Journal of Systems Integration, 10, 103–105 (2001)

Coordination as a Paradigm for Systems Integration

ANDREA OMICINI aomicini@deis.unibo.it
DEIS, Universitû di Bologna, Bologna, Italy

GEORGE A. PAPADOPOULOS george@cs.ucy.ac.cy
Department of Computer Science, University of Cyprus, Nicosia, Cyprus

According to [4], the term "coordination" basically means the management of the interaction among the entities of a system—whether they are agents, processes, molecules, individuals, or whatever. Generally speaking, the notion of coordination is a multi-disciplinary one, and has been given several specific definitions in the many research areas where it is relevant and commonly used, such as Programming Languages, Parallel and Distributed Information Systems, (Distributed) Artificial Intelligence, Internet Technologies and Software Engineering.

One of the fundamental characteristics in "coordination programming" is the separation of concerns between what is being computed and how the entities that comprise a system of computational units forming an application interact with each other. Thus, a coordination framework comprises three fundamental elements:

- the coordinables, i.e., the entities whose mutual interaction is ruled by a coordination model;
- the coordination media, i.e., the abstractions enabling the interaction among coordinables (e.g., channels, blackboards, etc.);
- the coordination laws, defining the behavior of the coordination media in response to interaction events.

This separation of concerns leads to a natural simplification process for the development of large scale complex systems. More to the point, the coordination paradigm enables the development of a coordinable independently from the application within which it will be incorporated, but also from any other coordinables. Coordinables can be developed with the aim of being independent and reusable computational units which will be assembled together with other similar units to form applications, the management of the assembled system itself being taken care of by the coordination framework employed which, effectively, provides a glueing mechanism between its individual components (namely coordinables). Thus, coordination can be seen as a form of components-based systems development, offering a mechanism able not only to develop individual components but, perhaps more importantly, also to integrate a number of them into complex systems. It can be shown, that coordination shares the same objectives with other similar approaches in systems development such as Software Architectures and Architecture Description Languages, Configuration Management and Languages, etc. [6].

It is quite interesting to note that in many coordination models and languages the individual coordinable units are treated as black boxes, whose internals are hidden to the outside world. This means that the development and deployment of these units can be decoupled in time and space from that of the other constituent parts of some application. It also means that the very nature of these units is immaterial, provided they have well known interfaces with which they communicate with the environment. Thus, they need not only be software units but also hardware devices and even people. This is another way to understand the inherent interdisciplinary nature of coordination and explain why it has been proved useful in modeling and realizing the integration of many types of systems including workflow systems, hybrid software/hardware systems, etc.

This special issue on "Coordination as a Paradigm for Systems Integration" comprises three papers that were selected from those accepted to appear in the Special Track on Coordination Models, Languages and Applications that was organized as part of the ACM 2000 Symposium on Applied Computing [1]. The call-for-papers for this track attracted 52 high quality submissions, out of which 17 were accepted as regular papers and two more as short papers. Among those accepted as regular papers the following three were selected to form this special issue.

The first two papers deal with coordination issues at the level of task parallelism whereas the third one addresses coordination at the higher level of modeling information systems activities. Pelagatti and Skillicorn [5] introduce the NOT model which allows adaptive node programs written in a variety of parallel languages to be connected together in an acyclic graph. The graph can be viewed as a coordination medium that allows program transformation in order to optimize resources, and provides the programmer with a mechanism for planning, assembling, scheduling and distributing the computational units forming a large scale system over, possibly geographically distributed, compute servers.

The second paper [3] introduces the coordination model of Activity Graphs which is effectively an intermediate layer for mapping programs expressed in skeletal languages (i.e., as skeletons) onto physical distributed architectures. The paper describes the mapping process from skeletons to an activity graph and the mapping of the graph itself down to MPI programs.

The final paper [2] addresses the issue of coordination at the higher level of workflow systems. The paper introduces an event based coordination framework that enables interactions and interoperability between workflows executing in different organizations, independent of both environment and executing platform.

The three papers comprising this special issue address some of the issues pertinent to the integration of large and complex systems. However, there are many other aspects of systems integration that can benefit from adopting the coordination paradigm, such as development and assembling of real-time components, building distributed multimedia (and possibly Internet-based) systems, cooperative computing, multi-agent based systems, etc. We believe that in the future we will watch coordination playing a more significant role in systems integration.

References

1. ACM SAC 2000, *Special Track on Coordination Models, Languages and Applications*, Como, Italy, 19–21 March, 2000, http://lia.deis.unibo.it/confs/SAC00.
2. F. Casati and A. Discenza, "Modelling and managing interactions among business processes," this special issue.
3. M. Cole and A. Zavanella, "Coordinating heterogeneous parallel systems with skeletons and activity graphs," this special issue.
4. T. Malone and K. Crowstone, "The interdisciplinary study of coordination." *ACM Computing Surveys* 26(1), pp. 87–119 (0000).
5. S. Pelagatti and D. B. Skillicorn, "Coordinating programs in the network of tasks model," this special issue.
6. G. A. Papadopoulos and F. Arbab, "Coordination models and languages," in M. V. Zelkowitz (ed.), "The engineering of large systems," *Advances in Computers*, Vol. 46, Academic Press, 1998, pp. 329–400.