Opportunities for Argument-centric Persuasion in Behaviour Change

Anthony Hunter[1]

Department of Computer Science, University College London,
Gower Street, London WC1E 6BT, UK

Abstract. The aim of behaviour change is to help people overcome specific behavioural problems in their everyday life (e.g. helping people to decrease their calorie intake). In current persuasion technology for behaviour change, the emphasis is on helping people to explore their issues (e.g. through questionnaires or game playing) or to remember to follow a behaviour change plan (e.g. diaries and email reminders). So explicit argumentation with consideration of arguments and counterarguments are not supported with existing persuasion technologies. With recent developments in computational models of argument, there is the opportunity for argument-centric persuasion in behaviour change. In this paper, key requirements for this will be presented, together with some discussion of how computational models of argumentation can be harnessed.

1 Introduction

Persuasion is an activity that involves one party trying to induce another party to believe something or to do something. It is an important and multifaceted human facility. For some occupations, persuasion is paramount (for example in sales, marketing, advertising, politics, etc). However, we are all confronted with the need to persuade others on a regular basis in order to get our job done, or to get our needs or wishes met.

Psychological studies show how there are many factors that can have a substantial influence on whether persuasion will be successful. For any given situation, these factors may include what we may describe as rational criteria such as the merits of what we are being persuaded to believe or do, or the reliability of the information that is being presented to us. But often, seemingly irrational criteria may become important in the success of persuasion such as whether we like the person who is trying to persuade us, or what our peers may think of us [1]. When it comes to products and services, seemingly trivial factors such as the type of packaging, the colour of a vendor's logo, or a celebrity endorsement, can make a big difference. Sales, marketing, advertising, politics certainly exploit these seemingly irrational criteria.

As computing becomes involved in every sphere of life, so too is persuasion a target for applying computer-based solutions. Persuasion technologies have come out of developments in human-computer interaction research (see for example the influential work by Fogg [2]) with a particular emphasis on addressing the need for systems to help people make positive changes to their behaviour, particularly in healthcare and healthy life styles. Over the past 10 years, a wide variety of systems have been developed to help users to control body weight [3], to reduce fizzy drink consumption [4], to increase physical exercise [5], and to decrease stress-related illness [6].

Many of these persuasion technologies for behaviour change are based on some combination of questionnaires for finding out information from users, provision of information for directing the users to better behaviour, computer games to enable users to explore different scenarios concerning their behaviour, provision of diaries for getting users to record ongoing behaviour, and messages to remind the user to continue with the better behaviour. These systems tend to be heavily scripted multi-media solutions and they are often packaged as websites and/or apps for mobile devices.

Interestingly, argumentation is not central to the current manifestations of persuasion technologies. The arguments for good behaviour seem either to be assumed before the user accesses the persuasion technology (e.g. when using diaries, or receiving email reminders), or arguments are provided implicitly in the persuasion technology (e.g. through provision of information, or through game playing). So explicit argumentation with consideration of arguments and counterarguments are not supported with existing persuasion technologies.

This creates some interesting opportunities for artificial intelligence, using computational models of argument, to develop persuasion technologies for behaviour change where arguments are central. This leads to an opportunity for what we may call *argument-centric persuasion for behaviour change*. Computational models of argument are beginning to offer ways to formalize aspects of persuasion, and with some adaptation and development, they have the potential to be incorporated into valuable tools for changing behaviours.

In argument-centric persuasion technologies, a system enters into a dialogue with a user to persuade them to undertake some action. An action might be abstract such as believing something, or deciding something, or it might be a physical action such as buying something, or eating something, or taking some medicine, or it might be to not do some physical action such as not buying something, or not eating something, etc. The dialogue may involve steps where the system finds out more about the users beliefs, intentions and desires, and where the system offers arguments with the aim of changing the users beliefs, intentions and desires. The system also needs to handle objections or doubts by the user represented by counterarguments with the aim of providing a dialectically winning position. To illustrate how a dialogue can lead to the presentation of an appropriate context-sensitive argument consider the following example.

