Chapter 4
State-of-the Art Concepts and Future Directions in Modelling Coordination

Abdelhamid Abdelhadi Mansor
University of Khartoum, Sudan

Wan Mohd Nasir Wan Kadir
Universiti Teknologi Malaysia, Malaysia

ABSTRACT

Coordination is becoming an increasingly important paradigm for systems design and implementation. With multiple languages and models for coordination emerging, it is interesting to compare different models and understand their strengths and weaknesses find common semantic models and develop mappings between formalisms. This will help us to gain a deeper insight into coordination concepts and applications, and also to establish a set of features/criteria for defining and comparing coordination models. In this chapter, the authors present the current work on modelling coordination based on the coordination features. The findings show that software elements have three distinct types of coordination needs—technical, temporal, and process—and that these needs vary with the member's role; geographic distance has a negative effect on coordination, but is mitigated by shared knowledge of the team and presence awareness; and shared task knowledge is more important for coordination among collocated members. The authors articulate propositions for future research in this area based on the analysis.

INTRODUCTION

Coordination is the harmonious adjustment or interaction of different things to achieve a goal or effect (Canal et al., 2005). Coordination languages and models are being developed to address the problem of managing the interactions among concurrent and distributed processes (De Vries et al., 2009). The underlying principle is the separation of computations by components and their interactions (Colman & Han, 2005; Gesbert et al., 2007). To achieve correct coordination (Zhu, 2008), rather than only considering dependency relations between multiple adaptations, this approach further focuses on dependency relations between managers at runtime. This work considers

DOI: 10.4018/978-1-4666-6026-7.ch004

a number of features including specification to identify and measure achievement of managerial goals to insure that the modelling provides mechanisms for structuring or modularising coordination activities and to verify that coordinated managers do not have any explicit action that may affect the coordination (Nogueira et al., 2012; Wang et al., 2012).

Coordination is a central issue in software agent systems in particular, and in distributed artificial intelligence (DAI) in general (Nwana et al., 1996). However, it has also been studied by researchers in diverse disciplines in the social sciences, including organization theory, political science, social psychology, anthropology, law and sociology. For example, organization theorists have investigated the co-ordination of systems of human beings, from small groups to large formal organizations. Economists have studied coordination in markets of separate profit-maximising firms (Haruna, 2012). Even biological systems appear to be coordinated though individual cells or ‘agents’ act independently and in a seemingly non-purposeful fashion. Human brains exhibit coordinated behaviour from apparently ‘random’ behaviours of very simple neurones. Essentially, co-ordination is a process in which agents engage in order to ensure a community of individual agents acts in a coherent manner.

Coordination is important in software development because it leads to benefits such as cost savings, shorter development cycles, and better-integrated products (Omicini & Viroli, 2011; Wang et al., 2012). Team cognition research suggests that members coordinate through team knowledge, but this perspective has only been investigated in real-time collocated tasks and we know little about which types of team knowledge best help coordination in the most geographically distributed software work. In this field study, we investigate the coordination needs of software teams, how team knowledge affects coordination, and how this effect is influenced by geographic dispersion.

In this chapter we introduce definitions, advantages and disadvantages of coordination from different views. Some examples will be used to discuss and clarify the concepts of coordination. The aim of this chapter is to discuss about what coordination is, the different kinds of coordination that can be demanded by a system, and the relationships, differences and similarities between coordination and other fields such as adaptation. Moreover, the section started discussing about what is software coordination in a broad sense. Thus, thinking about what coordination is, the first emerging question is what kind of coordination can be adapted in a software system.

BACKGROUND

Coordination languages and models are being developed to address the problem of managing the interactions among concurrent and distributed processes (Nogueira et al., 2012). The underlying principle is the separation of computations by components and their interactions (Colman & Han, 2005; Gesbert et al., 2007). To achieve correct coordination, rather than only considering dependency relations between multiple adaptations, this approach further focuses on dependency relations between managers at run-time.

There are a number of works in autonomic system and formal methods with different kinds of models that have been proposed. Actor model was used in various methods for modelling open, concurrent and distributed systems, in which actors interact and coordinate system tasks through messages. However, a review of the literature suggests that a few researchers have considered higher level coordination between managers before a decision has been made, so that qualitative demands can be specified and realized by utilizing the power of coordination models.

The underlying computation model of concurrency, developed by Dinges and Agha (2012), is based on the actor model of distributed objects