Towards an Enthymeme-Based Communication Framework

Extended Abstract

Alison R. Panisson
Department of Computing – UFSC
Araranguá, Brazil
alison.panisson@ufsc.br

Peter McBurney
Department of Informatics – KCL
London, UK
peter.mcburney@kcl.uk

Rafael H. Bordini
School of Technology – PUCRS
Porto Alegre, Brazil
rafael.bordini@pucrs.br

ABSTRACT

In this work, we give an operational semantics for speech acts that BDI agents can use to communicate enthymemes. The approach uses argumentation schemes as common organisational knowledge to guide the construction of enthymemes by the proponents of arguments. Such schemes are also used to guide the reconstruction of the intended argument by the recipients of such enthymemes.

KEYWORDS

Mulit-agent systems, Argumentation, Enthymemes

ACM Reference Format:
Alison R. Panisson, Peter McBurney, and Rafael H. Bordini. 2022. Towards an Enthymeme-Based Communication Framework: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

Enthymemes are arguments in which one or more statements (which are part of the argument) are not explicitly stated [17]. They represent more realistic arguments, in the sense that arguments made by humans usually do not have enough explicitly stated premises for the entailment of the claim. This is because there is common knowledge that can be assumed by the arguers, which allows them to encode arguments into a shorter message by ignoring the common knowledge [1]. In particular, in this work, we use Argumentation Schemes (AS) [17] as common organisational knowledge to guide the construction of enthymemes by the proponents of arguments, as well as to guide the reconstruction of the intended argument by the recipients of such enthymemes.

Definition 1 (Argumentation Scheme). *An argumentation scheme is a tuple $\langle \mathcal{SN}, C, \mathcal{P}, \mathcal{CQ} \rangle$ with $\mathcal{SN}$ the argumentation scheme name (which must be unique within the system), C the conclusion of the argumentation scheme, $\mathcal{P}$ the premises, and $\mathcal{CQ}$ the associated critical questions.*

Definition 2 (Argument). *An argument is a tuple $\langle S, c \rangle^{\theta}_{\mathtt{sn_i}}$, where $\mathtt{sn_i}$ is the name of the argumentation scheme $\langle \mathtt{sn_i}, C, \mathcal{P}, \mathcal{CQ} \rangle \in \Delta_{\mathtt{AS}}$, θ is a most general unifier for the premises in $\mathcal{P}$ and the agent's current belief base, S is the set of premises and the inference rule of the scheme used to draw c (the conclusion). That is, S includes all instantiated premises from $\mathcal{P}$ — i.e., for all $p \in \mathcal{P}$, $p\theta \in \mathcal{S}$ — and the inference rule corresponding to the scheme $(\mathcal{P} \Rightarrow C)$; the conclusion c is the instantiation $C\theta$ such that $S \models c$.*

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

In our approach, AS are specified on top of a MAS as part of an organisational model [6], therefore they are shared by agents. Also, agents are able to instantiate and evaluate arguments based on [5, 8, 10, 11]. Furthermore, agents share knowledge regarding the organisational structure of the system, such as roles, authority links, etc. We denote all knowledge coming from the organisational infrastructure available to agents as $\Delta_{\mathtt{Org}}$, and we emphasise that all agents are aware of the information in $\Delta_{\mathtt{Org}}$. We denote all AS available for agents to instantiate arguments in the MAS as $\Delta_{\mathtt{AS}}$. Finally, we denote all knowledge available to an agent $\mathtt{ag_i}$ as $\Delta_{\mathtt{ag_i}}$, where $(\Delta_{\mathtt{Org}} \cup \Delta_{\mathtt{AS}}) \subset \Delta_{\mathtt{ag_i}}$. Enthymemes are defined as follow:

Definition 3 (Enthymeme). *Let $\langle S, c \rangle^{\theta}_{\mathtt{sn}}$ be an argument to agent $\mathtt{ag_i}$. An enthymeme for $\langle S, c \rangle^{\theta}_{\mathtt{sn}}$ is a tuple $\langle S', c \rangle^{\theta}_{\mathtt{sn}}$, where $S' = (S \setminus (\Delta_{\mathtt{Org}} \cup \Delta_{\mathtt{AS}}))$.*

2 SEMANTICS FOR ENTHYMEMES

Considering the performatives `assert`, `question`, `justify`, `refuse`, and `accept` [3, 4], we gave the operational semantics that formalises the construction (encoding) and reconstruction (decoding) of enthymemes associated with the speech act used by agents during dialogues. We define the semantics of speech acts for argumentation-based dialogues using enthymemes in AgentSpeak [15] (and in particular the Jason dialect [2]) using a widely-known method for giving operational semantics to programming languages [14].

