SAsSy — Scrutable Autonomous Systems

Nava Tintarev, Roman Kutlak, Nir Oren, Kees Van Deemter
Matt Green, Judith Masthoff, Wamberto Vasconcelos
University of Aberdeen, email: n.tintarev@abdn.ac.uk

Abstract. An autonomous system consists of physical or virtual systems that can perform tasks without continuous human guidance. Autonomous systems are becoming increasingly ubiquitous, ranging from unmanned vehicles, to robotic surgery devices, to virtual agents which collate and process information on the internet. Existing autonomous systems are opaque, limiting their usefulness in many situations. In order to realise their promise, techniques for making such autonomous systems scrutable are therefore required. We believe that the creation of such scrutable autonomous systems rests on four foundations, namely an appropriate planning representation; the use of a human understandable reasoning mechanism, such as argumentation theory; appropriate natural language generation tools to translate logical statements into natural ones; and information presentation techniques to enable the user to cope with the deluge of information that autonomous systems can provide. Each of these foundations has its own unique challenges, as does the integration of all of these into a single system.

1 Introduction

An *autonomous system* consists of physical or virtual systems that can perform tasks without continuous human guidance. Autonomous systems are becoming increasingly ubiquitous, ranging from unmanned vehicles, to robotic surgery devices, to virtual agents which collate and process information on the internet. Such systems can potentially replace humans in a variety of tasks which can be dangerous (such as refuelling a nuclear reactor), mundane (such as crop picking), or require superhuman precision (as in robotic surgery). Note that some of these tasks could be safety critical, in the sense that human life can be at risk if the autonomous system does not behave as intended. The reasoning processes driving an autonomous system can range from simple reactive mechanisms (potentially interacting to create complex behaviours c.f. [1]), to rule based systems such as [11] which are widely studied in the agents literature to very complex formalisms (see [12] for an overview). While increasing reasoning complexity can enable an autonomous system to handle a wider range of situations, modelling and verifying the operation of such systems becomes increasingly difficult.

A *distributed autonomous system (DAS)* consists of multiple autonomous components (often referred to as *agents*), which can communicate with each other while cooperating or competing to achieve some set of goals. The complexities of autonomous systems, particularly when distributed, means that humans struggle to establish why a system chose to behave as it did, to identify 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*. Such opacity is exacerbated by the formal models typically used to drive the reasoning behaviour in such systems — a human (and particularly a non-expert) often struggles to communicate with an autonomous system. This lack of understanding can lead to unrealistic expectations of an autonomous system, or alternatively to a lack of trust in it, causing inefficiencies at best, and leading to dangerous outcomes in the worst cases. Such problems limit the adoption of these types of systems.

The SAsSy project[1] proposes to investigate computational mechanisms for providing transparency to humans regarding the internal workings of a DAS. More specifically, we seek to utilise formal argumentation techniques to explain (to a human) *why* some plan was chosen for execution by the system, and to allow the human to provide additional information which can be used to modify the plan. We also aim to identify the best ways to present the explanations (primarily as text, but also in diagrammatic form) to a human operator, based on extensive knowledge acquisition, user modelling and user evaluation.

2 Example

Let us provide a simple example from the railway domain that illustrates some of the issues. In this scenario multiple stakeholders have to agree on a plan for maintaining railway lines. We assume two actions are possible: `Move(equipment, from, to)` and `Repair(equipment, location)`. Repair is performed using equipment, or more specifically a crane c_1 or c_2, and the crane has to be moved to a location before repair of that location can take place. Two locations, a and b need to be repaired. Location b has to be repaired before location a (an example of a constraint; cannot be violated). Each of the stakeholders is represented by an agent and each agent has different preferences and requirements. In our scenario, agent α prefers not to use crane c_2 (an example of preference; can be violated). Now assume the following starting 'state of the world' S_0: Locations a and b need to be repaired, crane c_1 is at location a and crane c_2 is at location b. S_G is the goal state – the desired state of the world where both locations a and b are fixed and all constraints are fulfilled.

