Information Sciences 230 (2013) 132–146

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Data mining agent conversations: A qualitative approach to multiagent systems analysis

Emilio Serrano [a,*], Michael Rovatsos [b], Juan A. Botía [a]

[a] Universidad de Murcia, Facultad de Informática, Campus Universitario de Espinardo, 30100 Murcia, Spain
[b] School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, United Kingdom

ARTICLE INFO

Article history:
Received 29 May 2012
Received in revised form 4 December 2012
Accepted 21 December 2012
Available online 8 January 2013

Keywords:
Agent communication language
Interaction protocol
Interaction analysis
Data mining
Agent-oriented software engineering

ABSTRACT

This paper presents a novel method for analysing the behaviour of multiagent systems on the basis of the semantically rich information provided by agent communication languages and interaction protocols specified at the knowledge level. More low-level communication mechanisms only allow for a *quantitative* analysis of the occurrence of message types, the frequency of message sequences, and the empirical distributions of parameter values. Quite differently, the semantics of languages and protocols in multiagent systems can help to extract *qualitative* properties of observed conversations among agents. This can be achieved by interpreting the logical constraints associated with protocol execution paths or individual messages as the *context* of an observed interaction, and using them as features of learning samples. The contexts "mined" from such analyses, or *context models*, can then be used for various tasks, e.g. for predicting others' future responses (useful when trying to make strategic communication decisions to achieve a particular outcome), to support ontological alignment (by comparing the properties of logical constraints attached to messages across participating agents), or to assess the trustworthiness of agents (by verifying the logical coherence of their behaviour). This paper details a formal approach that describes our notion of context models in multiagent conversations, an implementation of this approach in a practical tool for mining qualitative context models, and experimental results to illustrate its use and utility.

1. Introduction

One of the cornerstones of agent technology is the loose coupling between agents achieved by introducing standardised high-level agent communication languages (ACLs, e.g. FIPA-ACL [6]) and interaction protocols. As opposed to low-level interaction mechanisms for computer systems (like those used in traditional distributed computing), these advanced languages and protocols attempt to capture shared meaning for messages exchanged in multiagent systems. This helps to ensure that, despite the heterogeneity among individual agents who cannot observe each other's internal states, some level of interoperability can be achieved in practice, so that large-scale open multiagent systems can be implemented in the real world.

At the same time, when it comes to *analysing* agent-based systems, the openness of these systems limits the available data to what goes on *among* agents (i.e. observations of message exchanges, at least if we assume that the observer does not have access to all internal details of all agents in the system [15]). However, the structure and "knowledge-level" assumptions captured in ACLs and interaction protocols is semantically rich and can be used to partially compensate for the loss of transparency caused by agent-level encapsulation, which makes the mental states of an agent opaque to others.

* Corresponding author. Tel.: +34 968 367882.
E-mail addresses: emilioserra@um.es (E. Serrano), mrovatso@inf.ed.ac.uk (M. Rovatsos), juanbot@um.es (J.A. Botía).

http://dx.doi.org/10.1016/j.ins.2012.12.019

As an example of this, consider a message $\texttt{inform}(A, B, X)$ with the usual meaning that agent A informs B of a fact X (where X is taken from some domain ontology or "content language"). The use of this message type is usually tied to preconditions like $(Bel\ AX)$ stating that A in fact believes X to be true. While B is unable to verify whether this is *actually* the case (or A is lying/has a different interpretation of the *Bel* modality), the use of the message entitles B to operate under the assumption that $(Bel\ AX)$ is true for A. For example, if B contested X, it would be unreasonable for a protocol to allow A to state that she never claimed X. Therefore, at a *pragmatic* level, any semantic "annotations" (pre- and post-conditions) of messages that an agent is uttering can be used as assumptions about that agent's mental state (or, e.g. in commitment-based semantics [7], about their perception of a social state).

