open issues and recommendations conclude each of the service overviews.

Part Three of the book looks at the SA Forum middleware in action, that is, at the different aspects of the practical use of the specifications presented in Part Two.

It starts with the overview of the programming model used throughout the definition of the different service APIs. There is a system in the API definitions of the different specifications and Chapters 11
and 12 serve as Ariadne’s thread in what seem to be a labyrinth. This is followed by a bird’s-eye view at the two most important open source implementations of the SA Forum specifications: OpenSAF and OpenHPI.

To help integrators and application developers to use these middleware implementations in Chapter 14 we discuss different levels of integration of the VideoLAN Client (VLC) application originally not developed for high availability. This exercise demonstrates in practice how an application can take advantage of the SA Forum Availability Management Framework even without using any of its APIs. Of course better integration and better user experience can be achieved using the APIs and additional services, which is also demonstrated.

After this ‘hands on’ exercise the problem of migrating large scale legacy applications is discussed. This chapter gives an excellent insight not only for those considering such migration, but also to designers and developers of new applications. It demonstrates the flexibility of the SA Forum specifications which people usually realize only after developing an intimate relationship with them. The mapping of the abstractions defined by the specifications is not written in stone and it is moldable to meet the needs of the situation. This is demonstrated on the example of two different database integrations with the SA Forum middleware depending on the functionality inherent in the database.

The final chapter of Part Three takes yet again a different perspective. It discusses the issues complementary to the specifications but necessary for the operation of the SA Forum middleware. It introduces the use of formal models and techniques to generate system configurations and upgrade campaigns necessary for the Availability and the Software Management Frameworks to perform their tasks. This approach was part of the vision of the SA Forum specifications as they defined the concepts enabling such technology opening the playground for tool vendors.

We could have continued exploring the subject with many exciting applications, but we had to put an end as we reached our page limit as well as the deadline for delivering the manuscript. So we leave the rest of the journey to the reader who we hope will be well equipped after reading our book to start out with their own experimentations.
Acknowledgments

The group of people that were essential for the creation of this book are the Service Availability (SA) Forum’s Technical Working Group representatives of the different member companies; who concocted the specifications and provided a challenging yet inspiring environment for learning and growing in the field. We cannot possibly list all the participants without missing a few, so we will not do so. There were however a few outstanding:

We had extremely constructive and rewarding discussions with the SA Forum Software Management Working Group when we were creating the Software Management Framework, for which we would like to thank Peter Frejek, Shyam Penubolu, and Kannan Kasturi. We probably should not forget about another regular participant of our marathon-length conference calls: the Dog whose comments broke the seriousness of the discussions.

We would like to thank Fred Herrmann, who left his fingerprints over most if not all SA Forum service specifications, and for the numerous stimulating discussions and debates which made the experience so much more exciting. And in the debates it was a pleasure to have the calming wisdom of Dave Penkler. Dave was also instrumental in the writing and reviewing of this book. We are grateful to him for graciously stepping up and helping out with key chapters when we were under pressure of time and short of a pair of fresh eyes.

We are deeply obliged to our co-authors for helping us create this book. For most of them this meant the sacrifice of their spare time – stealing it from their families and friends to deliver the chapters and with that make the book so much more interesting.

Finally we would like to thank Wiley and in particular Sophia Travis for recognizing the vision in our book proposal and helping us through the stress of the first book with such an ease that it truly felt like a breeze.

From Maria

First and foremost I would like to thank the generosity of Ericsson and within that of my managers Magnus Buhrgard and Denis Monette for allotting me the time to work on this book and their continuous support and trust that it would be completed. Not that I ever had a doubt, but it definitely took more time and efforts than I anticipated. Their support made the whole project possible.