There are many plans that achieve the goal. Figure 1 shows the two simplest plans. Suppose the system recommends plan A and a human user has to decide whether to follow the plan.

The user may ask why the system chose plan A over plan B. Plan A contains more steps (an extra Move action). We note that the question can be asked and answered at different levels of granularity. For example, a user may request an explanation for the transition $s_0 \rightarrow s_1$ ("Why did you move c1 first?") or for the plan itself ("Why

[1] `http://www.abdn.ac.uk/ncs/computing/research/ark/projects/current/sassy/`

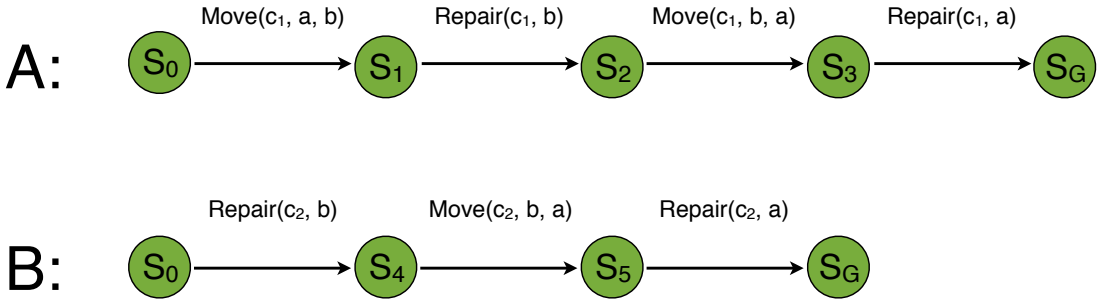

Figure 1. An example of two possible plans. A plan can be viewed as a sequence of transitions changing the state of the world. S_0 is the initial state and S_G is the final (goal) state. The plan can be verbalised as, e.g.:*"Move crane c_1 to location b and fix it. Then move the crane to location a and fix location a. "*

did you select a plan with more steps?"). It is also possible to argue against competing plans, or for the winning plan.

A possible explanation for the transition may be: *"So that c_1 could repair location b."* (why plan A) or *"Using c_2 at b goes against α's preference."* (why not plan B) . For the plan, the explanation may be: *"Because this means that we respect α's preference not to use crane c_2* (why plan A), or *"B involves the use of crane c_2, which goes against the preference of agent α."* (why not plan B).

We have presented a very simple example, that is limited to a single decision or choice point. A more complex plan might involve a large number of decisions (as well as a larger number of competing plans). However we believe it to be sufficient to illustrate the majority of challenges scrutability of plans and arguments brings forward. In the next section we elaborate on these challenges.

3 Challenges

Agents execute actions in order to achieve goals. Within a distributed system, the choice of actions to execute depends on both the environment and the actions of other agents. Goals usually require more than one action to be executed before they are achieved, and agents therefore generate *plans*, either individually, or through interactions with other agents. We therefore believe that in order to understand a distributed autonomous system, a human must be able to understand the plans within the system — including constraints, dependencies, and why a given plan may have been chosen over other plans. Given such an understanding, the human can *critique* the plans if necessary, providing the agents with additional information which they can use to modify their plans. Section 2 gives an example of supporting a human in making a choice between two plans where there is both a preference in terms of which resources to use and a constraint in terms of the order in which certain things can happen.

Recent research has investigated using formal argumentation to perform distributed planning [15]. Much of the literature in argumentation theory concerns itself with the *status* of an argument, that is, whether the argument licenses certain conclusions. This work assumes the existence of a static set of arguments, and computes an argument's status based on this set. Intuitively, arguments can be seen as a debate, or reasoning for and against a course of action. The psychology of human reasoning as validation for argumentation semantics is a largely unexplored area [10], but a recent strand of work takes its inspiration from human dialogue to find intelligible explanations of an argument's status [2]. A popular approach to the use of argumentation in planning requires the identification of appropriate *argumentation schemes* — common templates for argument — which are able to describe the planning process [5, 9].

