On Enactability of Agent Interaction Protocols: Towards a Unified Approach

Extended Abstract

Angelo Ferrando
Liverpool University
Liverpool, United Kingdom
angelo.ferrando@liverpool.ac.uk

Michael Winikoff
University of Otago
Dunedin, New Zealand
michael.winikoff@otago.ac.nz

Stephen Cranefield
University of Otago
Dunedin, New Zealand
stephen.cranefield@otago.ac.nz

Frank Dignum
Umeå University
Umeå, Sweden
frank.dignum@umu.se

Viviana Mascardi
University of Genova
Genova, Italy
viviana.mascardi@unige.it

ABSTRACT

Interactions between agents are usually designed from a global viewpoint. However, the implementation of a multi-agent interaction is distributed. This difference can introduce problems. For instance, it is possible to specify protocols from a global viewpoint that cannot be implemented as a collection of individual agents. This leads naturally to the question of whether a given (global) protocol is *enactable*. We consider this question in a powerful setting (trace expressions), considering a range of message ordering interpretations (specifying what it means to say that an interaction step occurs before another), and a range of possible constraints on the semantics of message delivery, corresponding to different properties of the underlying communication middleware.

KEYWORDS

Agent Interaction Protocols, Enactability, Enforceability, Implementability, Realizability, Projectability, Trace Expressions

ACM Reference Format:

Angelo Ferrando, Michael Winikoff, Stephen Cranefield, Frank Dignum, and Viviana Mascardi. 2019. On Enactability of Agent Interaction Protocols: Towards a Unified Approach. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Agent Interaction Protocols (AIP) are formal or informal specifications describing the communicative behaviour of heterogeneous and distributed agents inside a multi-agent system (MAS). These global protocols[1] unambiguously denote which interactions are allowed, when they are allowed, and in which order. They are global because they model the entire MAS from a high-level perspective and not from the point of view of each single participant. This level of abstraction gives the software engineer freedom of choice on how to implement the protocols and how to "enforce" the interaction order. Consider for example the following AIP involving three different agents *Alice*, *Bob* and *Carol*:

$$AIP_1 = Alice \stackrel{M1}{\Longrightarrow} Bob \;\cdot\; Alice \stackrel{M2}{\Longrightarrow} Carol$$

where $a1 \stackrel{M}{\Longrightarrow} a2$ denotes the interaction of $a1$ and $a2$ exchanging the message M and "$\cdot$" denotes interaction concatenation. This is a simple example of a possible formal definition of a global interaction protocol. The constraint on the message ordering is clear: $M1$ must occur before $M2$. Even though this is a well-stated constraint from a high-level viewpoint, how can it be enforced by the involved parties? Or, from a more practical viewpoint, how can the involved parties implement the protocol? If we consider AIP_1, the constraint that $M1$ must occur before $M2$ can be enforced in four different ways [9]: *Alice* must send $M1$ before sending $M2$ (denoted "SS"), *Alice* must send $M1$ before *Carol* can receive $M2$ (SR), *Bob* must receive $M1$ before *Alice* can send $M2$ (RS), or finally, *Bob* must receive $M1$ before *Carol* can receive $M2$ (RR). Of these four interpretations, if we consider the first one, it can be easily enforced by the involved parties, because *Alice* enforces the sending order (she is the sender for both the messages). Instead, if we consider the third interpretation, how can *Alice* know that *Bob* has already received $M1$ to correctly send $M2$? She can not. This means that, depending on the message ordering we choose for interpreting the protocol, the latter may or may not be enforceable. Naturally, by knowing the communication model we could have more guarantees on how the messages are delivered, consequently changing the outcome.

This kind of issue is not new in the field and various authors use different terms for global protocols that can be enforced: conformant [16], enforceable [2, 8], enactable [9], implementable [20], projectable [5, 14], realizable [18, 21]. The concept behind these names is however the same: by executing the localised versions of the protocol implemented by each participant, the global protocol behaviour is obtained, with no additional communication. We will use the term *enactability* [9] to denote this property. However, despite the large amount of work on enactability, there is no existing work that considers both the intended *message ordering* and the *communication model* of the infrastructure in which the agents will be implemented, that recognises the need to use a *decision structure* to enforce consistent choices, and that provides an

[1] These are modelled using a range of formalisms, including global types [6], Petri nets [17], WS-CDL [22], AUML [15], statecharts [13], and causal logic [12].

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

implementation for checking protocol enactability. Together, these are the innovative and original features of our contribution[2].