I am also grateful to the MAGIC team of Concordia University. The professors: Ferhat Khendek, Rachida Dssouli, and Abdelwahab Hamou-Lhadj, the students Ali Kanso, Setareh Kohzadi, Anik Mishra, Ulf Schwekendiek, Pejman Salehi, and the post-docs: Pietro Colombo and Abdelouahed Gherbi. They provided me with a completely different learning experience. All of them had their own approach to the problem and in the discussions I had to learn to investigate the subject from many different sometimes unconventional angles and answer questions that within industry were taken for granted. These discussions and working together on the problems led me to a fresh look and a deeper understanding of the subject all facilitating (at least in my belief) a better delivery.
Finally I would like to thank my colleagues in Montreal and across the sea in Stockholm who were the initiators of this project with their requests and suggestions, who joined my family and friends, in supporting and encouraging me in my writing from the beginning.

A heartfelt thank to all of you.

Maria Toeroe
September, 2011

From Francis

The undertaking to write a book is a daunting commitment even in the best of times, having to do it in my spare time after the day job was rather demanding. My contribution to this book would not have been possible if it was not for the thoughtful understanding and unreserved support from my wife Riikka, who has the shared belief that this book project was good for me. She deserves a medal for putting up with my long evenings and weekends of writing.

As if my lack of time were not enough, I went through one round of company reorganization and was under the threat of lay-off for some weeks – a slightly different kind of redundancy I originally planned to think about. My warm thank you goes to Minna Uimonen, who has always encouraged me and reminded me of the Finnish *sisu* during this difficult time. I am grateful to all my friends for their kind wishes and understanding of my short disappearance. I look forward to re-integrating with the community and do what I do best – as a highly available ‘Chief Entertainment Officer.’