Formal argument theory represents arguments using a logical language. In order to make arguments accessible to non-technical users, a system must be able to translate formulas of this formal logical language into natural language. Surprisingly little work in *natural language generation* has considered how such logical statements can be expressed in a clear and understandable manner. Substantial work on proof presentation exists, but this tends to focus on the proof structure (e.g., by omitting easily inferable information [4]) rather than the individual propositions. Despite useful early work in the 1980's [17], what is largely missing is an algorithm that converts each input formula into its most accessible form. NLG researchers have typically simplified by departing from a fixed logical form in which crucial presentation decisions have already been made (e.g., [6, 14]). For example, suppose the system produces the following argument in support of an action A:

$$(\neg success \rightarrow \neg q)\ \&\ (\neg q \rightarrow \neg p)\ \&\ (\neg p \rightarrow \neg A).$$

This formula is unnecessarily complex, and existing NLG systems that express this "word-by-word" would tend to generate something like: *"If success is not achieved then q is false. If q is false then p is false. If p is false then Action A is not performed"*. A much simpler presentation would be possible, however, e.g. *"q guarantees success; p implies q; Action A results in p"*. To allow natural language to do this, however, the formula above first needs to be converted into the equivalent

$$(q \rightarrow success)\ \&\ (p \rightarrow q)\ \&\ (A \rightarrow p).$$

There are two challenges here: Firstly, one needs to find out what is an optimally understandable format for expressing a given proposition. This is an empirical question that can only be solved using extensive experimentation with human subjects. Secondly, one needs to find algorithms for actually finding that optimal form. This can be extremely challenging computationally, particularly if the logic is very expressive.[2] This is a famous open problem in Natural Language Generation (NLG), known as the Logical Form Equivalence problem ([13], [16]). We shall attack the problem by relaxing the requirement that it is always necessary to find the unique *optimal* form, settling for

[2] Logical formulas are typically equivalent to infinitely many other formulas, and finding out if two formulas are equivalent is undecidable in many logics, or else computationally very expensive.

heuristics that give good results in most situations. A simple example of a heuristic that might prove to go a long way is: *shorter formulas are easier to understand than longer ones*. Finding the most useful heuristics is an important challenge for the experimental work in this area.

Other Information Presentation (IP) issues that our application gives rise to include aggregation and summarisation of information. For example, if a large number, say n, of locations need to be repaired, it would be cumbersome to spell this it in n separate steps (*"First move to location 1 and repair it; then move to location 2 and repair it; ...; finally move to location n and repair it."*). It is much better to combine these events in one phrase (saying "each location") and to leave out any obvious information. For instance, the system might simply say *"Repair each location, starting with the nearest one and ending with the one furthest away"*, which aggregates the repairs actions and leaves out the 'move' actions because they are inferable.

Beyond the elucidation of an individual plan, it is important to be able to explain why one plan is superior to another, and this raises Information Presentation (IP) challenges of its own. Consider Figure 1, for example. In an initial interaction it may be suitable to give a high-level explanation of the choice between the plans A and B, as in *"A is proposed because we wish to respect α's preference not to use crane c_2"*. However, the user may also want to ask questions about particular actions, such as why a certain crane was used or moved first, and this may require more detailed explanations. Since different users may have different information requirements, the system should make use of adaptation and user modelling, to decide about the level of abstraction, and the level of detail, that is required in a given situation. Managers, for example, may only require a small amount of high-level information, whereas others may need to scrutinise the plan in more detail, for example to make sure they agree with the assumptions on which it is based.

Traditionally, diagrams have played an important role in argumentation, with diagrams expressing networks of arguments supporting and attacking each other [3]. These diagrams, however, become cluttered when networks are large and would benefit from aggregation and summarization as well. Given that each modality has its own advantages [7, 8], we shall explore how language can complement these diagrams, for instance by placing text inside a diagram, or by using one modality at a meta-level with respect to the other. For example, a caption may be generated that highlights an important aspect of a diagram, for instance *"Several arguments attacked this claim, but each proved contentious"*.

Given that the dialogue occurs as dynamic process involving inter-dependent components, evaluation of these components is difficult, but critical. For example, it may be necessary to evaluate the scrutability on multiple levels of abstraction such as an entire plan, the argumentation in favour of a plan, and individual facts or arguments. The evaluation therefore forms a final challenge in this program of work.