The operational semantics is given by a set of inference rules that define a transition relation between agent configurations $\langle ag, C, M, T, s \rangle$ [16], where[1]: i) An agent ag is a set of beliefs bs and a set of plans ps; ii) An agent circumstance C is a tuple $\langle I, E \rangle$ where I is a set of *intentions* $\{i, i', \ldots\}$. Each intention i is a stack of partially instantiated plans and E is a set of *events* $\{(te, i), (te', i'), \ldots\}$, in which each event is a pair (te, i), where te is a triggering event and i is an intention — a stack of plans in the case of an internal event, or the empty intention T in the case of an external event; iii) M is a tuple $\langle In, Out \rangle$ whose components characterise the aspects of communicating agents (typically asynchronous). In is the mail inbox, which includes all messages addressed to this agent. Elements In have the form $\langle mid, id, ilf, cnt \rangle$, where mid is a message identifier, id identifies the sender of the message, ilf is the illocutionary force of the message, and cnt its content; Out is where the agent posts messages it wishes to send; messages in this set have exactly the same format as above, except that here id refers to the agent to which the message is to be sent; iv) T is a tuple with temporary information originally defined in [16]; in this paper, we only need the ι component, which keeps track of a particular intention being

[1] We use here only the components that are needed to define our semantics.

considered along the execution of a reasoning cycle; v) The current step within an agent's reasoning cycle is symbolically annotated by $s \in \{\text{ProcMsg}, \text{ExecInt}\}$, with ProcMsg the step for processing a message from M, and ExecInt for executing the selected intention.

The semantics of AgentSpeak makes use of "selection functions" which allow for user-defined components of the agent architecture. We use here only the S_M function, which is used to select one message from an agent's mail inbox, as originally defined in [16]. In the interest of readability, we adopt the following notational conventions in our semantic rules: i) If C is an AgentSpeak agent circumstance, we write C_E to make reference to the E component of C(similarly for other components); ii) We write: $b[s(id)]$ to identify the origin of a belief b, where id is an agent identifier (s is an abbreviation for *source*); and $b[dec(\mathtt{sn_i})]$ to identify information that was decoded from an enthymeme, guided by the AS $\mathtt{sn_i}$; iii) We use a function `prem(S)` which returns all premises in the support of the argument $\langle S, c\rangle$.

(**ExActSndAssert**)

$$\frac{T_t = i[head \leftarrow .\mathtt{send}(id, \mathtt{assert}, c); h]}{\langle ag, C, M, T, \text{ExecInt}\rangle \longrightarrow \langle ag, C', M', T, \text{ProcMsg}\rangle}$$

where:

$$\begin{array}{rcl} M'_{Out} & = & M_{Out} \cup \{\langle mid, id, \mathtt{assert}, c\rangle\} \\ C'_I & = & (C_I \setminus \{T_t\}) \cup \{i[head \leftarrow h]\} \end{array}$$

(**ExActSndJustify**)

$$\frac{\begin{array}{c} T_t = i[head \leftarrow .\mathtt{send}(id, \mathtt{justify}, \langle S, c\rangle^{\theta}_{\mathtt{sn_i}}); h] \\ S \subset \Delta_{\mathtt{ag}} \quad \Delta_{\mathtt{ag}} \models c \quad \langle \mathtt{sn_i}, C, \mathcal{P}, CQ\rangle \in \Delta_{\mathtt{AS}} \\ \forall p \in \mathcal{P}, p\theta \in \mathtt{prem}(S) \quad c = C\theta \end{array}}{\langle ag, C, M, T, \text{ExecInt}\rangle \longrightarrow \langle ag, C', M', T, \text{ProcMsg}\rangle}$$

where:

$$\begin{array}{rcl} M'_{Out} & = & M_{Out} \cup \{\langle mid, id, \mathtt{justify}, \langle S', c\rangle^{\theta}_{\mathtt{sn_i}}\rangle\} \\ & & \text{with } S' = S \setminus (S \cap (\Delta_{\mathtt{Org}} \cup \Delta_{\mathtt{AS}})) \\ C'_I & = & (C_I \setminus \{T_t\}) \cup \{i[head \leftarrow h]\} \end{array}$$

Sending an `assert`**,** `question`**,** `accept`**, and** `refuse` **message:** when an agent executes the internal action for sending a message with these performatives, that message is posted in the agent mailbox, M_{Out}, and the current agent intention is updated, removing the internal action, given that its execution is completed. In the semantic rule **ExActSndAssert**, the intention being considered is given by T_t, and it corresponds to $i[head \leftarrow .\mathtt{send}(id, \mathtt{assert}, c); h]$, in which the current step of the plan adopted to reach that particular goal is to execute the action $.\mathtt{send}(id, \mathtt{assert}, c)$. Thus, after executing that action, that particular intention is updated to $i[head \leftarrow h]$, given that action has already been executed by the agent[2].

