IRM PRESS
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.irm-press.com
ITB9323

Chapter X

Temporal Interaction Diagrams for Multi-Process Environments

T. Y. Chen
Swinburne University of Technology, Australia

Iyad Rahwan
University of Melbourne, Australia

Yun Yang
Swinburne University of Technology, Australia

ABSTRACT

This chapter introduces a novel notion of temporal interaction diagrams for distributed and parallel programming. An interaction diagram is a graphical view of computation processes and communication between different entities in distributed and parallel processes. It can be used for the specification, implementation and testing of interaction policies in distributed and parallel systems. Expressing interaction diagrams in a linear form, known as

fragmentation, facilitate automation of design and testing of such systems. Existing interaction diagram formalisms lack the flexibility and capability of describing more general temporal order constraints. They only support rigid temporal order, and, hence, have limited semantic expressiveness. We propose an improved interaction diagram formalism in which more general temporal constraints can be expressed. This enables us to capture multiple valid interaction sequences using a single interaction diagram.

INTRODUCTION

Various attempts have been made to formalize interaction among computational entities, such as distributed, parallel, object-oriented and multi-agent systems (Hoare, 1985; Magee, Dulay, Eisenbach & Kramer, 1995; Ronnquist & Low, 1996; Koskimies, Männistö, Systä & Tuomi, 1996; Kinny, 1998; Chen, Rahwan & Yang, 2002; Bauer, 1999; Bauer, Müller & Odell, 2001). Traditionally, the system design stage involves a description of the steps taken in processing a particular task. In distributed systems, however, the implementation requires a clear picture of the separate computational threads of different processes. In parallel systems, it would be very helpful to be able to express the computation flow descriptions for different processes. Interaction diagrams are designed to support the representation and processing of these mingling activities. The linear representation of these diagrams facilitates the automation of the processes of diagram manipulation for design, report generation and testing. In particular, testing can be performed by comparing execution traces against specifications expressed in terms of interaction diagrams.

However, existing interaction diagram formalisms support quite rigid temporal order constraints only. An execution trace is said to be valid if it satisfies the interaction diagram. In other words, there is no way of specifying multiple valid traces without writing multiple versions of fixed execution traces. This can become a difficult job, especially in systems with sophisticated interactions. In such systems, the number of valid interactions can be quite large, and there is a demand to more concisely express such flexibility in a single interaction diagram. Our research is a step towards achieving this goal.

In this chapter, we give an overview of existing interaction diagram formalisms and present an enhancement of a particular framework in such a way as to support more flexible interaction specification. The chapter is organized as follows. In the next section, we present an overview of an existing interaction diagram framework and introduce the basic notation to be used throughout this chapter. Then we introduce our novel extension. After that, an example demonstrating the merit of our framework is presented. Then we discuss current and future trends in interaction diagram frameworks. Finally, we conclude and summarize our ideas and results.