Scrutable Plan Enactment via Argumentation and Natural Language Generation

(Demonstration)

Martin Caminada
Computing Science
Univ. of Aberdeen, UK
m.caminada@abdn.ac.uk

Roman Kutlak
Computing Science
Univ. of Aberdeen, UK
r.kutlak@abdn.ac.uk

Nir Oren
Computing Science
Univ. of Aberdeen, UK
n.oren@abdn.ac.uk

Wamberto W. Vasconcelos
Computing Science
Univ. of Aberdeen, UK
w.w.vasconcelos@abdn.ac.uk

ABSTRACT

Autonomous systems suffer from *opacity* due to the potentially large number of sophisticated interactions among many parties and how these influence the outcomes of the systems. It is very difficult for humans to scrutinise, understand and, ultimately, work with such systems. To address this shortcoming, we developed a demonstrator which uses formal argumentation techniques, coupled with natural language generation, to explain the rationale of a hybrid software-human many-party joint plan during its enactment.

1. INTRODUCTION

With advances in planning and reasoning techniques, autonomous systems are increasingly able to generate complex plans to achieve their goals. While such plans are often better than those that a human planner can create, they can only take into account the information available to the system. Within a human-agent team, a person is often in a position to provide such information to the system, but may be unaware of what the system does or does not know.

Clearly, a straightforward interrogation of an autonomous system's knowledge base is infeasible due to its size. One strategy which we investigate involves focusing on *relevant* knowledge by allowing the human to query the system's proposed plan. By identifying the justifications for this plan, the human can be satisfied that the system has appropriate knowledge. On the other hand, if a gap in information is identified, the human can add this knowledge to the system, causing replanning to take place.

We propose a dialogue-based approach to plan understanding: a human is able to interrogate the system through such a dialogue, requesting justifications for actions and adding (fresh) knowledge. Such justifications for actions arise due to preconditions for actions and their effects (with regards to the goal state), as well as the actions required to bring about the preconditions for later actions. The human can therefore repeatedly ask for justifications, with the system recursing down from actions and their preconditions all the way to facts within its knowledge base as required. Alternatively, the human can, at any point, accept the system's justifications and move onto a different line of questioning.

An additional novel feature of our system is its ability to handle inconsistencies and non-monotonicity – while some action may be appropriate in most situations, an exceptional specialisation of such situations may require an entirely different action. We provide this ability in our system, as well as the dialogical aspect using formal argumentation theory. Planning and argument theory deal with formal logics, and, to facilitate non-expert use of the system, we also introduce a natural language generation component to translate logical formulae onto English.

In Figure 1 we show the components of our architecture and how they relate. An interface ("UI") provides means for users to interact with the system. The "Controller" connects

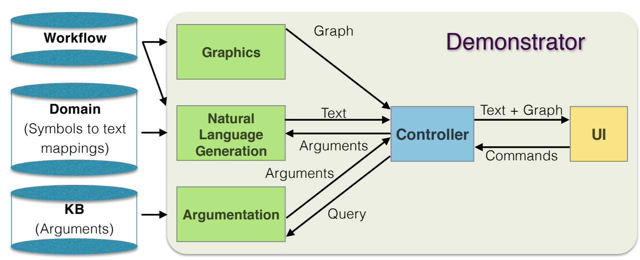

Figure 1: Architecture

the UI with other components, invoking them and combining their outputs: (from the top) a graphic representation of workflows, a textual (English) representation of a formal argument (including proofs), and the formal arguments themselves (to be presented in English). Ours is a modular architecture in which workflows (generated off-line), domain knowledge (for text generation) and knowledge base(s) (to build arguments) can be exchanged.

2. ARGUMENTATION

The decisions on what actions to take are made using formal argumentation techniques [3]. For each decision, the relevant arguments are constructed, from a knowledge base consisting of strict and defeasible rules. Construction is done in an ASPIC-style way [1, 4], with an argument being a tree-like structure of rules, where the conclusion of a child-rule feeds into the antecedent of its parent-rule. The notion of attack is defined in terms of rebut and undercut [1]. Unlike ASPIC+, our work utilises unrestricted rebut, allowing a claim to be rebutted if a defeasible rule was used at any point to infer it.

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

Such unrestricted rebut should be more natural for humans than ASPIC+'s restricted rebut; we addressed several important underlying theoretical issues related to unrestricted rebut.

Underpinning our approach to scrutability is the ability of a human to engage in dialogue to understand *why* some claim is (or is not) considered true. To this end we have extended work such as [2], creating a dialogical proof procedure based on the grounded semantics for structured argumentation. In our setting, this proof procedure allows a dialogue participant to identify the reasons behind a given plan.