Sending a `justify` **message:** when an agent executes the internal action for sending a message with the performative `justify`, the agent needs to have an argument for that particular conclusion[3] that was drawn using the AS $\mathtt{sn_i}$, according to Definition 2. Such argument is encoded into an enthymeme $\langle S', c\rangle^{\theta}_{\mathtt{sn_i}}$, where all common knowledge is removed from the argument support, i.e., the organisational knowledge $\Delta_{\mathtt{Org}}$ and the contents of the used AS $\Delta_{\mathtt{AS}}$ are removed. The corresponding message is posted in the agent's mailbox, M_{Out}, and the current agent intention is updated, removing the internal action, given that its execution is completed. The semantic rule **ExActSndJustify** implements the process for encoding an argument as an enthymeme.

[2]Note that the semantic rules for the performatives `question`, `accept`, and `refuse` are similar to rule **ExActSndAssert**, thus they were omitted.

[3]While we use, in this work, the *thoughtful* attitude [12, 13], other agent attitudes could be used just as well.

(**Question**)

$$\frac{S_M(M_{In}) = \langle mid, sid, \mathtt{question}, q\rangle}{\langle ag, C, M, T, \text{ProcMsg}\rangle \longrightarrow \langle ag, C', M', T, \text{ExecInt}\rangle}$$

where:

$$\begin{array}{rcl} M'_{In} & = & M_{In} \setminus \{\langle mid, sid, \mathtt{question}, q\rangle\} \\ C'_E & = & C_E \cup \{\langle +questions(sid, q), \mathsf{T}\rangle\} \end{array}$$

(**Assert**)

$$\frac{S_M(M_{In}) = \langle mid, sid, \mathtt{assert}, p\rangle}{\langle ag, C, M, T, \text{ProcMsg}\rangle \longrightarrow \langle ag', C', M', T, \text{ExecInt}\rangle}$$

where:

$$\begin{array}{rcl} ag'_{bs} & = & ag_{bs} \cup \{p[s(sid)]\} \\ M'_{In} & = & M_{In} \setminus \{\langle mid, sid, \mathtt{assert}, p\rangle\} \\ C'_E & = & C_E \cup \{\langle +asserts(sid, p), \mathsf{T}\rangle\} \end{array}$$

(**Justify**)

$$\frac{\begin{array}{c} S_M(M_{In}) = \langle mid, sid, \mathtt{justify}, \langle S', c\rangle^{\theta}_{\mathtt{sn_i}}\rangle \\ \langle \mathtt{sn_i}, C, \mathcal{P}, CQ\rangle \in \Delta_{\mathtt{AS}} \quad \forall p\theta \in S', p \in \mathcal{P} \quad c = C\theta \end{array}}{\langle ag, C, M, T, \text{ProcMsg}\rangle \longrightarrow \langle ag', C', M', T, \text{ExecInt}\rangle}$$

where:

$$\begin{array}{rcl} M'_{In} & = & M_{In} \setminus \{\langle mid, sid, \mathtt{justify}, \langle S', c\rangle^{\theta}_{\mathtt{sn_i}}\rangle\} \\ ag'_{bs} & = & ag_{bs} \cup \{p[s(sid)]\theta \mid \text{for all } p \in \mathcal{P} \text{ and } p\theta \in S'\} \cup \\ & & \{p[s(sid), dec(\mathtt{sn_i})]\theta \mid \text{for all } p \in \mathcal{P} \text{ and } p\theta \notin S'\} \\ & & \cup \{(\mathcal{P} \Rightarrow C)[s(sid), dec(\mathtt{sn_i})]\} \\ C'_E & = & C_E \cup \{\langle +justifies(sid, \langle S', c\rangle^{\theta}_{\mathtt{sn_i}}), \mathsf{T}\rangle\} \end{array}$$

Receiving `question` **or** `refuse` **messages:** the agent receiving the message will just remove the message from its mailbox and an event will be generated for that. The event is handled as usual by the agent's plans, which determine its strategy in the dialogue.

Receiving `assert` **or** `accept` **messages:** the agent receiving the message will remove the message from its mailbox, update its belief base with the message content properly annotated with the sender as the source of that information, and an event will be generated for that. The event is handled by the agent's plans according to its strategy in the dialogue.