Francis Tam
September, 2011
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>3rd generation</td>
</tr>
<tr>
<td>3PP</td>
<td>Third Party Product</td>
</tr>
<tr>
<td>ABI</td>
<td>Application Binary Interface</td>
</tr>
<tr>
<td>AIS</td>
<td>The SA Forum Application Interface Specification</td>
</tr>
<tr>
<td>AMF</td>
<td>The SA Forum Availability Management Framework</td>
</tr>
<tr>
<td>AMM</td>
<td>Availability Management Middleware</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ARP</td>
<td>Address Resolution Protocol</td>
</tr>
<tr>
<td>ASN.1</td>
<td>Abstract Syntax Notation One</td>
</tr>
<tr>
<td>ATCA</td>
<td>Advanced Telecommunication Computing Architecture</td>
</tr>
<tr>
<td>ATL</td>
<td>ATLAS Transformation Language</td>
</tr>
<tr>
<td>BASH</td>
<td>Born Again Shell</td>
</tr>
<tr>
<td>CASE</td>
<td>Computer-Aided Software Engineering</td>
</tr>
<tr>
<td>CCB</td>
<td>Configuration Change Bundle</td>
</tr>
<tr>
<td>CGBP</td>
<td>Carrier Grade Base Platform</td>
</tr>
<tr>
<td>CIM</td>
<td>Common Information Model</td>
</tr>
<tr>
<td>CIMOM</td>
<td>Common Information Model Object Manager</td>
</tr>
<tr>
<td>CKPT</td>
<td>the SA Forum Checkpoint Service</td>
</tr>
<tr>
<td>CLC-CLI</td>
<td>component life-cycle command line interface</td>
</tr>
<tr>
<td>CLI</td>
<td>command line interface</td>
</tr>
<tr>
<td>CLM</td>
<td>the SA Forum Cluster Membership Service</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>COTS</td>
<td>commercial-off-the-shelf</td>
</tr>
<tr>
<td>CPU</td>
<td>central processing unit</td>
</tr>
<tr>
<td>CSI</td>
<td>component service instance</td>
</tr>
<tr>
<td>CST</td>
<td>component service type</td>
</tr>
<tr>
<td>DAM</td>
<td>dependability analysis modeling</td>
</tr>
<tr>
<td>DAT</td>
<td>domain alarm table</td>
</tr>
<tr>
<td>DBMS</td>
<td>database management system</td>
</tr>
<tr>
<td>DET</td>
<td>domain entity tree</td>
</tr>
<tr>
<td>DIMI</td>
<td>Diagnostics Initiator Management Instrument</td>
</tr>
<tr>
<td>DMTF</td>
<td>Distributed Management Task Force</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DN</td>
<td>distinguished name</td>
</tr>
<tr>
<td>DNS</td>
<td>domain name server</td>
</tr>
<tr>
<td>DRT</td>
<td>domain reference table</td>
</tr>
<tr>
<td>(E)AM</td>
<td>external active monitoring</td>
</tr>
<tr>
<td>EE</td>
<td>execution environment</td>
</tr>
<tr>
<td>ETF</td>
<td>entity types file</td>
</tr>
<tr>
<td>EVT</td>
<td>the SA Forum Event Service</td>
</tr>
<tr>
<td>FAR</td>
<td>Federal Acquisition Regulation</td>
</tr>
<tr>
<td>FOSS</td>
<td>free open source software</td>
</tr>
<tr>
<td>FRU</td>
<td>field replaceable unit</td>
</tr>
<tr>
<td>FT</td>
<td>fault tolerant</td>
</tr>
<tr>
<td>ftp</td>
<td>file transfer protocol</td>
</tr>
<tr>
<td>FUMI</td>
<td>Firmware Upgrade Management Instrument</td>
</tr>
<tr>
<td>GUI</td>
<td>graphical user interface</td>
</tr>
<tr>
<td>HA</td>
<td>high availability</td>
</tr>
<tr>
<td>HE</td>
<td>hardware element</td>
</tr>
<tr>
<td>HP</td>
<td>Hewlett-Packard</td>
</tr>
<tr>
<td>HPI</td>
<td>the SA Forum Hardware Platform Interface</td>
</tr>
<tr>
<td>HTTP</td>
<td>hypertext transmission protocol</td>
</tr>
<tr>
<td>HW</td>
<td>hardware</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines</td>
</tr>
<tr>
<td>ID</td>
<td>identifier</td>
</tr>
<tr>
<td>IDR</td>
<td>Inventory Data Record</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IFIP</td>
<td>International Federation for Information Processing</td>
</tr>
<tr>
<td>iLO2</td>
<td>HP Integrated Lights-Out 2</td>
</tr>
<tr>
<td>IMM</td>
<td>the SA Forum Information Model Management Service</td>
</tr>
<tr>
<td>I/O</td>
<td>input/output</td>
</tr>
<tr>
<td>IP</td>
<td>the Internet Protocol</td>
</tr>
<tr>
<td>IPMB</td>
<td>intelligent platform management bus</td>
</tr>
<tr>
<td>IPMI</td>
<td>intelligent platform management interface</td>
</tr>
<tr>
<td>ISP</td>
<td>in-service performance</td>
</tr>
<tr>
<td>ISV</td>
<td>independent software vendor</td>
</tr>
<tr>
<td>IT</td>
<td>information technology</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunication Union</td>
</tr>
<tr>
<td>Java EE</td>
<td>Java Enterprise Edition (formerly J2EE)</td>
</tr>
<tr>
<td>JCP</td>
<td>Java Community Process</td>
</tr>
<tr>
<td>JMX</td>
<td>Java Management eXtenstions</td>
</tr>
<tr>
<td>JSR</td>
<td>Java specification request</td>
</tr>
<tr>
<td>JVM</td>
<td>Java virtual machine</td>
</tr>
<tr>
<td>LDAP</td>
<td>lightweight directory access protocol</td>
</tr>
<tr>
<td>LOG</td>
<td>the SA Forum Log service</td>
</tr>
<tr>
<td>MAGIC</td>
<td>Modeling and Automatic Generation of Information for Configuration and upgrade campaigns for service availability</td>
</tr>
<tr>
<td>MARTE</td>
<td>OMG’s Modeling and Analysis of Real-Time Embedded systems</td>
</tr>
<tr>
<td>MDA</td>
<td>model driven architecture</td>
</tr>
<tr>
<td>MDE</td>
<td>model driven engineering</td>
</tr>
<tr>
<td>MDS</td>
<td>message distribution service</td>
</tr>
<tr>
<td>MIB</td>
<td>management information base</td>
</tr>
</tbody>
</table>