A modular platform for argument and dialogue

Snaith, Mark; Lawrence, John; Pease, Alison; Reed, Chris

Published in:
Computational Models of Argument

DOI:
10.3233/FAIA200540

Publication date:
2020

Licence:
CC BY-NC

Document Version
Publisher's PDF, also known as Version of record

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200540

A Modular Platform for Argument and Dialogue

Mark SNAITH [1], John LAWRENCE, Alison PEASE and Chris REED
Centre for Argument Technology, University of Dundee, UK

Keywords. dialogue, dialogue games, dialogue execution

1. Introduction

The Dialogue Game Execution Platform (DGEP) [1] allows human and virtual participants to engage in a structured dialogue following a specified protocol. In this short abstract, we present PAD: an open-source platform for argument and dialogue that builds on DGEP by wrapping it in a modular architecture that allows new functionality to be easily added. We introduce one such module, the *Dialogue Utterance Generator* (DUG), which finds propositional content to populate the abstract move types provided by DGEP.

2. Platform description

The **Dialogue Game Execution Platform (DGEP)** forms the core of PAD. Its function is to keep a record of a dialogue, accepting played moves and generating the resultant dialogue state including the current speaker and next available moves.

Game descriptions are written in a revised and updated version of the Dialogue Game Description Language (DGDL) [2]. Output from DGEP is the available legal move *types* based on the protocol being followed, without any consideration for the propositional content of those moves. DGEP provides a template that should subsequently be filled if the move type is selected; for instance, the template for an "argue" move may be:

```
{"reply":{"p":"go_to_cinema", "q":"$q"}}
```

After DGEP has generated a set of available move types, they can be passed to the **Dialogue Utterance Generator (DUG)**, which takes the available move types and attempts to find propositional content to instantiate them into concrete moves. The DUG itself follows a modular design that allows different sources of content to be used, even within the same dialogue.

Core to the DUG are a set of *content descriptors* and associated *content locators*. A content descriptor is linked to the move type and template provided by DGEP and

[1]Corresponding author. E-mail: m.snaith@dundee.ac.uk.

describes how variables in the "reply" object should be populated. A content locator provides an implementation of an algorithm that actually finds the content. As a concrete example, a content locator that queries a MySQL database will have the following content descriptor for an "argue" move type:

```
argue{
 @mysql("SELECT premise FROM arguments WHERE conclusion='$p';");
}
```

Continuing the previous example, this query would be instantiated with the content of "p" (i.e. "...WHERE conclusion='go_to_cinema';"). The query is then passed to the content locator for querying MySQL databases, with the result being assigned to "q" in the reply. The reply is then made available to a user or agent as a concrete move that can be played in the dialogue. If the content locator returns multiple values, a concrete move is created for each piece of content.

Other potential sources of content include AIFdb [3] (which in turn can allow a participant to contribute to a past argument or debate), argument mining [4], or a logical representation such as ASPIC+ [5]. It is however possible, in principle, for content to be obtained from any queriable source.

3. Using the platform

The source code for DGEP and the DUG is available at `https://github.com/arg-tech`. Both are also available as web services at `https://ws.arg.tech`.

Acknowledgements

This work was supported by the European Union's Horizon 2020 research and innovation programme under Grant Agreement #769553. The authors are grateful to colleagues on the project for their feedback during the design and implementation.

References

[1] Bex F, Lawrence J, Reed C. Generalising argument dialogue with the Dialogue Game Execution Platform. In: Parsons S, Oren N, Reed C, Cerutti F, editors. Fifth International Conference on Computational Models of Argument (COMMA 2014). IOS Press; 2014. p. 141–152.

[2] Wells S, Reed CA. A domain specific language for describing diverse systems of dialogue. Journal of Applied Logic. 2012;10(4):309–329.

[3] Lawrence J, Bex F, Reed C, Snaith M. AIFdb: infrastructure for the argument web. In: Verheij B, Szeider S, Woltran S, editors. Fourth International Conference on Computational Models of Argument (COMMA 2012). IOS Press; 2012. p. 515–516.

[4] Lawrence J, Reed C. Argument Mining: A Survey. Computational Linguistics. 2019;45(4):765–818.

[5] Prakken H. An abstract framework for argumentation with structured arguments. Argument and Computation. 2010;1(2):93–124.