Example 1. The system moves are odd numbered, and the user moves are even numbered: (1) Do you need a snack? (2) Yes. (3) What is available? (4) Cup cake or salad. (5) Which do you prefer? (6) Cup cake. (7) Why? (8) I need a sugar rush from the cup cake. (9) Why? (10) I need a sugar rush in order to do some work. (11) The sugar rush from a cup cake is brief. So your need for a sugar rush does not imply you need a cup cake. Do you agree? (12) Yes.

In this paper, we will identify the requirements of argument-centric persuasion as applied to behaviour change, review some of the key features of computational models of argument, and then consider a simple case study to illustrate how we might address behaviour change by harnessing a computational model of argument.

2 Requirements for argument-centric behaviour change

There is increasing demand for tools to support behaviour change. Many people need help in changing their behaviour on every day matters in some respect such as promoting healthy life styles (eating more fruit and veg, exercise), weight management (e.g. addressing overweight, bulimia, anorexia), addiction management (gambling, alcohol, drugs, etc), treatment compliance (e.g. self-management of diabetes), personal financial (e.g. borrowing less, saving more), education (starting or continuing with a course, studying properly), encouraging citizenship (e.g. voting, recycling, contributing to charities), safe driving (e.g. not exceeding speed limits, not texting while driving), addressing anti-social behaviour (e.g. aggressive behaviour, vandalism), etc. Further applications may include automated systems for responding to antisocial behaviour online (e.g. racism, sexism, etc).

In these behaviour change applications, the following are key requirements for argument-centric persuasion technologies.

(Requirement 1) Goal orientation The system should aim to be successful in its persuasion dialogue. This means that the system should aim for a dialogue that concludes with the user being persuaded (as manifested for example by the user agreeing to an intended action).

(Requirement 2) Context sensitivity The system should ask the user questions in order to take account of the user's context and concerns. This may include the user's preferences, mood, desires, etc, at the time of the dialogue. By understanding the user, the system is more likely to identify arguments that will be successful in persuading the user.

(Requirement 3) Maintaining engagement The system should aim to minimize the chances that the user disengages from the dialogue. This means that the system should try to avoid the user becoming bored, irritated, etc, by the dialogue. So the system may have to trade a longer sequence of moves that would be more likely to be convincing (if the user remained engaged) for a shorter and more engaging sequence of moves.

(Requirement 4) Language sensitivity The system should use a language in the dialogue (for instance, complexity, vocabulary, style, etc) that is appropriate for the user in the current context. This may include consideration of the user's preferences, mood, desires, etc. It may also include the user's attitude to authority, expertise, etc. For instance, for persuading a teenager to desist from anti-social behaviour, using the language of his/her music or sports idols is more likely to be successful than the language of a school teacher.

(Requirement 5) Argument quality The system should present arguments and counterarguments that are informative, relevant, and believable, to the user. If the system presents uninformative, irrelevant, or unbelievable arguments (from the perspective of the user), the probability of successful persuasion is reduced, and it may alienate the user.

(Requirement 6) Effort minimization The system should aim to minimize the effort involved on the part of the user. This means that the user should not be asked unnecessary questions or presented unnecessary claims, or arguments/counterarguments, etc.

In argument-centric persuasion for behaviour change the system enters into a dialogue with the user. In the short-term, we may envisage that this dialogue involves limited kinds of interaction in order for the user to offer specific kinds of information to the system, or for the system to ask specific queries of the user. The allowed dialogues moves are specified for the particular application. The dialogues between the system and the user are restricted to the moves specified at the interface (i.e. the graphical user interface) at each step of the dialogue. The user can initiate a dialogue, and then the system manages the dialogue by asking queries of the user, where the allowed answers are given by a menu, and by positing arguments. For instance, for a weight management application that is intended to help the user decrease calorie consumption, the kinds of queries and posits that can be made by the system might be the following.

