Mining Qualitative Context Models from Multiagent Interactions

(Extended Abstract)

Emilio Serrano
Universidad de Murcia
emilioserra@um.es

Michael Rovatsos
University of Edinburgh
michael.rovatsos@ed.ac.uk

Juan Botia
Universidad de Murcia
juanbot@um.es

ABSTRACT

We present 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. Contrary to analysis methods that rely on observing more low-level patterns of behaviour [3, 4], our method is based on exploiting the semantics. These languages and protocols which can be used to extract *qualitative* properties of observed interactions. This can be achieved by interpreting the logical constraints associated with protocol execution paths or individual messages as models of the *context* of an observed interaction, and using them as features of learning samples.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Theory, Design

Keywords

Agent communication languages, interaction protocols, interaction analysis, data mining, agent-oriented software engineering

1. INTRODUCTION

Consider a message `inform`(A, B, X) with the usual meaning that agent A informs B of a fact X. Use of this message type is usually tied to preconditions like $(Bel\ A\ \phi)$ stating that A in fact believes ϕ 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* predicate or of statement ϕ), use of the message entitles B to operate under the assumption that $(Bel\ A\ \phi)$ is true for A. For example, if B contested ϕ, it would be unreasonable for a protocol to allow A to state that she never claimed ϕ. So, at a *pragmatic* level, any semantic "annotations" (pre- and post-conditions) of messages that an agent is uttering can be used as assumptions about the former agent's mental state (or, e.g. in commitment-based semantics, about their perception of a social state).

By using semantic elements of protocols as features of interaction traces, which are available as data samples from past interactions, we can inductively derive *context models* i.e. logical theories that capture regularities in previously observed interactions. These 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, (2) to infer the definitions other agents apply when validating logical constraints during an interaction, and (3) to analyse the reliability and trustworthiness of agents based on the logical coherence of their utterances. Surprisingly, no previous work has addressed this potential use of semantic annotations of protocols, except some recent work in the area of ontology mapping [1, 2]. However, even these contributions only deal with ontological conflicts, and not with more general emergent properties of interactions.

Cite as: Mining Qualitative Context Models from Multiagent Interactions (Extended Abstract), Author(s), *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1215-1216.

2. FORMAL FRAMEWORK

We represent protocols in a very general way as graphs whose nodes are speech-act like messages placeholders, and whose edges define transitions among messages that give rise to message sequences specified as admissible according to the protocol. These edges will be labelled with logical constraints, i.e. formulas that all agents in the system are able to verify, and these act as guards on a given transition, so that the message corresponding to a child node can only be sent if the constraint(s) along its incoming edge from the parent node (the message just observed) can be satisfied.

We define a *protocol model* as a graph $G = (V, E)$ where each node $v \in V$ is labelled with a message $m(v) = q(X, Y, Z)$ with performative q (a string) and sender / receiver / content variables X, Y, and Z, and each edge is labelled with a (conjunctive) list of (say, n) constraints

$$c(e) = \{c_1(t_1, \ldots, t_k), \ldots, c_n(t_1, \ldots t_{k_n})\}$$

where each constraint $c_i(\ldots)$ has arity k_i, head c_i and arguments t_j which may contain constants, functions or variables (in general the label of an edge could be an arbitrary formula $\phi \in \mathcal{L}$ of a logical language $\mathcal{L}$). All variables that occur in such constraints are implicitly universally quantified. We also assume that all outgoing edges of a node result in messages with distinct performatives, i.e. for all $(v, v'), (v, v'') \in E$ $(m(v') = q(\ldots) \wedge m(v'') = q(\ldots)) \Rightarrow v' = v''$ so that each observed message sequence corresponds to (at most) one path in G by virtue of its performatives.

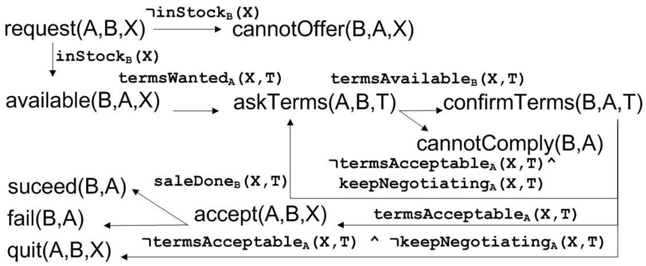

Figure 1: A simple negotiation protocol model.

Figure 1 shows an example protocol model in this generic format for illustration purposes. This figure presents a simple negotiation protocol model: A requests X, the initial response from B depends on availability; if X is available, A and B go through an iterative process of negotiating the terms for the purchase, depending on the *keepNegotiating*, *termsAcceptable*, and *termsAvailable* predicates; in case of acceptance (which implies payment), B may succeed or fail in delivering the product. Edge constraints are annotated with the variable representing the agent that has to validate them.

