Science of Computer Programming 115–116 (2016) 79–99

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Blackboard rules: From a declarative reading to its application for coordinating context-aware applications in mobile ad hoc networks

Jean-Marie Jacquet [a], Isabelle Linden [b], Mihail-Octavian Staicu [a,*]

[a] University of Namur, Faculty of Computer Science, Rue Grandgagnage 21, Namur, Belgium
[b] University of Namur, Department of Business Administration, Rempart de la Vierge 8, Namur, Belgium

ARTICLE INFO

Article history:
Received 1 February 2013
Received in revised form 7 May 2015
Accepted 8 August 2015
Available online 12 August 2015

Keywords:
Blackboard rules
Context awareness
Coordination

ABSTRACT

Thanks to improvements in wireless communication technologies and increasing computing power in hand-held devices, mobile ad hoc networks are becoming an ever-more present reality. Coordination languages are expected to become important means in supporting this type of interaction. To this extent we argue the interest of the Bach coordination language as a middleware that can handle and react to context changes as well as cope with unpredictable physical interruptions that occur in opportunistic network connections. More concretely, our proposal is based on blackboard rules that model declaratively the actions to be taken once the blackboard content reaches a predefined state, but also that manage the engagement and disengagement of hosts and transient sharing of blackboards. The idea of reactiveness has already been introduced in previous work, but as will be appreciated by the reader, this article presents a new perspective, more focused on a declarative setting.

1. Introduction

On the front-line of current research areas, mobility related technologies are being developed in an exponential manner as a natural response to our information needs, regardless of space or time. Such information is mostly related to the context in which the user finds himself and usually dependent on his current needs. In the field of mobility, applications must cope with constant context changes in order to provide relevant information on demand. With current development strongly tackling the mobile connectivity domain, we can understand why the seminal equation

$$programming = coordination + computation \text{ [11]}$$

remains of great relevance. *Coordination* is thus an equal partner to *computation* in the *programming* process. This is even an essential element in the fields of distributed systems and concurrency. To be as generic as possible, coordination acts as a middleware layer between a consumer and a producer. It must place a demand to an entity able to solve it and afterwards ensures that an answer is returned. Departing from these two ideas we can see how *mobility* and *coordination* are key concepts for which proper mechanisms of interaction are required. As such, our objectives in this paper are to tackle two important needs: supporting context awareness and solving mobility issues in ad hoc networks related to the

* Corresponding author.
E-mail addresses: jean-marie.jacquet@unamur.be (J.-M. Jacquet), isabelle.linden@unamur.be (I. Linden), mihail.staicu@unamur.be (M.-O. Staicu).

http://dx.doi.org/10.1016/j.scico.2015.08.003

unpredictability of their connections and topology. We propose a solution in the form of *rules* spanning over one or more device data spaces (henceforth *blackboards*). As a first snapshot, these rules take the form

$$in(a, X),\ nin(b, Y) \ \longrightarrow\ in(c, W),\ nin(d, Z) \qquad (1)$$

with the intuitive meaning that as long as some information X is (physically) present on a blackboard a and some information Y is not (physically) present on blackboard b, then we can deduce that some information W is (physically) available on blackboard c and some information Z is no longer (physically) available on blackboard d. By providing two operational readings of such declarative rules, we aim at obtaining both of these needed functionalities. First, by means of a *forward reading* (inferred from applying as soon as possible the rule from left to right), we model the reaction to context changes occurring on the blackboards of mobile devices. Intuitively, departing from the previous rule, a blackboard context is defined in terms of information being present and/or absent and can model a user's specific need. For example, the rule

$$in(a, \langle location,\ Brussels\rangle),\ nin(a, \langle weather,\ sunny\rangle) \ \longrightarrow\ in(a, \langle museums\rangle)$$

states that if a tourist is in Brussels and the forecasted weather is not sunny, then he would like to visit available museums. Note that this information is made available in an eager fashion, as soon as it becomes available. Second, through a *backward reading* (inferred from using the rule from right to left), we represent the links between blackboards that would model the connection state of two devices. Let us assume the same tourist visiting Brussels with a mobile device and let us suppose the presence of city info posts that can make available a broad range of information consisting of weather data, nearby parking places, public transportation, etc. As he walks around the city, his mobile device roams between different such info posts and eventually connects to these when coming within connection range, which can depend on the wireless technology used for the connection. By exploiting the events raised at the networking level, a backward reading rule of the form $in(b, X) \longrightarrow in(a, X)$ is generated on the tourist's blackboard in order to define the connection state of his mobile device (containing the blackboard a) and the info post (containing the blackboard b). This rule acts as a logical link to be exploited when needed. It allows for the transient sharing of the two blackboards and raises the need to avoid physically transpiring the information between the blackboards. In conjunction with this rule, the previous forward reading rule can be resolved since the info post will provide the needed information about the weather.

As the careful reader will have observed from this informal presentation, the introduction of the rules brings up several issues to be addressed as to whether information needs to be removed or not, when this is operated and how eagerly rules have to be applied. Before presenting our design choices in Section 3, we first introduce the concept of multiple blackboard coordination through rules in Section 2. Then we provide in Section 4 a formal semantics of the language and establish in Section 5 that rules do increase the expressiveness of Linda-like coordination languages. Section 6 presents some technical and architectural details of our implementation, in particular on mobile devices. Section 7 compares our work with related work in the field of mobile coordination. Section 8 presents some related work in terms of reactivity and how our proposal is positioned with respect to these. Finally, Section 9 draws our conclusions and presents possible guides for future work.

This paper is an extension of a paper presented at FOCLASA 2012 [15]. It mainly differs from it by the formal semantics, not treated in [15], as well as by the expressiveness results of Section 5.

2. Blackboard coordination through rules

Reactivity in coordination languages relies on the basic idea of triggering an event when a predefined condition is met. The purpose is to enrich the capabilities of classical tuple spaces in order to transform them from simple tuple containers into context-aware entities, responsive to the interactions performed by the agents. Extensions have been explored, either by defining more complex triggering conditions or by taking a more complex set of actions. In ReSpecT [22], the conditions are represented either by the execution of a given primitive or by a fixed time stamp. Optionally, auditions guards may impose restrictions on the success of the execution, on the originating agent, on the originating tuple center or on time intervals in which the event occurred. As regards the reaction, this may be a combination of either primitives, observation predicates or computations. LIME [21] treats reactions as pieces of code that specify the actions to be executed when a defined tuple is found on the tuple space. MARS [7] monitors primitives executed on the tuple space and renders it more precise by specifying additional details on the tuple to be matched, the type of primitive and the identity of the agent. Further details on other models are presented in Section 8.

Our approach, based on the principle of reactivity, is used to model the behavior of the system due to context changes on the tuple space and to handle the execution of mobile agents in scenarios that involve physical mobility of the hosts. Following the ideas exposed in LIME [21], when two or more devices come into each-other's range, their tuple spaces are merged in terms of information that is publicly available. A hard to grasp notion is the one of "*close enough*" for two devices to be in range and thus to be connected. Likewise, the idea of merging blackboards, although intuitive, is actually not a clearcut one. Let us address these two points in turn. On the one hand, closeness in practical terms is to be defined by the presence of a physical communication link, be it wired or wireless, that would enable a communication channel between the devices. From an abstract perspective, our solution benefits from the events raised by the operating system of the device and captured by the middleware. On the other hand, merging the blackboards should not be seen straightforward in the sense of transferring tuples over the network, but may also be though as providing *logical links*, as it was defined in [4].