September 10, 2020

The Open Argumentation PLatform (OAPL)

Simon WELLS [a]
[a] *School of Computing, Edinburgh Napier University, Scotland, U.K.*

Keywords. Argument Analysis, Open Source Tools, Argument Visualisation

1. Introduction

The Open Argumentation Platform (OAPL[1]) pronounced "*opal*" is a suite of argumentation software that includes APIs, libraries, and user interfaces that work together to support a range of argument-oriented computational tasks and associated pipelines. OAPL is an open platform that is built around a suite of open-development, free-software tools, released under a permissive license. By developing and promoting open standards, the goal is to develop sustainable argumentation software, that finds real world uptake beyond the argumentation theory community, and which can act as a flexible framework for investigating new, and extending existing, techniques in argument analysis, processing, visualisation, and reuse.

The tools that make up the platform are designed to support a range of argument-centric activities such as reasoning over argument resources, dialogically-oriented interaction, manual argument analysis, and automated argument analysis. The suite currently comprises seven software tools which can be combined and configured to form a variety of argument pipelines. These tools all aim to have the following:

- A simple but extensible underlying data model.
- Clear extension points for domain specific analysis & representation tasks.
- Tooling to support import from other formats, e.g. AML, AIF, &c.
- An open source canonical implementation.
- Supporting Documentation.
- Liberal (GPL3) licensing.
- A completely open development model including public GIT repository & public issue/bug tracking.

2. Tools

OAPL currently comprises the following software tools:

[1] `http://www.openargumentation.org`

SADFAce[2] is a simple JSON based Argument description format, software library, and supporting tools to enable developers and researchers to describe arguments and to easily reuse their data. The goal is to make it as easy as possible to incorporate argument data into modern software. SADFace has a simple but extensible model that is compatible with AIF and can serialise other formats such as AML. There are both Python 3 and JavaScript implementations of the core SADFace format as well as supporting tools for SADFace document creation, editing, and manipulation. SADFace currently forms the core of OAPL and is the *lingua franca* that underpins the integration of the other tools.

ArgDB[3] is the main Argument Database that provides persistence of SADFace documents. This is a CouchDB based datastore which natively stores SADFace JSON documents. A couch-app is used to provide a web search interface. ArgDB is designed to run in either private/local mode or as a public argument data server as part of the argument web.

MonkeyPuzzle[4] [1] is a browser-based user interface for manual argument analysis. The interface is centered around a resource pane that holds the resource being analysed and a visual workspace in which a graph based argument visualisation is constructed.

ALIAS[5] [2] is "A Library for Implementing Argumentation Systems" a Python implementation of a library for working with Dung frameworks. This is currently used to provide a mechanism for automated reasoning over argument resources, for example, from SADFace documents.

DGDL[6] & ADAMANT[7] are the Dialogue Game Description Language [3] and its associated Python-based dialogue game execution platform. These two technologies work together to enable dialogue games to be specified, run, and managed.

Canary[8] is the most recent addition to OAPL, an argument mining library that is currently under heavy development. Canary is a Python library that builds on existing natural language toolkits

Many of these tools were developed independently so the path to closer integration is ongoing and non-exclusive. Contributors, developers, bug-fixers, and users are all welcomed to the OAPL community.

References

[1] J. Douglas and S. Wells. Monkeypuzzle. In *Proceedings of the 17th International Workshop on Computational Models of Natural Argument (CMNA17)*, 2017.

[2] S. Wells and R. La Greca. Introducing alias. In *Proceedings of the 15th International Workshop on Computational Models of Natural Argument (CMNA15)*, 2015.

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

[2] `https://github.com/Open-Argumentation/SADFace`
[3] `https://github.com/Open-Argumentation/ArgDB`
[4] `https://github.com/Open-Argumentation/MonkeyPuzzle`
[5] `https://github.com/Open-Argumentation/ALIAS`
[6] `https://github.com/Open-Argumentation/DGDL`
[7] `https://github.com/Open-Argumentation/ADAMANT`
[8] `https://github.com/Open-Argumentation/Canary`