- Query for contextual information. For example, for the query, "How many cakes have you eaten today", the menu of answers could have "0", "1", "2", "3", "4", and "more than 4", and for the query, "Which of the following best describe how you feel today", the menu of answers could have "very hungry", "moderately hungry", "not hungry", and "full".
- Query for preferences over options. For example, for the query, "Which would you prefer to eat as a snack now", the menu of answers could have "Prefer cup cake to carrot", "Prefer carrot to cup cake", and "Indifferent between cup cake and carrot".
- Rebuttals with explanation. For example, suppose the user has chosen the reply "Prefer cup cake to carrot", the system could rebut this by saying "Remember your plan to lose weight therefore prefer carrot to cup cake".
- Undercuts with explanation. For example, suppose the user has chosen "Prefer cup cake to carrot" because "Cup cake gives a sugar rush and I need this to work late" (i.e. the user has selected "Cup cake gives a sugar rush and I need this to work late" from the menu of answers), the system could undercut this by saying "The sugar rush from a cup cake is brief, and so it will not help you to work late".

So we can adopt some intuitive requirements for argument-centric behaviour change, and we can impose some simple constraints on the mode of interaction between the user and the system in order to render the approach viable in the short-term. Obviously richer natural language interaction would be desirable, but it is less feasible in the short-term.

3 Computational models of argument

Computational models of argument reflect aspects of how humans use conflicting information by constructing and analyzing arguments. Formalizations of argumentation have been extensively studied, and some basic principles established. We can group much of this work in four levels as follows.

Dialectical level Dialectics is concerned with determining which arguments win in some sense. In abstract argumentation, originally proposed in the seminal work by Dung [7], arguments and counterarguments can be represented by a graph. Each node node denotes an argument, and each arc denotes one argument attacking another argument. Dung then defined some principled ways to identify extensions of an argument graph. Each extension is a subset of arguments that together act as a coalition against attacks by other arguments. An argument in an extension is, in a sense, an acceptable argument.

Logical level At the dialectic level, arguments are atomic. They are assumed to exist, but there is no mechanism for constructing them. Furthermore, they cannot be divided or combined. To address this, the logical level provides a way to construct arguments from knowledge. At the logical level, an argument is normally defined as a pair $\langle \Phi, \alpha \rangle$ where Φ is a minimal consistent subset of the knowledgebase (a set of formulae) that entails α (a formula). Here, Φ is called the support, and α is the claim, of the argument. Hence, starting with a set of formulae, arguments and counterarguments can be generated, where a counterargument (an argument that attacks another argument) either rebuts (i.e. negates the claim of the argument) or undercuts (i.e. negates the support of the argument). A range of options for structured argumentation at the logic level have been investigated (see [8–11] for tutorial reviews of some of the key proposals).

Dialogue level Dialogical argumentation involves agents exchanging arguments in activities such as discussion, debate, persuasion, and negotiation. Dialogue games are now a common approach to characterizing argumentation-based agent dialogues (e.g. [12–22]). Dialogue games are normally made up of a set of communicative acts called moves, and a protocol specifying which moves can be made at each step of the dialogue. Dialogical argumentation can be viewed as incorporating logic-based argumentation, but in addition, dialogical argumentation involves representing and managing the locutions exchanged between the agents involved in the argumentation. The emphasis of the dialogical view is on the interactions between the agents, and on the process of building up, and analyzing, the set of arguments until the agents reach a conclusion.

Rhetorical level Normally argumentation is undertaken with some wider context of goals for the agents involved, and so individual arguments are presented with some wider aim. For instance, if an agent is trying to persuade another agent to do something, then it is likely that some rhetorical device is harnessed and this will affect the nature of the arguments used (e.g. a politician may refer to investing in the future of the nation's children as a way of persuading colleagues to vote for an increase in taxation). Aspects of the rhetorical level include believability of arguments from the perspective of the audience [23], impact of arguments from the perspective of the audience [24], use of threats and rewards [25], appropriateness of advocates [26], and values of the audience [27–29].

