Science of Computer Programming 187 (2020) 102345

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Multi-agent systems with virtual stigmergy ☆

Rocco De Nicola [a], Luca Di Stefano [b,*], Omar Inverso [b]

[a] IMT School for Advanced Studies, Lucca, Italy
[b] Gran Sasso Science Institute (GSSI), L'Aquila, Italy

ARTICLE INFO

Article history:
Received 12 February 2019
Received in revised form 21 October 2019
Accepted 28 October 2019
Available online 6 November 2019

Keywords:
Multi-agent systems
Stigmergic interaction
Emergent behavior
Attribute-based communication
Agent-based modeling

ABSTRACT

We introduce a simple language for multi-agent systems that lends itself to intuitive design of local specifications. Agents operate on (parts of) a decentralized data structure, the stigmergy, that contains their (partial) knowledge. Such knowledge is asynchronously propagated across local stigmergies. In this way, local changes may influence global behavior. The main novelty is that our interaction mechanism combines stigmergic interaction with attribute-based communication. Specific conditions for interaction can be expressed in the form of predicates over exposed features of the agents. Additionally, agents may access a global environment. After presenting the language, we show its expressiveness by considering some illustrative case studies. We also include preliminary results towards automated verification via a mechanizable symbolic encoding that enables us to exploit verification tools developed for mainstream languages.

1. Introduction

Multi-agent systems are collections of autonomous agents that operate according to some local rules and a limited mutual awareness. They are a convenient formalism for representing several classes of complex systems and can support formal reasoning about them. An issue that arises when considering a multi-agent system is how to determine whether a global property of interest emerges from the combination of the local behaviors of the different individual agents. The availability of a formal description of a multi-agent system allows one to apply automated verification techniques and can be instrumental for obtaining strong guarantees about its global behavior. Simulation-based approaches, on the other hand, may be more effective when dealing with larger multi-agent systems, due to the considerably large state spaces resulting from their distributed and asynchronous nature. Therefore, the two approaches should be considered complementary to each other.

In this paper, we introduce a language for describing multi-agent systems that lends itself to an intuitive design of local specifications and that can be used as the basis for automated analysis. The language, which we call LAbS for Language with Attribute-based Stigmergies, is simple yet versatile enough to model several interesting classes of systems. It combines stigmergic interaction [1,2] with attribute-based communication [3]. A key concept of the language is that of *virtual stigmergy*, a distributed data structure that can model global knowledge. Each agent operates only on his local copy of the stigmergy, that stores his own (partial) knowledge of the system. Individual knowledge is then asynchronously propagated across other local stigmergies. Thus, changes by an agent may indirectly affect the behavior of another.

☆ Work partially funded by MIUR project PRIN 2017FTXR7S *IT MATTERS* (Methods and Tools for Trustworthy Smart Systems).
* Corresponding author.
E-mail address: luca.distefano@gssi.it (L. Di Stefano).

https://doi.org/10.1016/j.scico.2019.102345
0167-6423/

In the originally proposed version of the virtual stigmergy [4], agents are concrete entities, each at a specific position in space, that can communicate across the stigmergy only if they are within a given distance from each other. To increase expressiveness, we generalise stigmergic interaction to arbitrary *predicates* over exposed features, referred to as *attributes*, of the agents. In fact, our language has no explicit concept of position for agents, and thus of neighborhood. An agent can have instead local attributes, and predicates over these attributes can express the conditions for two agents to be allowed to exchange knowledge. Movement is no longer seen as a specific action; an agent may update the attribute that encodes its position by performing a standard update action.

The generalisation of stigmergic interaction outlined above increases the flexibility of the language and allows to model a wider class of systems. However, it is still not sufficiently expressive to naturally model those classes of multi-agent systems where the global *environment* plays a crucial role [5,6]. To address this shortcoming, we extend our language with tailored primitives to explicitly model actions on the environment.