The *semantics* of a protocol model G can be defined based on the pair $\langle \pi, \theta \rangle$ which returns the path and variable substitution that the message sequence $\mathbf{m}$ corresponds to in protocol model G. With this, we can define the *context* of $\mathbf{m}$ as $c(G, \langle m_1, \ldots, m_n \rangle) = \bigwedge_{i=1}^{n-1} c(e_i)\theta$ where $G(\mathbf{m}) = \langle \pi, \theta \rangle$. The basis of our analysis is the assumption that for any observed message sequence $\mathbf{m}$, the conjunction of edge constraints described by the context $c(G, \langle m_1, \ldots, m_n \rangle)$ was logically true at the time of the interaction.

Consider a protocol model G, and message sequences $\mathbf{m}$ obtained from past executions of G. Any such sequence can be translated to a pair $G(\mathbf{m}) = \langle \pi, \theta \rangle$ as defined above. Assuming that a set of such substitution-annotated paths are used as a training data, the extension proposed here is to *augment* the learning data by the logical context of the data samples, i.e. to include the logical formula $c(G, \mathbf{m})$ in the data samples, which can be directly inferred using the logical constraints provided by the definition of G. In other words, we view *qualitative* protocol mining as an informed version of data-driven interaction analysis where the background knowledge of *context* within which communication occurs is used to extract "richer" information about what is happening in a given system.

Due to the nature of multiagent interaction protocols, additional design decisions have to be made to deal with different agents, paths, variables, and loops before standard data mining machinery can be used (we omit the details of these issues for lack of space).

3. CASE STUDY

To illustrate the usefulness of our approach, we have analysed data generated in a car selling domain, where agents negotiate over cars using the protocol shown in figure 1. We experimented with two open source implementations of data mining techniques, the *J48* decision tree algorithm and the *NNge* classification-rule based algorithm, to show that our method does not depend on the use of a specific learning algorithm.

For the purposes of this case study, we assume that a single seller (S) is analysing the system evolution from its local point of view. In converting raw sequences of message exchanges to training data samples, we make the following choices: The seller (in role B), unaware of the decision-making rules of a set of 10 customers (in role A), performs the analysis, thus the learning input is restricted to the messages (nodes) and constraints (edges) of the customers. As far as variables occurring in constraints are concerned, we uniformly record all attributes contained in "terms" descriptions T, including a "?" (unknown) value for those not mentioned in a given execution trace. The seller tries to learn a model for the general outcome of the protocol (*S*uccessful, *N*eutral or *F*ailure). The table in figure 2 shows results for 10^3 to 10^5 negotiations where: *nn* is the number of negotiations, *time* is the time in seconds required to build the model, *cci* is the percentage of correctly classified instances (evaluated using 10-fold stratified cross-validation), *mae* is the mean absolute error and *rae* is the relative absolute error.

J48	**nn**	**time**	**cci**	**mae**	**rae**
	1.E3	0.05	83.4%	0.16	38.34%
	1.E4	0.48	97.58%	0.04	11.21%
	1.E5	5.09	99.96%	0.004	1.1%
Nnge	**nn**	**time**	**cci**	**mae**	**rae**
	1.E3	0.1	86.8%	0.08	19.91%
	1.E4	0.66	89.03%	0.07	16.6%
	1.E5	17.39	93.53%	0.04	9.79%

Figure 2: Experiment results.

These experiments, in which the protocol mining algorithms were able to accurately reconstruct the actual decision rules used by the customers, demonstrate that good models to predict the outcome of a protocol can be quickly built from the context of concrete executions of that protocol. They hint at the potential analyses that can be conducted and illustrate the usefulness of qualitative protocol mining in real-world scenarios[1].

4. REFERENCES

[1] M. Atencia and W. M. Schorlemmer. I-SSA: Interaction-Situated Semantic Alignment. In *OTM'08*, volume 5331 of *Lecture Notes in Computer Science*, pages 445–455. Springer, 2008.

[2] P. Besana and D. Robertson. Probabilistic Dialogue Models for Dynamic Ontology Mapping. In *URSW'08*, volume 5327 of *Lecture Notes in Computer Science*, pages 41–51. Springer, 2008.

[3] M. Nickles, M. Rovatsos, and G. Weiss. Expectation-oriented modeling. *Engineering Applications of Artificial Intelligence*, 18(8):891–918, 2005.

[4] E. Serrano, J. J. Gómez-Sanz, J. A. Botia, and J. Pavón. Intelligent data analysis applied to debug complex software systems. *Neurocomputing*, 72(13-15):2785 – 2795, 2009.

[1]Acknowledgments: This research work is supported by the Spanish Ministry of Science and Innovation under the grant AP2007-04080 and in the scope of the Research Projects TSI-020302-2010-129, TRA2009-0141 and through the Fundación Séneca within the Program 04552/GERM/06.