There are a number of proposals that formalize aspects of persuasion. Most are aimed at providing protocols for dialogues (for a review see [30]). Forms of correctness (e.g. the dialogue system has the same extensions under particular dialectical semantics as using the agent's knowledgebases) have been shown for a variety of systems (e.g. [31, 13, 15, 32]). However, strategies for persuasion, in particular taking into account beliefs of the opponent are under-developed.

There are a number of proposals for using probability theory in argumentation (e.g. [33–39]). For abstract argumentation, the epistemic approach involves assigning a probability value to each argument that denotes the degree to which it is believed [36, 38, 39]. An epistemic extension is then the set of arguments that have a probability value greater than 0.5. The justification approach involves a probability distribution over the subgraphs of the argument graph G [37, 38]. The probability that a set of arguments Γ is an extension (according to a particular dialectical semantics S such as grounded, preferred, etc) is the sum of the probability assigned to each subgraph G' of G for which Γ is an extension of G' (according the dialectical semantics S).

Probabilistic models of the opponent have been used in some strategies [40, 41] allowing the selection of moves for an agent based on what it believes the other agent believes. Utility theory has also been considered in argumentation (for example [42–45]) though none of these represents the uncertainty of moves made by each agent in argumentation. One approach that combines probability theory and utility theory (using decision theory) has been used in [46] to identify outcomes with maximum expected utility where outcomes are specified as particular arguments being included or excluded from extensions. Strategies in argumentation have also been analyzed using game theory [47–49], though these are more concerned with issues of manipulation, rather than persuasion.

So there is a range of formal systems for generating and comparing arguments and counterarguments, and for undertaking this within the context of a dialogue.

4 A simple case study

In order to illustrate how we can adapt computational models of argument for argument-centric persuasion for behaviour change, we will consider a simple case study for helping a user to decrease his or her calorie intake for weight management. Background knowledge for this application could for example be compiled empirically by monitoring behavioural change sessions with counsellors [50].

Computational models of argument normally consider two or more agents who exchange moves according to some protocol (which specifies what moves are necessary or permissible at each stage of a dialogue). These agents are equal participants in the dialogue, and there is no restriction in the range of arguments and counterarguments that they can make. For behaviour change, we have an asymmetric situation where one agent (the system) provides the queries, the arguments and the counterarguments, and the other agent (the user) replies to queries. In order to allow the user to make a counterargument, the counterargument will also be generated by the system, and presented to the user in a query. The user either agrees or disagrees with it as a reply. For example, if the system has made an argument such as "You should eat salad because it is good for you", the user could be queried as to whether it subscribes to the counterargument "Do you refuse to eat salad because you think it is boring to eat?". As we will see later, the replies made by the user are restricted to those available by a menu of answers generated by the system.

So we require a set of statements that can be used in the dialogues. Each statement is atomic and cannot be divided into substatements. Each is represented by a sentence of English (a sy-sentence) as illustrated next.

