Fundamenta Informaticae 73 (2006) i-ii
IOS Press

Foundations on Coordination Languages and Software Architectures

Preface

Modern information systems rely more and more on combining concurrent, distributed, mobile and heterogenous components. This move from old systems, typically conceived in isolation, induces the need for new languages and software architectures. In particular, coordination languages have been proposed to cleanly separate computational aspects and communication. On the other hand, software architects face the problem of specifying and reasoning on non-functional requirements. All these issues are widely perceived as fundamental to improve software productivity, enhance maintainability, advocate modularity, promote reusability, and lead to systems more tractable and more amenable to verification and global analysis.

The aim of the Foclasa'02 workshop was to bring together researchers working on the foundations of component-based computing, coordination, and software architectures. As an answer to the call for papers, 13 papers were selected for presentation at the workshop. The best six papers, according to the results of the anonymous peer reviews, were invited to participate in the special issue. The extended versions of these six papers went through a second round of anonymous peer reviews, and their revised versions are included in this special issue.

The paper by I. Linden et al. proposes a theoretical study of an extension of the seminal coordination language Linda proposed by Carriero and Gelerner. This extension refines Linda's set of inter-agent communication primitives with a new matching mechanism based on pairs composed of attribute names associated with their values. Computations in this language are described by means of an operational semantics, reporting the whole traces of executions. The non-compositionality of this intuitive operational semantics motivates the design of a compositional and fully abstract denotational semantics.

An appealing property of Linda's primitives is that they act asynchonously and on the basis on information. Consequently Linda naturally proposes an open distributed model in which producers and consumers are completely decoupled. As a result, any consumer can listen to any information produced. The paper by R. Gorrieri et al. presents a novel model in which security properties may be expressed. Three basic mechanisms are described: partition of Linda's tuple space, extra information added to tuples to authenticate producers and consumers, and access control policies based on the kind of operations an agent performs on a tuple.

The paper by M. Viroli and A. Omicini highlights coordination from the point of view of services. To that end, they propose an ontology in which the entities and abstractions involved in the coordination process are clearly separated in three different roles: the coordinated entities, the interaction space, and the coordination media. The evolution of the whole system is expressed in terms of the single coordination medium's interactive behaviour, which is explicitly characterised in terms of production and consumption of communication events through the interaction space.

The paper by S. Orzan and J. van de Pol presents a distributed software architecture based on communication by means of read and write primitives on a global set. In addition to the model, an implementation is described and proved correct by using the ACP process algebra framework. Moreover, the expressiveness power of the model is studied thanks to this ACP setting.

The paper by J. Guillen-Scholten et al. proposes an alternative approach to coordination based on mobile channels. In this work, computations are seen as performed by components accessible through their interfaces and linked by special connections, possibly being dynamically reconfigured. A trace semantics of this model is provided by the paper together with an implementation in Java.

Finally, the paper by A. Vallecillo et al. studies a framework for typing the behavior of objects in component models. In particular, they show how session types can be used not only for high level specifications of complex object interactions but also to define interoperability tests at the protocol level.

Running the Foclasa'02 workshop and building this issue would have been impossible without the help of numerous persons. We are grateful to L. Brim, M. Kretinsky, and A. Kucera for their help in organizing the workshop in Brno. We also like to thank the anonymous reviewers for their excellent job as well as the program committee members of Foclasa'02, R. De Nicola, J. Fiadeiro, R. Gorrieri, P. Inverardi and A. Porto, for their help in selecting the papers for the workshop and for this issue.

The guest editors

Antonio Brogi
Department of Computer Science, University of Pisa, Pisa, Italy
brogi@di.unipi.it
Jean-Marie Jacquet
Institute of Informatics, University of Namur, Namur, Belgium
jmj@info.fundp.ac.be
Joost Kok
Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands
joost@liacs.nl