Future Generation Computer Systems 56 (2016) 124–139

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Logic-based modeling of information transfer in cyber–physical multi-agent systems

Christian Kroiß [a,*], Tomáš Bureš [b]

[a] Ludwig Maximilian University of Munich, Institute for Informatics, Munich, Germany

[b] Charles University in Prague, Faculty of Mathematics and Physics, Prague, Czech Republic

HIGHLIGHTS

- Logic-based modeling of information transfer in multi-agent systems.
- Considers stochastic nature of communication in the cyber–physical setting.
- Allows property specification with first-order time-bounded LTL.
- Can be used for statistical model checking.

ARTICLE INFO

Article history:
Received 7 February 2015
Received in revised form
7 September 2015
Accepted 11 September 2015
Available online 25 September 2015

Keywords:
Statistical model checking
cyber–physical systems
Situation calculus
Discrete event simulation

ABSTRACT

In modeling multi-agent systems, the structure of their communication is typically one of the most important aspects, especially for systems that strive toward self-organization or collaborative adaptation. Traditionally, such structures have often been described using logic-based approaches as they provide a formal foundation for many verification methods. However, these formalisms are typically not well suited to reflect the stochastic nature of communication in the cyber–physical setting. In particular, their level of abstraction is either too high to provide sufficient accuracy or too low to be practicable in more complex models. Therefore, we propose an extension of the logic-based modeling language SALMA, which we have introduced recently, that provides adequate high-level constructs for communication and data propagation, explicitly taking into account stochastic delays and errors. In combination with SALMA's tool support for simulation and statistical model checking, this creates a pragmatic approach for verification and validation of cyber–physical multi-agent systems.

1. Introduction

With SALMA (Simulation and Analysis of Logic-Based Multi-Agent Systems) [1], we have recently introduced an approach for modeling and analysis of multi-agent systems that is aimed to provide a lightweight solution for approximated verification through *statistical model checking* [2] with the system model still being grounded on a rigorous formal foundation. SALMA's modeling language is based on the well-established *situation calculus* [3], a first-order logic language for describing dynamical systems.

In this paper, we provide an extension of SALMA (and the situation calculus in general) to explicitly address one aspect that is particularly important for *cyber–physical* [4] multi-agent systems, namely the distributed gathering and transfer of information. Agents not only have to continuously sense their environment, but also share these readings with other agents, acquire information of others, and participate in coordination activities. In the cyber–physical context, these information transfer processes are subject to stochastic effects, e.g. due to sensor errors or unreliable communication channels. Furthermore, accuracy and timing of information transfer processes can strongly influence the behavior of the whole system. In particular, the efficacy of mechanisms for self-adaptation or optimization typically degrades when certain time-constraints are violated or the accuracy of sensors is insufficient.

Using pure logical formalisms like the basic situation calculus for describing such scenarios results in rather verbose and low-level representations that are not practicable in more complex cases. What is needed instead are high-level constructs that establish a bridge between the underlying logical semantics and the typical requirements for modeling information transfer in

* Corresponding author.
E-mail addresses: kroiss@pst.ifi.lmu.de (C. Kroiß), bures@d3s.mff.cuni.cz (T. Bureš).

http://dx.doi.org/10.1016/j.future.2015.09.013

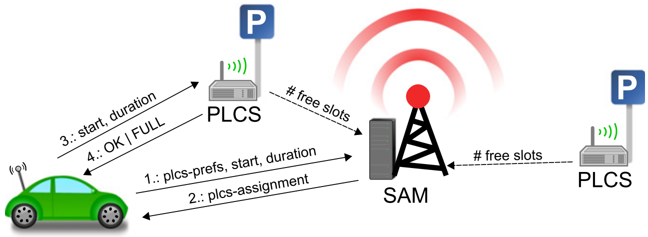

Fig. 1. Optimized parking lot assignment scenario.

multi-agent CPS. Although higher-level extensions on top of the situation calculus have been designed for related aspects like sensing and knowledge (e.g. [5]), there has, to our knowledge, not been a detailed reflection of information propagation in CPS in the context of the situation calculus.

We have therefore developed a generic model of information transfer that is based on a stochastic timed version of the situation calculus and allows capturing a wide range of effects that may be imposed on information transfer processes. Additionally, we have defined a set of macro-like abstractions for common information transfer scenarios within CPS, such as message passing or sensor data propagation. This creates a concise interface for the modeler that hides the stochastic details of information propagation but makes them fully accessible in simulation and verification.