Receiving a `justify` **message:** when an agent selects a `justify` message from its mailbox, the message is removed from the mailbox, the agent's belief base is updated with all information contained in the support of the intended argument, annotating clearly which formulæ have been received from the sender and which have been decoded, and an event $+justifies(sid, \langle S', c\rangle^{\theta}_{\mathtt{sn_i}})$ is generated for that. The event can be handled by the agent's plans according to its strategy in the dialogue. It is important to mention that, when the agent receives a justify message, the content is an enthymeme. Thus, the enthymeme is decoded into the original sender's argument, guided by the AS.

3 CONCLUSION

In this work, we proposed an operational semantics that specifies an enthymeme-based communication approach for BDI agents. Our approach is based on the idea that agents share AS and knowledge from the organisation, using that information to construct enthymemes from arguments, as well to reconstruct arguments from the enthymemes communicated by agents. We defined our operational semantics making reference to general components of the BDI architecture, therefore any language based on concepts such as beliefs, intentions, etc., could also benefit from our formalisation. Our work moves towards an enthymeme-based communication framework in MAS, extending our previews work [7], using the framework for AS proposed in [8, 10]. In future work, we intend to explore our approach for enthymemes in explainable MAS [9].

REFERENCES

[1] Elizabeth Black and Anthony Hunter. 2008. Using enthymemes in an inquiry dialogue system. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 437–444.

[2] 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.

[3] Peter McBurney and Simon Parsons. 2004. Locutions for Argumentation in Agent Interaction Protocols.. In *AC (Lecture Notes in Computer Science, Vol. 3396)*, Rogier M. van Eijk, Marc-Philippe Huget, and Frank Dignum (Eds.). Springer, 209–225.

[4] Peter McBurney and Simon Parsons. 2009. Dialogue Games for Agent Argumentation. In *Argumentation in Artificial Intelligence*, Guillermo Simari and Iyad Rahwan (Eds.). Springer US, 261–280.

[5] Alison R. Panisson and Rafael H. Bordini. 2016. Knowledge Representation for Argumentation in Agent-Oriented Programming Languages. In *2016 Brazilian Conference on Intelligent Systems, BRACIS*.

[6] Alison R Panisson and Rafael H Bordini. 2017. Argumentation schemes in multi-agent systems: A social perspective. In *International Workshop on Engineering Multi-Agent Systems*. Springer, 92–108.

[7] Alison R Panisson and Rafael Heitor Bordini. 2017. Uttering only what is needed: Enthymemes in multi-agent systems. In *Proceedings of the 16th International Conference on Autonomous Agents & Multiagent Systems (AAMAS-2017), 2017, Brasil*.

[8] Alison R Panisson and Rafael H Bordini. 2020. Towards a computational model of argumentation schemes in agent-oriented programming languages. In *2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. IEEE, 9–16.

[9] Alison R Panisson, Débora C Engelmann, and Rafael H Bordini. 2021. Engineering explainable agents: an argumentation-based approach. In *International Workshop on Engineering Multi-Agent Systems (EMAS)*.

[10] Alison R Panisson, Peter McBurney, and Rafael H Bordini. 2021. A computational model of argumentation schemes for multi-agent systems. *Argument & Computation* 3 (2021), 357–395.

[11] Alison R. Panisson, Felipe Meneguzzi, Renata Vieira, and Rafael H. Bordini. 2014. An Approach for Argumentation-based Reasoning Using Defeasible Logic in Multi-Agent Programming Languages. In *11th International Workshop on Argumentation in Multiagent Systems*.

[12] Simon Parsons and Peter McBurney. 2003. Argumentation-based dialogues for agent co-ordination. *Group Decision and Negotiation* 12, 5 (2003), 415–439.

[13] Simon Parsons, Michael Wooldridge, and Leila Amgoud. 2002. An analysis of formal inter-agent dialogues. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1* (Bologna, Italy) *(AAMAS '02)*. ACM, New York, NY, USA, 394–401.

[14] Gordon D Plotkin. 1981. A structural approach to operational semantics. (1981).

[15] Anand S. Rao. 1996. AgentSpeak(L): BDI agents speak out in a logical computable language. In *Proceedings of the 7th European workshop on Modelling autonomous agents in a multi-agent world : agents breaking away: agents breaking away* (Einhoven, The Netherlands) *(MAAMAW '96)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 42–55.

[16] Renata Vieira, Álvaro Moreira, Michael Wooldridge, and Rafael H. Bordini. 2007. On the Formal Semantics of Speech-act Based Communication in an Agent-oriented Programming Language. *J. Artif. Int. Res.* 29, 1 (June 2007), 221–267.

[17] D. Walton, C. Reed, and F. Macagno. 2008. *Argumentation Schemes*. Cambridge University Press.