$\mathtt{s}_0$ = "User needs a meal"	$\mathtt{s}_1$ = "The best choice is a burger"
$\mathtt{s}_2$ = "The best choice is fish"	$\mathtt{s}_3$ = "User needs to slim for fun run"
$\mathtt{s}_4$ = "User needs to slim to be healthy"	$\mathtt{s}_5$ = "Fun run training uses lots of energy"
$\mathtt{s}_6$ = "User needs occassional treats"	$\mathtt{s}_7$ = "User has eaten 10 burgers this week"
$\mathtt{s}_8$ = "User wants a burger"	$\mathtt{s}_9$ = "User wants fish"
$\mathtt{s}_{10}$ = "Options are a burger or fish"	$\mathtt{s}_{11}$ = "System has failed to persuade user"
$\mathtt{s}_{12}$ = "User agrees with argument $\mathtt{a}_1$"	$\mathtt{s}_{13}$ = "User disagrees with argument $\mathtt{a}_1$"
$\mathtt{s}_{14}$ = "User agrees with argument $\mathtt{a}_2$"	$\mathtt{s}_{15}$ = "User disagrees with argument $\mathtt{a}_2$"
$\mathtt{s}_{16}$ = "User agrees with argument $\mathtt{a}_5$"	$\mathtt{s}_{17}$ = "User disagrees with argument $\mathtt{a}_5$"
$\mathtt{s}_{18}$ = "User agrees with argument $\mathtt{a}_4$"	$\mathtt{s}_{19}$ = "User disagrees with argument $\mathtt{a}_4$"

We also require a set of queries. Each query is atomic and cannot be divided into subqueries. Each is represented by a sentence of English (a wh-sentence) as illustrated next.

$\mathtt{q}_0$ = "Do you need a meal?"
$\mathtt{q}_1$ = "What are the options?"
$\mathtt{q}_2$ = "What do you prefer?"
$\mathtt{q}_3$ = "Do you agree with argument $\mathtt{a}_1$?"
$\mathtt{q}_4$ = "Do you agree with argument $\mathtt{a}_2$?"
$\mathtt{q}_5$ = "Why do you disagree with argument $\mathtt{a}_1$?"
$\mathtt{q}_6$ = "Why do you disagree with argument $\mathtt{a}_2$?"
$\mathtt{q}_7$ = "Do you agree with argument $\mathtt{a}_5$?"
$\mathtt{q}_8$ = "Do you agree with argument $\mathtt{a}_4$?"

In order to generate arguments, we require a logic. For our case study, we use a simple implicational logic. For this, we require a set of literals, and a set of rules of the form $\alpha_1 \wedge \ldots \wedge \alpha_m \rightarrow \beta_1 \wedge \ldots \wedge \beta_n$ where $\alpha_1, \ldots, \alpha_m$ and $\beta_1, \ldots, \beta_n$ are literals. Each literal is either a statement or an atom of the form $\mathtt{ok}(\mathtt{rule_i})$ where the index i is a number. For our example, the following are rules.

$$\mathtt{s}_3 \land \mathtt{ok(rule1)} \rightarrow \mathtt{s}_2 \land \neg \mathtt{s}_1$$
$$\mathtt{s}_4 \land \mathtt{ok(rule2)} \rightarrow \mathtt{s}_2 \land \neg \mathtt{s}_1$$
$$\mathtt{s}_5 \land \mathtt{ok(rule3)} \rightarrow \neg \mathtt{ok(rule1)}$$
$$\mathtt{s}_6 \land \mathtt{ok(rule4)} \rightarrow \neg \mathtt{ok(rule2)}$$
$$\mathtt{s}_7 \land \mathtt{ok(rule5)} \rightarrow \neg \mathtt{ok(rule4)}$$

Using the simple implicational logic, an argument is a tuple $\langle \Phi, \alpha \rangle$ where Φ contains one or more rules, and some literals, such that α can be obtained from Φ using modus ponens (but no other proof rule), and no subset of Φ entails α using modus ponens. An argument $\langle \Phi, \alpha \rangle$ attacks an argument $\langle \Psi, \beta \rangle$ iff there is a literal $\gamma \in \Psi$ such that $\{\alpha, \gamma\}$ is inconsistent according to classical logic. For our case study, the arguments and attacks are given in Figure 1.