Acknowledgements

This research has been carried out within the project "Scrutable Autonomous Systems" (SAsSY), funded by the Engineering and Physical Sciences Research Council (EPSRC, UK), grant ref. EP/J012084/1.

REFERENCES

[1] Rodney A. Brooks, 'Intelligence without representation', *Artificial Intelligence*, **47**, 139–159, (1991).
[2] Martin Caminada and Mikolaj Podlaszewski, 'Grounded semantics as persuasion dialogue', in *COMMA*, eds., Bart Verheij, Stefan Szeider, and Stefan Woltran, volume 245 of *Frontiers in Artificial Intelligence and Applications*, pp. 478–485. IOS Press, (2012).
[3] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games.', *Artificial Intelligence*, **77**, 321–257, (1995).
[4] Helmut Horacek, 'Generating inference-rich discourse through revisions of rst-trees', in *AAAI*, pp. 814–820, (1998).
[5] Rolando Medellin-Gasque, Katie Atkinson, Peter McBurney, and Trevor J. M. Bench-Capon, 'Arguments over co-operative plans', in *TAFA*, eds., Sanjay Modgil, Nir Oren, and Francesca Toni, volume 7132 of *Lecture Notes in Computer Science*, pp. 50–66. Springer, (2011).
[6] Yael Dahan Netzer, Michael Elhadad, and Ben Gurion, 'Generating determiners and quantifiers in hebrew', in *ACL workshop on Semitic Languages*, pp. 89–96, (1998).
[7] Jon Oberlander, Richard Cox, Padraic Monaghan, Keith Stenning, and Richard Tobin, 'Individual differences in proof structures following multimodal logic teaching', in *COGSCI*, pp. 201–206, (1996).
[8] M. Petre, 'Why looking isnt always seeing: Readership skills and graphical programming', *CACM*, **39**, 33–42, (1995).
[9] Luc De Raedt, Christian Bessière, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas, eds. *ECAI. Including Prestigious Applications of Artificial Intelligence (PAIS) System Demonstrations Track*, volume 242 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2012.
[10] Iyad Rahwan, Mohammed Iqbal Madakkatel, Jean-Francois Bonnefon, Ruqiyabi Naz Awan, and Sherief Abdallah, 'Behavioural experiments for assessing the abstract argumentation semantics for reinstatement', in *Cognitive Science*, pp. 1483–1502, (2010).
[11] Anand S. Rao, 'AgentSpeak(L): BDI agents speak out in a logical computable language', in *MAAMAW '96: Proceedings of the Seventh European workshop on Modelling autonomous agents in a multi-agent world : agents breaking away*, pp. 42–55, Eindhoven, The Netherlands, (1996).
[12] S. Russell and P Norvig, *Artificial Intelligence: A Modern Approach (Second Edition)*, Prentice Hall, 2003.
[13] S. M. Shieber, 'The problem of logical form equivalence', *Computational Linguistics*, **19**, 179–190, (1993).
[14] Xiantang Sun and Chris Mellish, 'An experiment on free generation from single rdf triples', in *ENLG*, pp. 105–108, (2007).
[15] Yuqing Tang, Timothy J. Norman, and Simon Parsons, 'A model for integrating dialogue and the execution of joint plans', in *AAMAS (2)*, eds., Carles Sierra, Cristiano Castelfranchi, Keith S. Decker, and Jaime Simão Sichman, pp. 883–890. IFAAMAS, (2009).
[16] Kees van Deemter, 'Structured meanings in computational linguistics', in *13th International Conf. on Computational Linguistics (COLING-90)*, ed., H.Karlgren, pp. 85–89, Helsinki, Finland, (1990).
[17] Wahlster, Marburger, Jameson, and Busemann, 'Over-answering yes-no questions', in *8th Int. Joint Conference on Artificial Intelligence (IJCAI-83)*, ed., A.Bundy, pp. 643–646, Karlsruhe, Germany, (1983).