In the following sections, we first set the picture by introducing an example from the e-mobility domain that will be used throughout the paper to demonstrate the developed concepts. We then shortly give some necessary background about the situation calculus that will be needed to understand the mechanisms described later. Then, in Section 4, we introduce the basics of the SALMA approach and formally define a core part of its simulation semantics in Section 5 as a foundation for the discussion of the information transfer model. The main contribution of this article starts in Section 6, where we introduce our generic model for information transfer and describe precisely its realization on the basis of the situation calculus. After that, in Section 7, we define several extensions to SALMA's modeling language that provide pragmatic abstractions for the information transfer model. This is continued in Section 8, where the focus is set on the use of SALMA for statistical model checking in the context of information transfer processes. As an evaluation of our approach, Section 9 discusses its application to the example that was introduced in the beginning. There, we show some experimental results and assess experiences regarding the benefits of our approach. Finally, we give an overview of related work before we end the paper with a short conclusion.

2. Example: optimized parking lot assignment

As a running example to illustrate our approach, we employ the e-mobility case-study of the ASCENS EU project[1] that has been described before, e.g., in [6]. The case study focuses on a scenario in which electric vehicles compete for parking lots with integrated charging stations (PLCS) in an urban area. The goal is to find an optimal assignment of PLCS to vehicles. Technically, the assignment is performed by an agent called super-autonomic manager (SAM) that coordinates a number of PLCS. The structure of the information transfer within this scenario is outlined in Fig. 1. The basic idea is that vehicles send *assignment requests* to the SAM, including a start time, a duration, and a list of preferred PLCS that is compiled by the vehicle's on-board computer. The SAM tries to find optimal suggestions for parking lot assignments, based on the knowledge about driver's intentions, and on occupancy information that is sent repeatedly by the PLCS.

True to the distributed CPS principle, all the agents (vehicles, PLCS, SAM) are autonomous and communicate via some wireless data transmission infrastructure like a VANET or 3G/4G network. This implies that neither transmission delays nor the possibility of errors can be neglected. However, timing clearly plays an important role in the scenario described above. First of all, the reservation service would simply not be accepted if the delay between reservation requests and reservation responses was too high. Also, the communication timing affects the convergence of the optimization, thus directly it influences the functionality of the distributed CPS.

3. Background: situation calculus

In this article, we show how to capture the timing aspects in situation calculus models, while maintaining a practical level of abstraction that focuses on the core "business-level" functionality. However, before doing so, we briefly summarize the main principles of the situation calculus in this section, so as to provide a necessary background for the rest of the paper.

The situation calculus [3] is a first-order logic language for modeling dynamic systems. Its foundation is based on the notion of *situations*, which can be seen as histories of the world resulting from performing *action sequences*. Actions can either be deliberately executed by agents or *exogenous*, i.e. external *events* caused by the environment. Situation terms are then formed recursively by the function $do(a, s)$ that denotes the execution of action a in situation s. Consequently, the term

$$do(a_n, do(a_{n-1}, do(\ldots, do(a_1, S_0)\ldots)))$$

stands for the situation that results when the action sequence $\langle a_1, \ldots, a_n\rangle$ is executed in the initial situation S_0.

The state of the world in a given situation is defined by the set of all *fluents*, which are situation-dependent predicates or functions. Since the models discussed here are meant to be used in *discrete event simulation*, time itself is simply modeled as an integer fluent named *time* that is increased with each simulation step. How other fluents are affected by *actions* and *events* is defined by *successor state axioms (SSAs)* that recursively relates the next situation to the current one. In fact, a situation calculus model has to contain one SSA for every fluent that define exactly when a *boolean (relational) fluent* is true, or when a *functional fluent* has a certain value. As a simple example, the following axiom states that a vehicle is driving in a situation $do(a, s)$ either when a start event occurred, or when no stop event occurred and the vehicle had been driving in the situation before.

$$driving(vehicle, do(a, s)) \equiv a = start(vehicle)$$
$$\lor\ (\neg(a = stop(vehicle)) \land driving(vehicle, s)).$$

Additionally, a situation calculus model also contains *precondition axioms* that define whether or not an action or event is possible in a given situation. In general, both the effects of actions and events, and also the occurrence of exogenous actions, are of stochastic nature. Consequently, simulation involves sampling from a set of probability distributions that the modeler can define as part of the simulation's configuration (cf. Section 7).

In SALMA, the situation calculus is used together with a forward reasoning technique called *progression* [3, chap. 9] that basically uses the successor state axioms to create a new *snapshot* of the world state and uses this as the initial situation for the next simulation step. In contrast to the original situation calculus reasoning method, *regression* [3, chap. 4.5], progression actually "forgets the past" and is therefore not suited for many theorem

[1] www.ascens-ist.eu.