Quite surprisingly, this aspect of data analysis has been overlooked in the existing literature (see Section 2). Existing approaches remain at the *quantitative* level, i.e. any measurements they take are based on assessing the observed values of some attributes of the interaction. A binary distinction between "interaction was successful or not" is often employed, sometimes also a measurement of the quality of different attributes along numerical scales, e.g. speed, price, reliability, etc. While in non-agent scenarios this may be the only kind of data that is available, if one focuses only on quantitative analysis, a lot of additional structural information goes "to waste" in a sense when considering ACL-based multiagent system interactions.

The contribution of this paper is to fill this gap by exploring the use of data mining techniques over semantically rich interaction protocols defined using typical ACLs. By using semantic elements of protocols as features of interaction traces, which are available as data samples from past interactions, we can inductively derive what we call *context models*, i.e. logical theories that capture regularities in previously observed interactions. Note that the protocol definition or *protocol model* needs to include semantic annotations to allow the approach presented in this paper to build an effective context model automatically, and, in this paper, we will make the assumption that protocol specifications include such annotations.

Context models, which essentially capture generalised information about the conditions under which a protocol reaches a certain outcome, can be used for various purposes: (1) to make predictions about future behaviour (e.g. under what circumstances a peer is likely to deliver a product of reasonable quality); (2) to infer the definitions other agents use when validating logical constraints during an interaction (e.g. when acceptance of a certain type of offer indicates the range of a variable X for which the predicate $acceptable(X)$ holds true for an agent); and (3) to analyse the reliability and trustworthiness of agents based on the logical coherence of their utterances (e.g. if an agent has been observed to suggest that both $P(o)$ and $\neg P(o)$ are true of the same object o). Moreover, by simply grouping the data from interactions with several agents together and analysing it with a data mining algorithm, it is easy to generalise over individual "theories" held by them to develop a more global picture of the views held by an entire set of agents.

The remainder of the paper is structured as follows: After reviewing related work in Section 2, we define the formal machinery that is needed to develop qualitative data mining methods for interaction protocols in Section 3. Sections 4 and 5 discuss an implemented system for qualitative data mining over interaction protocols, and empirical results obtained in a case study, respectively. Section 6 concludes.

2. Related work

Most existing work on run-time agent systems analysis typically addresses the testing, debugging, validation and verification of these systems [17].

Regarding the analysis of *single* agents, the literature is heavily influenced by the BDI (Beliefs, Desires, Intentions) model. Sudeikat et al. [20] propose to check the consistency of beliefs, the achievement of goals and the process of plan execution in the JADEX agent platform. A similar analysis is enabled by the Agent Factory framework [5] which offers run-time views of the agent's internal state through its agent viewing tool. The INGENIAS platform [9] also provides run-time mental state inspection and testing besides static checking of the specification.

As far as *interaction* analysis is concerned, many agent development frameworks offer graphical tools that can be used for visual inspection, and ways of filtering messages to reduce the amount of data shown (e.g. [11]). As the amount of data increases and manual inspection becomes impractical, automated analysis methods are needed. Here, existing approaches focus on verification of observed interactions against a pre-defined model, and on manual debugging based on identification of incorrect protocol executions. Padgham et al. [13], for example, propose methods for translating protocols specified in AUML to a Petri-net based formalism, and to report errors when protocol specifications are not followed correctly during execution. In a similar vein, Chopra et al. [4] have recently formalised the semantic relationship between agents and protocols in order to verify whether a protocol supports the achievement of particular agent goals and whether the specification of the participating agents supports the satisfaction of particular commitments. These contributions provide verification mechanisms that can be used by the designer at runtime. With a stronger focus on design-time analysis, Wooldridge et al. [22] propose a language for the design and automatic verification of multiagent systems called MABLE. This imperative programming language uses the SPIN model checker to automatically verify properties of multiagent systems.

Despite their important contributions, the main limitation of these approaches is that they can only verify the correctness of protocol executions based on observed interactions, but they cannot derive any *additional* knowledge about the emergent behaviour of the system apart from whether agents are behaving correctly or not.

There is also work that focuses on measuring the performance of a multiagent system in terms of the frequency of message types or message patterns [15]. These measurements are used to identify problems in the system, or for deriving