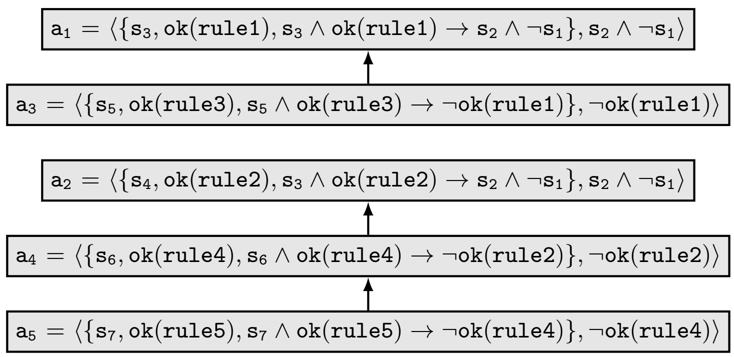

Fig. 1: The argument graph for the running example. Here there are two arguments for the goal of the persuasion (namely $\mathtt{a}_1$ and $\mathtt{a}_2$).

The system can make four types of dialogue move: (1) $\mathtt{Ask(q)}$ where $\mathtt{q}$ is a query; (2) $\mathtt{Posit(a)}$ where $\mathtt{a}$ is an argument; (3) $\mathtt{Attack(a, a')}$ where $\mathtt{a}$ and $\mathtt{a'}$ are arguments; and (4) $\mathtt{Claim(s)}$ where $\mathtt{s}$ is a statement. The user can only make one kind of move which is $\mathtt{Reply(s)}$ where $\mathtt{s}$ is a statement. An execution is a sequence of states where each state is a set of dialogue moves (as illustrated in Table 1).

The protocol is specified using a set dialogue rules. For each dialogue rule, the antecedent specifies when the rule can fire (i.e. the preconditions of the rule), and the consequent specifies the moves that follow from the firing of the rule (i.e. the postconditions of the rule).

For the conditions of dialogue rules, there are five types of predicate: $\mathtt{Start}$ which only holds in the initial state of the dialogue; $\mathtt{Asked(q)}$ which holds when the previous state has the move $\mathtt{Ask(q)}$; $\mathtt{Replied(s)}$ which holds if the previous state has the move $\mathtt{Reply(s)}$; $\mathtt{Know(a)}$ which holds if the system can use its knowledgebase together with the statements $\mathtt{s}$ which occur in moves $\mathtt{Reply(s)}$

Step	Moves
0	$\{\texttt{Ask}(\texttt{q}_0)\}$
1	$\{\texttt{Reply}(\texttt{s}_0)\}$
2	$\{\texttt{Ask}(\texttt{q}_1)\}$
3	$\{\texttt{Reply}(\texttt{s}_{10})\}$
4	$\{\texttt{Ask}(\texttt{q}_2)\}$
5	$\{\texttt{Reply}(\texttt{s}_8)\}$
6	$\{\texttt{Posit}(\texttt{a}_2), \texttt{Ask}(\texttt{q}_4)\}$
7	$\{\texttt{Reply}(\texttt{s}_{14})\}$
8	$\{\texttt{Claim}(\texttt{s}_2)\}$

Table 1: An execution of the dialogue rules for the running example.

in previous states to obtain the argument $\texttt{a}$; and $\texttt{Ucut}(\texttt{a}, \texttt{a}')$ which holds it $\texttt{a}$ is an argument that attacks argument a'.

A dialogue rule is of the form $\alpha_1 \vee \ldots \vee \alpha_m \Rightarrow \beta_1 \vee \ldots \vee \beta_n$ where each $\alpha_i \in \{\alpha_1, \ldots, \alpha_m\}$ is a conjunction of conditions (where conditions are of the form $\texttt{Start}$, $\texttt{Asked}(\texttt{q})$, $\texttt{Replied}(\texttt{s})$, $\texttt{Said}(\texttt{s})$, $\texttt{Know}(\texttt{a})$, and $\texttt{Ucut}(\texttt{a}, \texttt{a}')$), and each β_i in $\{\beta_1, \ldots, \beta_n\}$ is a conjunction of moves. An example of a protocol containing a number of dialogue rules is given in Figure 2. By editing the dialogue rules, different protocols (and hence different state models) can be obtained.