2 RESULTS AND DISCUSSION

Our approach starts from the definition of a global AIP τ – using the trace expression formalism [1] – and then following these steps[3].

Choose the Message Ordering Interpretation. We choose a message ordering interpretation *moi* (based on [9, 16]) and we formalise the semantics $[\![\tau]\!]_{moi}$ which is a variant of the trace expression semantics $[\![\tau]\!]$ that is defined in terms of *events* (sending/receiving of messages) rather than interactions. The possible sequences of events are constrained: given a situation where τ specifies that $M1$ must occur before $M2$, we constrain the sequence of events with the appropriate constraint on events corresponding to the selected *moi*. For example, given the *moi* that M_1 before M_2 means that M_1 must be sent before M_2 is sent (SS), we apply that constraint.

Identify the Communication Model. We identify the communication model *CM* (based on [7] with a standard synchronous communication model in addition) used by our MAS, and we formalise its semantics by defining the corresponding language of event traces that incorporates the appropriate restrictions, ruling out event sequences that violate the communication model *CM*. We then intersect this language with the set of traces generated by $[\![\tau]\!]_{moi}$, obtaining the new semantics for global API τ: $[\![\tau]\!]^{CM}_{moi}$.

Distribute the global protocol. Supposing that τ involves agents $a_1, \cdots, a_n$, we define the *distribution* of τ, denoted $\ulcorner\tau\urcorner$, as $\ulcorner\tau\urcorner = \tau^{a_1} \| \cdots \| \tau^{a_n}$. Here, τ^A denotes the projection[4] of τ onto agent A and the $\|$ operator defines parallelism between the projections[5]. To generate the set of traces recognized by $\ulcorner\tau\urcorner$ we need to define a *decision structure* $d(\tau)$. The heart of the issue is that the trace expression notation offers a choice operator ($\vee$), which is adequate for global protocols. However, for local protocols it is important to be able to distinguish between a choice that represents a free (local) choice, and a choice that is forced by earlier choices. Using the decision structure, we can define the semantics for the distributed protocol $[\![\tau]\!]_{dist} = \bigcup_{dt \in d(\tau)} [\![\tau^{a_1} \| \cdots \| \tau^{a_n}]\!]^{dt}$, where $[\![\tau]\!]^{dt}$ is the standard semantics constrained[6] with dt. As before, we denote the intersection with the language denoted by *CM* as $[\![\tau]\!]^{CM}_{dist}$.

Finally, we have everything we need to define the notion of *weak* and *strong enactability* for τ.

Definition 2.1 (Strongly/Weakly Enactable). Let τ be an interaction protocol, $\{a_1, a_2, \cdots, a_n\}$ the set of agents involved in τ, *moi* a message ordering interpretation and *CM* a communication model. We say that τ is strongly (or weakly) enactable for *moi* semantics in model *CM* if and only if the distribution of τ through projection on its agents $\{a_1, a_2, \cdots, a_n\}$ recognizes the same (or, respectively, a subset of) traces recognized by τ. Formally:

$$enact(\tau)^{CM}_{moi} \quad \text{iff} \quad [\![\tau]\!]^{CM}_{\text{dist}} = [\![\tau]\!]^{CM}_{moi}$$
$$weak\_enact(\tau)^{CM}_{moi} \quad \text{iff} \quad [\![\tau]\!]^{CM}_{\text{dist}} \subseteq [\![\tau]\!]^{CM}_{moi}$$

[2] For details, see the full version of this paper [11]
[3] A more detailed explanation can be found in: https://arxiv.org/abs/1902.01131.
[4] The projection retains only those aspects of the protocol that are relevant for the agent.
[5] Each projection represents an autonomous behaviour for a single agent.
[6] The decision structure influences the interpretation of the choice operator.

The table below shows the results of applying this definition to AIP_1, with different message ordering interpretations, and different communication models, from the strictest (CM1) to least strict (CM6)[7]. This table, in which ✔and (✔) denote strongly and weakly enactable respectively, has been automatically generated by our prototype implementation in Haskell[8].

The results in the table show that all of the four message ordering interpretations can be implemented for AIP_1, but some require quite strict guarantees from the communication middleware (e.g. for RS we need to have essentially synchronous communication). For this protocol, where both messages are sent by the same agent, the SS message ordering can be enforced with any communication model. The SR *moi* shows where weak enactability is useful: in this situation the distributed protocol cannot enforce exactly the desired constraints of the global protocol, but it is possible to enforce *stricter* constraints. For AIP_1 and SR, the desired constraint is that Carol receives M_2 after Alice sends M_1. The distributed protocol cannot enforce this, but it can enforce the stronger constraint that M_2 is sent (and therefore also received) after Alice sends M_1.