This work extends our original presentation of the language [7] in several ways. The new syntax and semantics support *multiple stigmergies* and improve the specification of situated systems. Multiple stigmergies allow us to naturally describe further interesting classes of systems, where agents can communicate in different ways. For instance, they can be used to directly model multi-robot systems where robots have multiple sensors and communication devices, and decide the equipment to use depending on specific environmental conditions. As for situated systems, environment variables may now directly occur in expressions and guards. This makes it easier to describe systems where agents also interact via the environment.

New case studies have been added to those related to *birds flocking*, *robots foraging* and *opinion formation* considered in [7]. To vindicate the flexibility of our language and its ability of expressing different interesting classes of systems, in this paper we also model *Boids* [8], and *population protocols* [9,10]. The former is a widely-used model of flocking behavior as observed in different classes of natural systems. It extends the flocking case study by allowing additional interaction strategies, for getting closer and avoiding collisions, and not only for moving in the same direction. The latter are a type of *gossip protocols* that rely on a distributed communication paradigm inspired by the spreading of epidemics and by the gossip phenomenon observed in social networks. For both classes of systems, we do provide experimental results about their automated analysis. Our modeling of *Boids* systems shows the benefits of adding multiple stigmergies to the language and, to the best of our knowledge, our work reports the first results about formal verification of such systems; previous investigations only exploited simulation-based techniques.

The rest of the paper is organized as follows. In Section 2 we present a revised version of the formal semantics of the core language, allowing us to define systems where agents interact indirectly through multiple stigmergies. In Section 3 we demonstrate the features of the language by modeling the *Boids* system, and include preliminary results about its automated analysis together with a discussion about the impact on verification of the different parameters of the specified system and of the used verification tools. In Section 4 we further enrich the language with environment-oriented primitives and show how LAbS can naturally model other classes of systems dealing with *robot foraging*, *opinion formation* and *gossiping*. In Section 5 we summarise our main achievements, compare our work with others, and suggest directions for future research.

2. The LAbS language

In this section we introduce LAbS (Language with Attribute-based Stigmergies), a language that has been designed to program multi-agent systems (MAS). The interaction mechanisms of LAbS are inspired by the specific form of stigmergic interaction originally proposed with the Buzz language [1] that we generalise by exploiting attribute-based communication as introduced in [3].

A key concept of LAbS is the *virtual stigmergy*, a distributed data structure that models the global knowledge of the system. Each agent maintains a local copy of (part of) this data structure, that contains his own (partial) knowledge of the system. We call these copies *local stigmergies*. An agent reads from and writes to his local stigmergy only. Knowledge is silently and asynchronously propagated across local stigmergies. This way, indirect agents interaction is achieved.

Formally, local stigmergies $L \in \mathcal{L}$ are partial functions that map keys to timestamped values: $\mathcal{L} = \mathcal{K}_L \hookrightarrow \mathcal{V} \times \mathbb{N}$, where $\mathcal{K}_L$ and $\mathcal{V}$ are the sets of allowed keys and values, respectively. We use natural numbers to represent timestamps. If $(x, v, t) \in L$, we say that v is the *value* of x and that t is its *timestamp* in the local stigmergy L. We refer to these as $value(L, x)$ and $time(L, x)$, respectively. We write $L(x) = \bot$ whenever $\forall v.\forall t.(x, v, t) \notin L$.

The operations on the stigmergy and the propagation mechanism are the following. When an agent *writes* a key-value pair into his local stigmergy, a timestamp is retrieved from a global clock and bound to the pair. If the local stigmergy contains an entry with the same key, it is replaced by the new one. The new data is then automatically (though asynchronously) propagated to neighboring agents. For agents in the neighborhood that already have a value bound to the same key but with a newer timestamp, the propagation has no effect; all the other agents update their local stigmergy, and in turn, propagate the new value. In the long run, this process allows information to be spread throughout the system. Conversely, each time an agent *reads* from its local stigmergy, a key confirmation request is sent to the neighborhood to confirm whether the data just accessed is up-to-date. This will, in turn, trigger the propagation of more recent information from the local stigmergies nearby, the update of any older entries, and again their propagation.