3. PLANS

Our demonstrator uses a many-party global plan, describing actions and their pre- and post-conditions. Additionally, our plans are non-linear, with choice points ("forks") describing parallel or alternative courses of action, as well as loops (for repetitive actions or portions of the plan) and the "joining" of alternative sub-plans onto a common (ensuing) sub-plan. We assume that our plans have been assembled "off-line", and we concentrate on their *enactment*. Another important contribution of our work is the use of a natural language generation (NLG) component, allowing a non-technical user to understand the reasons behind the plan. Such an NLG component is, by necessity, domain dependent – a different content and presentation is required to generate, for example, a weather forecast as opposed to a basketball game summary. The demonstrator was designed so that the domain rules can be supplied in the form of an ontology, and our NLG component then provides mappings from literals used in the ontology's rules and phrases into English (or potentially some other language), which are then combined into sentences.

4. THE DOMAIN

We consider the unmanned aerial vehicle (UAV) domain. Here, a human operator monitors a UAV, which functions at various different levels of autonomy. We concentrate on levels where the UAV shows the operator the next action and the operator provides the confirmation of the action. Alternatively, the UAV can have an implicit consent and the operators intervene only when they disagree with the UAV's decision. At these levels of autonomy, operators require a detailed understanding of why a plan should be executed, and what its effects will be in order for a mission to be successfully executed. It is also important to allow operators to explore what alternatives were considered and to be able to update possibly obsolete information.

Within our scenario, we assume that the UAV has a mission to take photos at a particular area. After that, the UAV should land, normally at the base. If an event that prevents the UAV from safely reaching the base occurs, the UAV will select the appropriate airfield to land on. Figure 2 shows

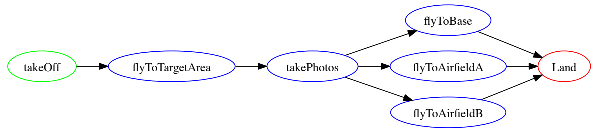

Figure 2: A potential plan for a UAV

```
Why rejected flyToBase?
System: UAV requires immediate landing due to system
    malfunction.
Why rejected flyToAirfieldA?
System: No visual landing at airfield A possible be-
    cause low visibility at airfield A.
Why rejected alpA?
System: No automated landing possible at airfield A
    because no Instrumental Landing System detected at
    airfield A.
```

Figure 3: Sample System Output

how this scenario is represented in our system. We now suppose the UAV finishes taking photos and instead of flying to base, the UAV informs the operator that the next action is `flyToAirfieldB` (UAV is going to fly to airfield B) instead of the expected action `flyToBase` (flying to the base). The dialog-like interaction in Figure 3 shows how an operator might question the system's decisions. In the current version, the operator is limited to using symbolic constants for the literals in the system's knowledge base. In future versions, we shall implement a parser capable of mapping English to logical formulae. The output of the system, however, is capable of producing basic English sentences. These sentences are more natural than the corresponding rules such as "`lvA --> -vlpA`" representing "if there is low visibility at airfield A, visual landing at the site is not possible".

5. CONCLUSIONS & FUTURE WORK

Our demonstrator currently concentrates on the explanation of existing plans. However, we are currently extending our approach to deal with the complete plan life-cycle. Therefore, we will utilise argumentation to facilitate human understanding during plan creation (allowing the human to introduce additional relevant information into the plan generation process of which the system might have been unaware); plan enactment (the current focus of our work); exception handling and replanning (allowing the human to understand and critique the re-planning process). To do so, we will extend techniques such as those described in [5] to use structured argument, and study how such arguments should be presented to humans in a non-technical manner while preserving clarity. Finally, our current work is rooted in classical planning, and we also intend to consider how argumentation and explanation can best be applied to richer planning formalisms (e.g. partial order planning).

6. REFERENCES

[1] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Art. Intell.*, 171(5-6):286–310, 2007.

[2] M. Caminada and M. Podlaszewski. Grounded semantics as persuasion dialogue. In *Procs COMMA 2012*, pages 478–485, 2012.

[3] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Art. Intell.*, 77(2):321–357, 1995.

[4] S. Modgil and H. Prakken. A general account of argumentation and preferences. *Art. Intell.*, 195(0):361 – 397, 2013.

[5] P. Pardo and L. Godo. A temporal argumentation approach to cooperative planning using dialogues. In *CLIMA*, volume 8143 of *LNCS*. Springer, 2013.