$Alice \overset{M1}{\Longrightarrow} Bob \cdot Alice \overset{M2}{\Longrightarrow} Carol$				
CM	RS	RR	SS	SR
CM1 (RSC)	✔	✔	✔	✔
CM2 (FIFO n-n)	✘	✔	✔	(✔)
CM3 (FIFO 1-n)	✘	✔	✔	(✔)
CM4 (FIFO n-1)	✘	✘	✔	(✔)
CM5 (causal)	✘	✘	✔	(✔)
CM6 (fully async)	✘	✘	✔	(✔)

In the future, we will address both theoretical and practical issues. On the theoretical side, we will carry out a systematic analysis of the relationships between the communication model and the message ordering interpretation, to identify those combinations that provide some guarantees by design. We will also consider the relationship between enactability and distributed monitorability [10], since the two notions are related.

On the practical part, we plan to improve our working prototype to provide a useful tool to assess protocols for enactability. Apart from providing a user-friendly interface, a key issue to address will be to provide a way to isolate the part of a non-enactable protocol that makes it non-enactable. To stress-test the prototype and assess its performance from a qualitative and quantitative viewpoint we plan to create a library of interaction protocols known to be "problematic" w.r.t. enactability, and perform systematic experiments.

Finally, this work highlighted the need to characterise existing agent infrastructures such as Jade [3], Jason [4] and Jadex [19] in terms of the communication model they support. This would allow us to state whether a protocol is enactable on a given infrastructure, strengthening the potential of our proposal to be exploited in real applications.

ACKNOWLEDGMENTS

The first author was supported by UK Research and Innovation, and EPSRC Hubs for Robotics and AI in Hazardous Environments: EP/R026173 (ORCA), and EP/R026084 (RAIN).

[7] RSC = Realizable with Synchronous Communication: a send of a message can only be followed (immediately) by receiving that message. FIFO n-n = messages are globally ordered; FIFO 1-n = messages from a sender are received in the order they were sent. FIFO n-1 = messages to a recipient are received in the order they were sent. Causal = messages are delivered according to the causality of their emission.
[8] ~300 LOC, available on the web at: http://enactability.altervista.org/

REFERENCES

[1] Davide Ancona, Angelo Ferrando, and Viviana Mascardi. 2016. Comparing Trace Expressions and Linear Temporal Logic for Runtime Verification. In *Theory and Practice of Formal Methods (LNCS)*, Vol. 9660. Springer, 47–64.

[2] Marco Autili and Massimo Tivoli. 2014. Distributed Enforcement of Service Choreographies. In *Proceedings of the 13th International Workshop on Foundations of Coordination Languages and Self-Adaptive Systems, FOCLASA 2014, Rome, Italy, 6th September 2014. (EPTCS)*, Javier Cámara and José Proença (Eds.), Vol. 175. 18–35. https://doi.org/10.4204/EPTCS.175.2

[3] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. 2007. *Developing Multi-Agent Systems with JADE.* Wiley.

[4] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. 2007. *Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology).* John Wiley & Sons.

[5] Marco Carbone, Kohei Honda, and Nobuko Yoshida. 2007. Structured Communication-Centred Programming for Web Services. In *Programming Languages and Systems, 16th European Symposium on Programming, ESOP 2007, Held as Part of the Joint European Conferences on Theory and Practics of Software, ETAPS 2007, Braga, Portugal, March 24 - April 1, 2007, Proceedings (Lecture Notes in Computer Science)*, Rocco De Nicola (Ed.), Vol. 4421. Springer, 2–17. https://doi.org/10.1007/978-3-540-71316-6_2

[6] Giuseppe Castagna, Mariangiola Dezani-Ciancaglini, and Luca Padovani. 2011. On Global Types and Multi-party Sessions. In *Formal Techniques for Distributed Systems - Joint 13th IFIP WG 6.1 International Conference, FMOODS 2011, and 31st IFIP WG 6.1 International Conference, FORTE 2011, Reykjavik, Iceland, June 6-9, 2011. Proceedings (Lecture Notes in Computer Science)*, Roberto Bruni and Jürgen Dingel (Eds.), Vol. 6722. Springer, 1–28. https://doi.org/10.1007/978-3-642-21461-5_1

