DAMN: Defeasible Reasoning Tool for Multi-Agent Reasoning
Abdelraouf Hecham, Madalina Croitoru, Pierre Bisquert

To cite this version:

HAL Id: lirmm-02393877
https://hal-lirmm.ccsd.cnrs.fr/lirmm-02393877
Submitted on 4 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
DAMN: Defeasible Reasoning Tool for Multi-Agent Reasoning

Abdelraouf Hecham,1 Madalina Croitoru,1 Pierre Bisquert2
1 LIRMM, University Of Montpellier, France, {hecham,croitoru}@lirmm.fr
2 IATE, INRA, Montpellier, France, pierre.bisquert@inra.fr

December 4, 2019

Abstract

This demonstration paper introduces DAMN: a defeasible reasoning platform available on the web. It is geared towards decision making where each agent has its own knowledge base that can be combined with other agents to detect and visualize conflicts and potentially solve them using a semantics. It allows the use of different defeasible reasoning semantics (ambiguity blocking/propagating with or without team defeat) and integrates agent collaboration and visualization features.

1 Relevance and Related Work

One of the aims of knowledge representation and conflict-tolerant reasoning is to help with the decision making process where the knowledge of different stakeholder agents can be put together in order to help reach a decision or at least detect conflicts. In order to reason with combined knowledge, a conflict-tolerant reasoning mechanism is needed. Defeasible reasoning [7] allows to reason with incomplete knowledge where conclusions can be challenged by additional information. It provides different semantics given its systematic reliance on a set of intuitions and rules of thumb (ambiguity propagating, blocking, team defeat), which have been long debated between logicians.

Statement Graphs (SG) [5] have been defined in order to represent different semantics of Defeasible Logics [1] in a single formal framework.

This demonstration paper presents a platform called DAMN1 that implements SG and provides a data engineer with the following features:

• Multi-agent integration: allowing agents to have their own knowledge bases (having, potentially, nothing in common).
• Real-time agent collaboration with access control.
• Defeasible reasoning on the combined knowledge featuring ambiguity propagating/blocking, with or without team defeat, used to solve contradictions between agents.

1https://hamhec.github.io/damn/home
• Visualization of the resulting reasoning graph.
• Importing knowledge bases using DLGP formats [2], or a JSON file.

A presentation video explaining all of the features is available online at http://tiny.cc/mx2hcz.

2 Technical Details

Let us consider the following knowledge: kowalski is undeniably a penguin with potentially broken wings, penguins are bird who do not fly, generally birds fly (a less trustworthy rule), and one cannot say that birds with broken wings fly. The associated SG, with the queries fly(kowalski) and notFly(kowalski) is shown in Figure 1.

![Figure 1: Statement Graph example.](image)

**DAMN Workflow.** The typical workflow of the DAMN platform is: (1) Each user creates a project that will contain the different knowledge bases; (2) The user can add or remove agents along with their knowledge bases (that can be imported from DLGP files); (3) The user can invite other users to update the $KB$ of some or all agents in real-time; (4) The user can build the SG using the $KB$ of some or all the agents and run queries with a certain semantics; (5) Last, the user can interact with the resulting graph (see Figure 1 for an example of a graph).

**DAMN Architecture.** The DAMN architecture (cf. Figure 2) is composed of:
• A client side application (UI built with Angular) for user interaction that displays the agents and their knowledge bases along with graph visualization features.
Figure 2: DAMN architecture.

- A collaboration server that handles the interaction between different users on the same project (a Spring Websocket server).
- A reasoning server that contains an implementation of SG (Java REST server) based on GRAAL [2] extended with features specific to defeasible reasoning (semantics, preferences between rules, etc.).
- A persistence server that stores the data (a MongoDB server).

**Demonstration Application Scenario.** In the demonstration we will consider the real world scenario issued from the EU H2020 NoAW project, where 21 experts from different domains put together their knowledge in order to reason about how to manage wastes from wine production. We will show how DAMN allowed the detection of conflicts. For instance, due to obsolete knowledge one wine maker argued that the wine pomace should be sent to distilleries because it was mandatory by law. Another agent argued that it was no longer the case since 2016. This use-case is available along with other examples (such as a group of friends trying to choose a restaurant to go to) on the website of the tool (not accessible here for privacy reasons).

3 Significance and Discussion

Different tools for defeasible reasoning have been proposed in the literature, most notably, ASPIC+\(^2\) [8], DeLP\(^3\) [3], DEFT [4], ELDR [5], Flora-2 [9], and SPINdle [6]. However, each tool allows for a different set of defeasible reasoning features and none of them provides support for multi-agent collaboration or visualization.

\(^2\)http://aspic.cossac.org

\(^3\)http://lidia.cs.uns.edu.ar/delp_client
Given the recurrent need to reason with knowledge extracted from the Semantic Web, we opted for SG and their ability to represent most defeasible reasoning semantics and their support for existential rules. We have implemented SG and provided several features making it usable in a distributed decision making context. We evaluate the usefulness of our tool by considering a practical decision making scenario issued from the H2020 NoAW aiming at finding novel wine waste management methods.

4 Acknowledgments

The authors acknowledge the support of the CASDAR Docamex Programme from the French Ministry of Agriculture (2016–2020) and the H2020 NoAW European project.

References


