An Extensible Space-Based Coordination Approach for Modeling Complex Patterns in Large Systems[⋆,⋆⋆]

Eva Kühn, Richard Mordinyi, and Christian Schreiber

Vienna University of Technology, Institute of Computer Languages,
Space Based Computing Group and Complex Systems Design and Engineering Lab,
Argentinierstr. 8, A-1040 Wien
{eva,richard,cs}@complang.tuwien.ac.at

Abstract. Coordination is frequently associated with shared data spaces employing Linda coordination. But in practice, communication between parallel and distributed processes is carried out with message exchange patterns. What, actually, do shared data spaces contribute beyond these? In this paper we present a formal representation for a definition of shared spaces by introducing an "extensible tuple model", based on existing research on Linda coordination, some Linda extensions, and virtual shared memory. The main enhancements of the extensible tuple model comprise: means for structuring of spaces, Internet- compatible addressing of resources, more powerful coordination capabilities, a clear separation of user data and coordination information, support of symmetric peer application architectures, and extensibility through programmable aspects. The advantages of the extensible tuple model (XTM) are that it allows for a specification of complex coordination patterns.

1 Introduction

The coordination theory was founded by Malone and Crowston and described as "managing dependencies between activities". In [16] it is argued that coordination makes sense only if tasks are interdependent. Additionally, the theory suggests that standardized coordination mechanisms can be applied to specific coordination problems. Ciancarini therefore describes a generic coordination model [4] as a triple of {E, M, L}. It suggests to have a clear separation between the specification of the communication entities of a system and the specification of their interactions or dependencies. In the model, {E} stands for either physical or logical entities to be coordinated. These can be data (structures), software processes, services, agents, or even human beings interacting

⋆ We would like to thank Stefan Craß, Geri Joskowicz, Hans Moritsch, Gernot Salzer, Thomas Scheller, Vesna Sesum-Cavic, and Ralf Westphal for their helpful discussions on this topic.

⋆⋆ The project is partly funded by TripCom (IST-4-027324-STP project, http://www.tripcom.org) and CAPI (project at TU-Vienna) of the Institute of Computer Languages.

T. Margaria and B. Steffen (Eds.): ISoLA 2008, CCIS 17, pp. 634–648, 2008.

with computer-based systems. {M} represents the coordination media (i.e. communication channels) serving as a connector between the entities and enables communication, which is a mandatory prerequisite for direct coordination [26,6]. Such coordination media may be message-passing systems, pipes, tuple spaces etc. {L} specifies the coordination laws between the entities defining how the interdependences have to be resolved and therefore, semantically define the coordination mechanisms.

From the point of view of designing a language for distributed systems the idea of associative communication based on a shared data space is one of the most interesting paradigms [3]. This is because a shared space allows to clearly separate the issue of controlling coordinating communication entities from the issue to control a single entity. A Tuple Space [22] is an example of this kind of languages. It is a well-known coordination model such as Linda [8], JavaSpaces [7] and TSpaces [15]. Tuple spaces are flat and unstructured multisets of tuples that can be accessed via very basic output, read, and input operations.

In the tuple space approach processes communicate with the other entities in the environment by writing tuples (ordered sequences of data) into the tuple space. Sharing of data via spaces [2] is not a novel paradigm. It comes from parallel processing and was later considered for distributed environments. Due to its high-level abstraction of communication by simply reading and writing data from/into a shared space this paradigm fits to growing dynamics and collaboration in the network [29]. The processes interested in retrieving information useful for coordinating their activities perform blocking **rd** or **in** operations specified via a template. In case several tuples match the template of a data-retrieval operation, only one of them is selected non-deterministically.

The limitation of current tuple space implementations is that they support template-matching only. This is problematic if any other form of coordination is needed, like FIFO. In such cases the coordination entity itself has to manage ordering of the tuples the right way and agree about it with other entities. Thus, the implementation of the coordination entity must contain functionalities that a coordination media should provide. However, queries involving relational comparison operators cannot be implemented with template matching. The proposed approach aims to describe a generic and extensible coordination model based on tags, upon which any kind of coordination laws can be modeled, starting with simple ones like FIFO and KEY, to more complex concurrent collection patterns, and finally also patterns that as e.g. described in [9] also cope with distributon. The extensible coordination approach is realized by means of the coordination media XVSM (extensible virtual shared memory) [25], [14] clearly separating the responsibilities between coordination middleware and entity again. The reason for adding additional forms of coordination laws to a space coordination medium are based on our experiences with programming real applications:

- Developers consider Linda template matching as too unstructured in comparison to query facilities offered by databases. Being forced to pack coordination information into tuple content is a drawback [17].