A dialogue rule fires if the conditions $\alpha_1, \ldots, \alpha_m$ hold in the current state. If it fires, then one of $\beta_1, \ldots, \beta_n$ gives the moves for the next state. In this way, the dialogue rules are executable. The choice of which β_i in $\beta_1, \ldots, \beta_n$ to use for the next state is non-deterministic. If the dialogue rule is a system dialogue rule, then the system chooses, otherwise the user chooses. An execution terminates when there are no more moves that can be made by either the user or the system. The set of all executions for a protocol can be arranged as a state model as in Figure 3.

We now consider the user model (i.e. the model of the user which reflects the estimated uncertainty about what the user knows and believes). The first kind of uncertainty to consider is *perceptual uncertainty* which is the uncertainty about which arguments and attacks the user is aware of. The second kind of uncertainty is *epistemic uncertainty* which is uncertainty about which arguments and attacks the user believes.

From the system perspective, perceptual uncertainty concerns what arguments and attacks the user might start with. For instance, returning to Example 1, the system might not know if the user is aware of the fact the sugar rush from a cup cake is brief, and hence whether the user is aware of arguments based on it. We assume that once the system has told the user about an argument or attack, the user is aware of it, and hence that uncertainty falls to zero. Then from the system perspective, epistemic uncertainty concerns which arguments or attacks the user chooses to ignore because they do not believe them. Recent developments in probabilistic argumentation (e.g. [35–39] offer possibilities for capturing perceptual and epistemic uncertainty in user models. Furthermore,