[7] Florent Chevrou, Aurélie Hurault, and Philippe Quéinnec. 2016. On the diversity of asynchronous communication. *Formal Aspects of Computing* 28, 5 (2016), 847–879. https://doi.org/10.1007/s00165-016-0379-x

[8] Gero Decker and Mathias Weske. 2007. Local Enforceability in Interaction Petri Nets. In *Business Process Management, 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007, Proceedings (Lecture Notes in Computer Science)*, Gustavo Alonso, Peter Dadam, and Michael Rosemann (Eds.), Vol. 4714. Springer, 305–319. https://doi.org/10.1007/978-3-540-75183-0_22

[9] Nirmit Desai and Munindar P. Singh. 2008. On the Enactability of Business Protocols. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, Dieter Fox and Carla P. Gomes (Eds.). AAAI Press, CA, USA, 1126–1131. http://www.aaai.org/Library/AAAI/2008/aaai08-178.php

[10] Angelo Ferrando, Davide Ancona, and Viviana Mascardi. 2017. Decentralizing MAS Monitoring with DecAMon. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, Kate Larson, Michael Winikoff, Sanmay Das, and Edmund H. Durfee (Eds.). ACM, 239–248. http://dl.acm.org/citation.cfm?id=3091164

[11] Angelo Ferrando, Michael Winikoff, Stephen Cranefield, Frank Dignum, and Viviana Mascardi. 2019. *On Enactability and Monitorability of Agent Interaction Protocols.* arXiv e-print 1902.01131.

[12] Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner. 2004. Nonmonotonic causal theories. *Artificial Intelligence* 153, 1-2 (2004), 49–104. https://doi.org/10.1016/j.artint.2002.12.001

[13] David Harel. 1987. Statecharts: a visual formalism for complex systems. *Science of Computer Programming* 8, 3 (1987), 231 – 274. https://doi.org/10.1016/0167-6423(87)90035-9

[14] Kohei Honda, Nobuko Yoshida, and Marco Carbone. 2008. Multiparty asynchronous session types. In *Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2008, San Francisco, California, USA, January 7-12, 2008*, George C. Necula and Philip Wadler (Eds.). ACM, 273–284. https://doi.org/10.1145/1328438.1328472

[15] Marc-Philippe Huget and James Odell. 2005. Representing Agent Interaction Protocols with Agent UML. In *Agent-Oriented Software Engineering V: 5th International Workshop, AOSE 2004, Revised Selected Papers*, James Odell, Paolo Giorgini, and Jörg P. Müller (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 16–30. https://doi.org/10.1007/978-3-540-30578-1_2

[16] Ivan Lanese, Claudio Guidi, Fabrizio Montesi, and Gianluigi Zavattaro. 2008. Bridging the Gap between Interaction- and Process-Oriented Choreographies. In *Sixth IEEE International Conference on Software Engineering and Formal Methods, SEFM 2008, Cape Town, South Africa, 10-14 November 2008*, Antonio Cerone and Stefan Gruner (Eds.). IEEE Computer Society, CA, USA, 323–332. https://doi.org/10.1109/SEFM.2008.11

[17] James L. Peterson. 1977. Petri Nets. *Comput. Surveys* 9, 3 (Sept. 1977), 223–252. https://doi.org/10.1145/356698.356702

[18] Pascal Poizat and Gwen Salaün. 2012. Checking the Realizability of BPMN 2.0 Choreographies. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12).* ACM, New York, NY, USA, 1927–1934. https://doi.org/10.1145/2245276.2232095

[19] Alexander Pokahr, Lars Braubach, and Winfried Lamersdorf. 2005. Jadex: A BDI Reasoning Engine. In *Multi-Agent Programming: Languages, Platforms and Applications*, Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah Seghrouchni (Eds.). Springer US, Boston, MA, 149–174. https://doi.org/10.1007/0-387-26350-0_6

[20] Zongyan Qiu, Xiangpeng Zhao, Chao Cai, and Hongli Yang. 2007. Towards the theoretical foundation of choreography. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy (Eds.). ACM, 973–982. https://doi.org/10.1145/1242572.1242704

[21] Gwen Salaün, Tevfik Bultan, and Nima Roohi. 2012. Realizability of Choreographies Using Process Algebra Encodings. *IEEE Transactions on Services Computing* 5, 3 (2012), 290–304. https://doi.org/10.1109/TSC.2011.9

[22] W3C. 2005. Web Services Choreography Description Language Version 1.0. (2005). https://www.w3.org/TR/ws-cdl-10/.