Preface

This book is a collection of articles about the influence that the recent greater scope and availability of wide area networks is having on the semantics, design, and implementation of programming languages. The Internet has long provided a global computing infrastructure but, for most of its history, there has not been much interest in programming languages tailored specifically to that infrastructure. More recently, the Web has produced a widespread interest in global resources and, as a consequence, in global programmability. It is now commonplace to discuss how programs can be made to run effectively and securely over the Internet.

The Internet has already revolutionized the distribution and access of information, and is in the process of transforming commerce and other areas of fundamental importance. In the field of programming languages, the Internet is having a deep revitalizing effect, by challenging many fundamental assumptions and requiring the development of new concepts, programming constructs, implementation techniques, and applications. This book is a snapshot of current research in this active area.

The articles in this book were presented at the Workshop on Internet Programming Languages, which was held on May 13, 1998 at Loyola University, Chicago, USA. The papers submitted to the workshop were screened by the editors. After the workshop, the presented papers were refereed by an external reviewer and one of the editors, resulting in the current selection.

This workshop provided a forum for the discussions of all aspects of computer languages for wide area systems, including specification languages, programming languages, semantics, implementation technologies, and application experience.

Together the papers give a good impression of active research topics in this area. Mobility is one theme that is touched by several of the papers, either in the form of mobile objects, mobile agents, computation migration, or Tuple Space migration. Also, agent technology, communication constructs, security, and fault tolerance are important issues addressed in many papers. Finally, the papers clearly show that Internet programming languages are a fruitful research area for theoretical and experimental, as well as application-driven work.

The paper *"Location-Independent Communication for Mobile Agents: A Two-Level Architecture"* by Peter Sewell, Paweł T. Wojciechowski, and Benjamin C. Pierce presents a two-level calculus for distributed computation. The low-level calculus can be fairly directly implemented on existing distributed infrastructures, while the high-level calculus, which includes location-independent communication primitives between mobile agents, is implemented on top of the low-level calculus. The critical high-level primitive is the reliable delivery of a message to an agent, wherever the agent is. Two infrastructure algorithms are described for the location-independent delivery of messages to mobile agents. The first algorithm uses a simple centralized database of agents and their current locations. The second algorithm has distributed site maps where each site maintains forwarding pointers for agents that have left. Both algorithms are described as formal translations from the high-level to the low-level calculus. The meaning of the calculi, and hence of the algorithms, is precisely given using the standard techniques of process semantics.

The paper *"A Lightweight Object Migration Protocol"* by Peter Van Roy, Per Brand, Seif Haridi, and Raphaël Collet describes a fault-tolerant protocol that supports lightweight mobile objects. The resulting model ensures that the two basic requirements, centralized semantics and predictable network behavior, are supported. To achieve this, issues of distributed execution, freely mobile objects, and fault-tolerant protocol are addressed. A distribution graph models the various language entities that are subject to distributed execution. Lightweight mobility is achieved by maintaining local copies of the code and by lazily copying the state. Finally, a distributed algorithm to implement the fault-tolerant protocol is described.

The paper *"Seal: A Framework for Secure Mobile Computations"* by Jan Vitek and Giuseppe Castagna presents the Seal calculus, a process calculus with hierarchical protection domains. The paper discusses the requirements for modeling Internet computation, including comparisons with other well-known concurrent formalisms. The Seal calculus is proposed as a foundation for computation over the Internet, and it includes, in particular, sophisticated mobility and security mechanisms. In this framework, communication may be local within a domain, or cross a domain boundary, in which case it is subject to policy controls. As a special case of communication, processes (and process hierarchies) may be moved from place to place. Seals have great control on interactions that happen within them; for example, communication and migration must be explicitly enabled and agreed upon by all parties.

The paper *"A Run-Time System for WCL"* by Antony Rowstron and Stuart Wray describes a coordination language (WCL) for Internet and Web-based agent systems. WCL is based on the Tuple Space model introduced by Linda and gives support for distributed (rather than parallel) applications. WCL supports several additional Tuple Space primitives, such as asynchronous communication and bulk data transfers. The paper describes an experimental run-time system for WCL, supporting tuple space migration. Also, it gives performance results on a distributed system running at three sites in the UK and Germany.

The paper *"PML: A Language Interface to Distributed Voice-Response Units"* by J. Christopher Ramming describes the design and implementation of a Phone Markup Language (PML). PML is used to program interfaces for Voice Response Units (VRU). PML is an imperative language that is based on HTML, but is intended for use in speech contexts. The paper introduces the basic constructs of PML and describes its abstract syntax. Issues, such as efficiency, static analysis, safety, and security are among the requirements that PML implements. An example that illustrates the use of PML is provided.

The paper *"Derivatives: A Construct for Internet Programming"* by Dominic Duggan describes a novel communication construct: a non-blocking receive operation that allows execution to continue speculatively, based on assumptions about the received value (such speculative value is called a derivative). A "case" construct, examining a derivative, introduces a rollback point that is activated if the value that eventually replaces the derivative is not the expected one. A “sendout” construct provides a non-rollbackable action. The paper describes the formal semantics of derivatives and, in particular, a semantics for rollback, which is a complex concept that is often useful in distributed algorithms.

The paper *“Network Programming Using PLAN”* by Michael Hicks, Pankaj Kakkar, Jonathan T. Moore, Carl A. Gunter and Scott Nettles discusses a programming language for active networks. An active network allows users to add functionality to network routers, but a key issue is how to do this in a secure way. PLAN (Packet Language for Active Networks) is a simple language that is limited enough to allow a secure implementation, yet flexible enough to support several interesting applications of active networks. The paper describes the language and several applications, including a tracerouter, multicast, and adaptive routing.

The referees that assisted us in the selection process were:

Cedric Fournet, Microsoft Research, UK
Andrew D. Gordon, Microsoft Research, UK
Wilhelm Hasselbring, Tilburg University, The Netherlands
James Jennings, Tulane University, USA
Orlando Karam, Tulane University, USA
Gilles Muller, INRIA/IRISA, France
John H. Reppy, Bell Laboratories, Lucent Technologies, USA
Nicole Terry, Tulane University, USA
Maarten van Steen, Vrije Universiteit Amsterdam, The Netherlands
Scott William, Tulane University, USA

July 1999

Henri Bal
Boumediene Belkhouche
Luca Cardelli

Contents