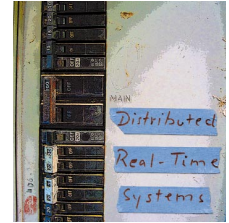


Real-Time Distributed Object Computing: An Emerging Field



While the field of object-oriented real-time computing (ORC) is young, it is growing quickly because it offers such a wide range of applicability, from complex real-time systems to the next generation of computing and communication devices. The purpose of this special issue is to present significant advances in both the art and science of ORC technology.

Eltefaat Shokri
Sun Microsystems

Phillip Sheu
University of California,
Irvine

In the 1980s, distributed systems technology became increasingly popular in the world of real-time computing, so much so that the adoption of RT distributed computing systems started to accelerate during the 1990s. And today, RT distributed computing is a field of its own, and one of the fastest-growing areas in technology.

Basic object-oriented computing technology has prevailed over non-RT business data processing, as evidenced by the numerous success stories about effectively using OO methodologies to reduce the development complexity and maintenance costs of large-scale applications. On the other hand, RT systems engineering techniques have moved very slowly toward maturity, and they have not advanced sufficiently in the areas of modular design and analysis, the main battle cry of OO proponents.

As a result, the RT systems engineering community has had to struggle with the strong demand for analysis, design, and implementation of large-scale RT applications. Only recently has the integration of these two important fields—OO design and RT computing—attracted researchers from both academia and industry.

A QUICKLY GROWING FIELD

While the field of object-oriented real-time computing (ORC) is young, it is growing quickly because it offers such a wide range of applicability, from complex real-time systems to the next generation of computing and communication devices. The increasing popularity of ORC conferences attests to this.

IEEE has two regular conferences focused on ORC:

- the annual International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC) series that started in 1998, and
- the International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS) series that started in 1994 and emphasizes the dependability and the quality-of-service aspects of ORC.

The purpose of this special issue is to present significant advances in both the art and science of ORC technology. One way to look at this collection of articles is to consider them a snapshot taken at the early stage of ORC technology evolution. Of course we're limited by the realities of publishing—namely the page constraints—so our snapshot will only be a relatively small part of the complete picture. This snapshot, however, is the result of our best efforts to grasp the most significant elements of the technology's evolution.

With this collection of articles, we've tried to emphasize innovative and practical solutions for integrating OO computing technologies into RT systems engineering methods. We are pleased to be able to present four valuable articles that cover a wide range of issues that center on developing object-oriented, real-time distributed computing systems.

THE SNAPSHOT

In "The Real-Time Specification for Java," Greg Bollella and James Gosling examine the technical

We believe that ORC will soon become one of the most active research fields in the domain of computer science and engineering.

issues and the rationale for the design choices of the real-time Java specification, defined by the Real-Time for Java Experts Group (RTJEG). Real-time Java is an important technology that will help us move toward the realization of a widely used real-time software development platform. The article presents, in particular, the semantics of scheduling, memory management, and synchronization—the core issue for real-time applications. Bollella and Gosling target both the general software development community and real-time application practitioners.

OMG's real-time CORBA defines standard features for specifying the configurations and controls of processor resources and communication resources to achieve certain end-to-end quality of service as an integral part of object interfaces. The real-time CORBA community has identified several focus points including

- somewhat restricted fixed-priority-based applications, and
- more general dynamic-priority-based applications.

In “An Overview of the Real-Time CORBA Specification,” Douglas C. Schmidt and Fred Kuhns discuss the real-time CORBA approaches to defining QoS attributes for distributed objects for resource management—focusing in particular on the fixed-priority approach. The article also describes the general directions in which enforcement of the QoS requirements in CORBA-based distributed real-time applications is moving.

UML has become an industry standard and is enjoying widespread acceptance. In “A Generic Framework for Modeling Resources with UML,” Bran Selic presents a generic framework for incorporating QoS attributes into the Unified Modeling Language. Selic's work represents an effort to deal with one of the main concerns of real-time systems: resource specification and QoS modeling. We hope that Selic's article will encourage others from both industry and academia to explore further extensions of UML to model essential aspects of real-time systems.

While other articles concentrate on incorporating real-time distributed systems needs and concerns as add-ons into the standard language or framework, Kane Kim's article, “APIs for Real-Time Distributed Object Programming,” discusses the issue of defining a high-level API for real-time distributed objects in a platform-independent manner that does not require new language translators. Kim's article looks at real-time distributed programming from the viewpoint of enabling current-generation C++ and Java programmers in the business application programming field to

easily enter the world of real-time distributed programming and start producing high-precision, high-level, real-time distributed objects. The article provides application interfaces for essential elements of high-level, real-time distributed programming, such as global time, guaranteed completion time of services, and time-triggered (or spontaneous) actions.

We believe that ORC as both a research field and a practice will grow rapidly from this point on. The basic programming tools for ORC, including those discussed in this special issue, will likely show an accelerated spreading pattern this year and for several years to come.

The levels of abstraction at which the real-time programmers can work in their programming activities will also steadily widen with the discovery of new upper ends. The basic goal of ORC is to realize major improvements in the productivity of the programmers and the reliability and maintainability of the real-time distributed computing systems produced. We are certain that we will hear about a growing number of success stories from the designers who apply ORC technologies to the development of real-time distributed computing systems.

On the other hand, the issues that the research community needs to tackle to enable full exploitation of the potential power of ORC are enormous. Many software engineering tools and formal foundations need to be established. Therefore, we believe that ORC will soon become one of the most active research fields in the domain of computer science and engineering. We hope that *Computer's* readers, especially the real-time systems community, will benefit from this collection of articles. ★

Acknowledgments

We thank the authors of all submitted articles and the reviewers for helping us prepare this special issue.

Eltefaat Shokri is a staff engineer at High Availability Telecom Software, Sun Microsystems. His research interests include real-time object-oriented distributed systems and high availability. Contact him at eltefaat.shokri@eng.sun.com.

Phillip Sheu is a professor in the Electrical and Computer Engineering Department at the University of California, Irvine. His research interests include object-oriented visual databases and object-oriented interactive systems. Contact him at sheu@ece.uci.edu.