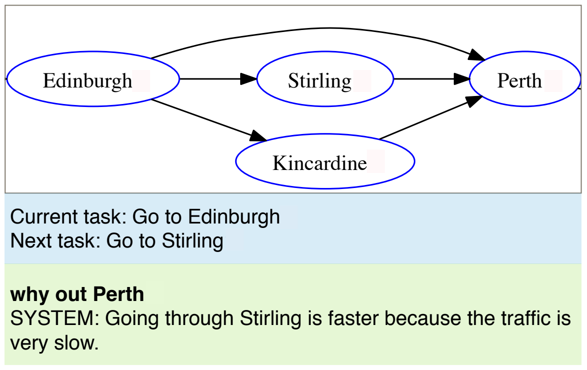
Demo: Making Plans Scrutable with Argumentation and Natural Language Generation

Nava Tintarev
University of Aberdeen
Aberdeen, UK
n.tintarev@abdn.ac.uk

Roman Kutlak
University of Aberdeen
Aberdeen, UK
r.kutlak@abdn.ac.uk



Current task: Go to Edinburgh
Next task: Go to Stirling

why out Perth
SYSTEM: Going through Stirling is faster because the traffic is very slow.

Figure 1: Example workflow and a dialogue excerpt.

IUI'14, Feb 24-27 2014, Haifa, Israel
ACM 978-1-4503-2729-9/14/02.
http://dx.doi.org/10.1145/2559184.2559202

Abstract

Autonomous systems perform tasks without human guidance. Techniques for making autonomous systems scrutable and, hence, more transparent are required in order to support humans working with such systems. The Scrutable Autonomous Systems (SAsSy) demo shows a novel way of combining formal argumentation and natural language to generate a human understandable explanation dialogue. By interacting with SAsSy users are able to ask why a certain plan was selected for execution, why other alternatives were not selected, and to modify information in the system.

Author Keywords

Explanations, Argumentation, Natural Language, Agents

ACM Classification Keywords

H.5.2 [User Interfaces]: Natural Language, Interaction styles, Graphical user interfaces (GUI)

Background and application context

An *autonomous system* consists of physical or virtual entities, or *agents*, that can perform tasks without continuous human guidance. While increasing reasoning capacity can enable an autonomous system to handle a wider range of situations, modelling and verifying the operation of such systems becomes increasingly difficult.

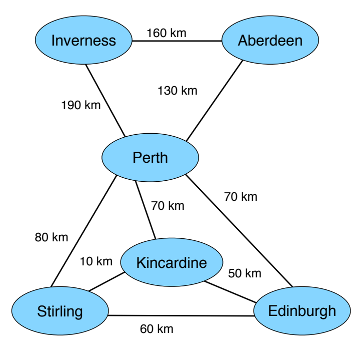


Figure 2: This map is included as an illustration of the different geographic locations used in the example. In our system, routes are represented in the knowledge base as rules of what kinds of actions are allowed.

This often means that people struggle to establish why a system chose to behave as it did. Users may have trouble identifying what alternative actions the system considered, and to determine why these alternatives were not selected for execution by the system. In other words, such systems are *opaque*. It is equally vital that a user can identify undesired actions before they are carried out, and interfere appropriately if need be: even if the user understands the system they need to be able to cancel actions or suggest alternatives with relative ease in a timely manner.

The SAsSy project[1] has for the last year been investigating computational mechanisms for providing transparency to humans regarding the internal workings of an autonomous system. To do this, we use formal argumentation in combination with natural language. The system explains which sequence of actions, or what plan, have been chosen for execution by the system and *why* a certain plan has been selected. That is, the user should be able to follow a chain of reasoning with arguments and counter arguments. The system also allows users to provide additional information which can be used to modify the arguments, and subsequently, the plan.

