



ELSEVIER

Science of Computer Programming 46 (2003) 1–3

**Science of
Computer
Programming**

www.elsevier.com/locate/scico

Preface

Issues in Coordination Languages and Architectures

About Coordination

Coordination languages are a class of programming languages expressly invented to manage the interactions among components of distributed software architectures. Gelernter and Carriero used the term *Coordination* in the following context:

Programming = Computation + Coordination.

They formulated this equation to explain the use of the coordination language Linda. Their point is that there should be a clear separation between the description of the *agents of the computation* and the description of their *interaction* in the overall architecture of a software system. On the one hand, such a separation facilitates the architectural reuse of code; on the other hand, the same interaction patterns occur in many different problems — so it becomes possible to reuse the coordination component as well!

Usually coordination languages are not full-fledged programming languages; rather, they are often defined as language extensions or scripting languages as they are mostly concerned with coordination issues.

Roughly, coordination research concentrates on the following topics:

- coordination models, namely the study of semantic models and logics suitable to describe coordination languages and their concepts: coordination entities, media, and rules;
- coordination mechanisms and language design, namely the study of new coordination primitives, languages, and related programming tools like compilers and debuggers; and
- coordination architectures and applications, namely the study of interaction structures, their specification and design into real software systems.

The collection [1] includes some early papers on the theme of coordination models and languages. The first International Conference on Coordination Languages and Models was held in Italy in April 1996 [2]. A selection of revised papers from that conference was published into a special issue of Science of Computer Programming in 1998 (Vol. 31, number 2/3). The second conference was held in Berlin in September

1997 [4]. The third conference was held in Amsterdam 1999 [3]. The fourth conference was held in Cyprus in 2000 [5]. The fifth conference will be held in New York, UK, in 2002. The papers here are a selection of completely revised papers from the third conference.

About this issue

We invited some selected participants from the third conference, to contribute to this SCP special issue. The submissions were subjected to the usual reviewing process and six papers were finally selected for inclusion in this special issue.

“*Static analysis of real-time component-based systems configurations*” by Attanasio, Corradini and Inverardi, is a paper about the static analysis of software architectures based on coordinable components.

“*Comparing coordination models and architectures using embeddings*” by Bonsangue, Kok, and Zavattaro, is a paper comparing computational expressiveness of some basic coordination models using the technique of embeddings.

“*On the expressiveness of coordination via shared dataspace*”, by Brogi and Jacquet, is another study on the expressiveness of coordination primitives based on three different flavors of shared dataspace that are basically some Linda extensions.

“*Erratic Fudgets: a semantic theory for an embedded coordination language*” by Moran, Sands, and Carlsson, introduces a theoretical model and a functional language including some special coordination features.

“*Using mobile code to provide fault tolerance in tuple space based coordination languages*” by Rowstron, is another paper about mobile code in the context of some specific coordination media.

“*Coordinating processes with secure spaces*” by Vitek, Bryce, and Oriol, addresses an important problem in designing languages, including coordination media used for mobile agents, namely security.

Acknowledgements

We thank the anonymous reviewers who helped us in the selection: their help was invaluable in putting together this special issue.

References

- [1] J.-M. Andreoli, C. Hankin, D. Le Métayer (Eds.), *Coordination Programming: Mechanisms, Models and Semantics*, Imperial College Press, Singapore, 1996.
- [2] P. Ciancarini, C. Hankin (Eds.), *First Internat. Conf. on Coordination Languages and Models*, Vol. 1061, *Lecture Notes in Computer Science*, Springer, Berlin, 1996.
- [3] P. Ciancarini, A. Wolf (Eds.), *Third Internat. Conf. on Coordination Languages and Models*, Vol. 1594, *Lecture Notes in Computer Science*, Springer, Berlin, 1999.
- [4] D. Garlan, D. LeMétayer (Eds.), *Second Internat. Conf. on Coordination Languages and Models*, Vol. 1594, *Lecture Notes in Computer Science*, Springer, Berlin, 1997.

- [5] A. Porto, G.C. Roman (Eds.), Fourth Internat. Conf. on Coordination Languages and Models, Vol. 1906, Lecture Notes in Computer Science, Springer, Berlin, 2000.

Paolo Ciancarini
*Dipartimento di Scienze dell'Informazione,
Università di Bologna, Italy
E-mail address: ciancarini@cs.unibo.it*

Alexander Wolf
*Department of Computer Science,
University of Colorado at Boulder, USA
E-mail address: alw@cs.colorado.edu*