The scenario is based on a delivery driver in Scotland who is delivering a package between two cities (from Edinburgh to Inverness) and driving back. In this case, the plan is a choice of routes with a sequence of driving actions to a number of intermediate locations. Each location serves as a potential choice point from which several other locations may be possible. Some routes yield shorter distances between points, but given other factors such as traffic and road conditions the shortest route is not always the best option. The driver executes the plan step by step, and questions the system as required during the execution.

[1]http://scrutable-systems.org

System description

Explanations have frequently been a component of intelligent systems (IS) such as expert systems [1, 5] and recommender systems [6, 8]. The explanation capabilities in expert systems have often been evaluated with users in terms of whether they increase acceptance of an intelligent system or acceptance of decisions. However, there are other reasons why explanations may be introduced to an IS including transparency and scrutability [8].

The demo shows a novel form of interaction between two core technologies: a formal reasoning mechanism and natural language generation tools to translate logical statements into English. Our system is developed in Python and is available under the BSD licence[2].

Reasoning mechanism. Our knowledge base uses two kinds of rules: *defeasible* ($==>$) and *strict* ($-- >$). While both rules act as an implication defeasible rules can have exceptions, while strict rules cannot. Intuitively, defeasible rules capture situations that are *usually* true.

Rules without pre-conditions on the left-hand side are used to represent asserted or assumed information. We use the terms *rule* and *information* interchangeably.

The reasoning mechanism in SAsSy is based on argumentation, where the arguments are derived from rules in the knowledge base. Arguments are represented as a directed graph, in which the nodes constitute arguments and arcs between nodes symbolise attacks between arguments.

Argumentation, unlike traditional rule-based reasoning, can reason even in the face of contradicting statements. This gives the added advantage of being able to provide

[2]https://bitbucket.org/rkutlak/sassy

The system allows the user to ask three types of questions:

1. *"What can I do next?"*: This is also visible from the workflow (Figure 1; e.g. Stirling1, Kincardine1 and Perth1).
2. *"Why does the system NOT say that I should do Y?"*: The user can ask why an option is rejected: "why out Perth1?". The system derives the relevant rules and translates them into natural language (e.g., "Going through Stirling is faster because the traffic is very slow.").
3. *"Why does the system say that a certain thing is true?"*: The user can type a question such as "why traffic_very_slow" in the dialogue field and receive an answer such as "The traffic is very slow because of an accident on the bridge.".

information about which alternatives were considered. Furthermore, argumentation lends itself to dialogue-based explanations where the explanation can be provided in a piece-meal fashion [3].

Until now, formal argumentation has largely focused on formal notions such as acceptability of arguments and characterizing undefeated arguments [4]. While there is a precedent of research on formal argumentation and natural language dialog [4], the work in SAsSy is novel in that it can explain plans (rather than individual steps) with multiple decision points and help users understand which alternative options were considered.

The development of our system is also driven by experiments with users. The psychology of human reasoning as validation for argumentation semantics is a largely unexplored area [7], but a recent strand of work takes its inspiration from human dialogue to find intelligible explanations of an argument's status (i.e., whether to accept or reject the conclusion of an argument) [2].

The Natural Language Generation.

Natural Language Generation (NLG) is the study of computer algorithms which produce understandable and appropriate texts in English or other natural languages, from some underlying non-linguistic data. In our case, the non-linguistic data are the rules capturing the knowledge in our system.

The rules are formed from literals (e.g., `snow_road`), which can be ambiguous or difficult to understand. We use NLG techniques to convert the literals to more natural text as well as to improve the presentation by removing unnecessary information. For example, since defeasible rules capture implications that are usually true, we do not present the exceptions to such rules to the user.

We plan to include a summary of the presented plan as well as to use other NLG techniques such as aggregation (combining simple sentences together for better presentation) and referring expression generation (e.g., using pronouns when referring to past entities) to improve the presentation of the information. Indeed, in order to effectively be able to communicate 'why' certain decisions are preferable, the user needs to be able to understand *what* the recommended plan is first.

Demo

Our demonstrator shows how argumentation can be used to support dialogue structure, and how natural language can be used to present reasoning rules in a way that is familiar to users. The possible actions are represented as a workflow which is visualized as a graph, accompanied by natural language descriptions. In addition, the user can contribute to the dialogue by asking questions. Figure 1 demonstrates the main features of the SAsSy demo.

Currently, user input is limited to controlled symbolic constants such as 'Perth1' or 'traffic_very_slow'. Later versions of the system will make explicit the vocabulary available to a user regardless of whether it is controlled or processed from natural language.

The user may choose to question the system several times. For example, when the system tells the user that they cannot take a certain sub-route the user can query what knowledge was used to derive this argument, such as the case with traffic_very_slow (see point 3 on the left).

A user can choose to manually override or alter the knowledge base. A manual override means they simply

The previous defeasible rule can be changed to the following rule:
`snow_road`
= (`-plough_road`) =>
`closed_road`,
which might be read as "*if a road is covered by snow, it is usually closed unless it has been ploughed*."

An example of a strict rule:
`flood_road` −− >
`closed_road`,
which might be read as "*if a road is flooded, it is closed*."

An example of a defeasible rule:
`snow_road` ==>
`closed_road`,
which might be read as "*if a road is covered by snow, it is usually closed*."

proceed as they wish, for example by typing "next Perth1". Conceptually, altering the knowledge base is important if a similar process is likely to be repeated – so that users do not have to manually override repeatedly. For example, the situation may have changed and the user may want to tell the system that the accident is no longer causing issues: **Driver**> retract accident_on_bridge

The system deletes the information of an accident on a bridge, which affects the argument against the default route to Perth. The system now no longer has a reason to recommend the route though Stirling, and returns to suggesting going straight to Perth.

The system also represents a preference ordering among rules in case of conflicts to mitigate deadlocks, or situations where the system is undecided. So, if two arguments attack each other, only the attack of the argument with the more preferred rule holds true.

Apart from out of date information, the user and the system may have different information. For example, the system may be telling the user that they cannot go through Kincardine because of the weight of their van, but has gotten the weight wrong. In such cases users can introduce new information to the system such as a new weight for the van.

The explanation capability in SAsSy is under development and guided by ongoing user experiments. Currently we are planning experiments on the best way to tailor information content to the areas of responsibility of a user.

Acknowledgments

We would like to thank our colleagues from the SAsSy team for their comments and suggestions. This research has been carried out within the project "Scrutable Autonomous Systems", funded by the Engineering and Physical Sciences Research Council, grant ref. EP/J012084/1.

References

[1] Buchanan, B. G., and Shortliffe, E. H., Eds. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison Wesley, 1984.

[2] Caminada, M., and Podlaszewski, M. Grounded semantics as persuasion dialogue. In *COMMA*, B. Verheij, S. Szeider, and S. Woltran, Eds., vol. 245 of *Frontiers in Artificial Intelligence and Applications*, IOS Press (2012), 478–485.

[3] Caminada, M., Podlaszewski, M., and Green, M. Explaining the outcome of knowledge-based systems; a discussion-based approach. In *AISB* (2013).

[4] Chesñevar, C. I., Maguitman, A. G., and Loui, R. P. Logical models of argument. *ACM Computing Surveys 32* (2000), 337–383.

[5] Darlington, K. Aspects of intelligent systems explanation. *Universal Journal of Control and Automation 1* (2013), 40–51.

[6] McSherry, D. Explanation in recommender systems. *Artificial Intelligence Review 24(2)* (2005), 179 – 197.

[7] Rahwan, I., Madakkatel, M. I., Bonnefon, J.-F., Awan, R. N., and Abdallah, S. Behavioural experiments for assessing the abstract argumentation semantics for reinstatement. In *Cognitive Science* (2010).

[8] Tintarev, N., and Masthoff, J. Evaluating the effectiveness of explanations for recommender systems: Methodological issues and empirical studies on the impact of personalization. *User Modeling and User-Adapted Interaction 22* (2012), 399–439.