$$
\begin{aligned}
\texttt{r}_0 &= \texttt{Start} \Rightarrow \texttt{Ask}(\texttt{q}_0)\\
\texttt{r}_1 &= \texttt{Replied}(\texttt{s}_0) \Rightarrow \texttt{Ask}(\texttt{q}_1)\\
\texttt{r}_2 &= \texttt{Replied}(\texttt{s}_{10}) \Rightarrow \texttt{Ask}(\texttt{q}_2)\\
\texttt{r}_3 &= \texttt{Replied}(\texttt{s}_8) \Rightarrow (\texttt{Posit}(\texttt{a}_1) \land \texttt{Ask}(\texttt{q}_3)) \lor (\texttt{Posit}(\texttt{a}_2) \land \texttt{Ask}(\texttt{q}_4))\\
\texttt{r}_4 &= \texttt{Replied}(\texttt{s}_9) \lor \texttt{Replied}(\texttt{s}_{12}) \lor \texttt{Replied}(\texttt{s}_{14}) \lor \texttt{Replied}(\texttt{s}_{16}) \Rightarrow \texttt{Claim}(\texttt{s}_2)\\
\texttt{r}_5 &= \texttt{Replied}(\texttt{s}_{13}) \Rightarrow \texttt{Ask}(\texttt{q}_5)\\
\texttt{r}_6 &= \texttt{Replied}(\texttt{s}_{15}) \Rightarrow \texttt{Ask}(\texttt{q}_6)\\
\texttt{r}_7 &= \texttt{Replied}(\texttt{s}_5) \land \texttt{Know}(\texttt{a}_3) \land \texttt{Ucut}(\texttt{a}_3, \texttt{a}_1) \Rightarrow (\texttt{Posit}(\texttt{a}_3) \land \texttt{Attack}(\texttt{a}_3, \texttt{a}_1) \land \texttt{Claim}(\texttt{s}_1))\\
\texttt{r}_8 &= \texttt{Replied}(\texttt{s}_6) \land \texttt{Know}(\texttt{a}_4) \land \texttt{Ucut}(\texttt{a}_4, \texttt{a}_3) \Rightarrow (\texttt{Posit}(\texttt{a}_4) \land \texttt{Attack}(\texttt{a}_4, \texttt{a}_1) \land \texttt{Ask}(\texttt{q}_8))\\
\texttt{r}_9 &= \texttt{Replied}(\texttt{s}_{18}) \land \texttt{Know}(\texttt{a}_5) \land \texttt{Ucut}(\texttt{a}_5, \texttt{a}_4) \Rightarrow (\texttt{Posit}(\texttt{a}_5) \land \texttt{Attack}(\texttt{a}_5, \texttt{a}_4) \land \texttt{Ask}(\texttt{q}_7))\\
\texttt{r}_{10} &= \texttt{Replied}(\texttt{s}_{17}) \Rightarrow \texttt{Claim}(\texttt{s}_{11})\\
\texttt{r}_{11} &= \texttt{Asked}(\texttt{q}_0) \Rightarrow \texttt{Reply}(\texttt{s}_0)\\
\texttt{r}_{12} &= \texttt{Asked}(\texttt{q}_1) \Rightarrow \texttt{Reply}(\texttt{s}_{10})\\
\texttt{r}_{13} &= \texttt{Asked}(\texttt{q}_2) \Rightarrow \texttt{Reply}(\texttt{s}_8) \lor \texttt{Reply}(\texttt{s}_9)\\
\texttt{r}_{14} &= \texttt{Asked}(\texttt{q}_3) \Rightarrow \texttt{Reply}(\texttt{s}_{12}) \lor \texttt{Reply}(\texttt{s}_{13})\\
\texttt{r}_{15} &= \texttt{Asked}(\texttt{q}_4) \Rightarrow \texttt{Reply}(\texttt{s}_{14}) \lor \texttt{Reply}(\texttt{s}_{15})\\
\texttt{r}_{16} &= \texttt{Asked}(\texttt{q}_5) \Rightarrow \texttt{Reply}(\texttt{s}_5)\\
\texttt{r}_{17} &= \texttt{Asked}(\texttt{q}_6) \Rightarrow \texttt{Reply}(\texttt{s}_6)\\
\texttt{r}_{18} &= \texttt{Asked}(\texttt{q}_7) \Rightarrow \texttt{Reply}(\texttt{s}_{16}) \lor \texttt{Reply}(\texttt{s}_{17})\\
\texttt{r}_{19} &= \texttt{Asked}(\texttt{q}_8) \Rightarrow \texttt{Reply}(\texttt{s}_{18})
\end{aligned}
$$

Fig. 2: Protocol for the running example. Rules $\texttt{r}_0$ to $\texttt{r}_{10}$ are for the system for querying the user, for presenting arguments and counterarguments, and for making claims. Rules $\texttt{r}_{11}$ to $\texttt{r}_{18}$ are for the user to respond to queries. The first state is obtained using $\texttt{r}_0$.

there are a number of possibilities for adapting recent proposals for strategies based on uncertain opponent models (e.g. [51, 40, 41, 46]) that may provide appropriate strategies for argument-centric persuasion for behaviour change.

Any dialogue that is generated by the protocol in this simple case study (i.e. Figure 3) involves few moves. However, for practical applications, the protocol would be substantially larger, and so some dialogues are potentially long. Long dialogues are much less likely to be followed by the user to the intended termination. Rather, the user will drop out, with the probability of dropping rising as the dialogue progresses. To address this problem, the user model can be used to determine what moves can be made by the system that are more likely to be successful with a smaller number of moves. This is a topic that hitherto has not been address in computational models of argument, though it does appear that it could be addressed using adaptation of probabilistic techniques for dialogical argumentation (as such as proposed in [40, 41, 46]).

In conclusion, we may be able to harness and adapt a number of established ideas in computational models of argument (at the dialectical, logical, and dialogue levels) to formalism the persuasion dialogues for behaviour change. Dialogue rules provide a flexible framework for specifying protocols. These can be easily changed whilst leaving the underlying algorithm for their execution

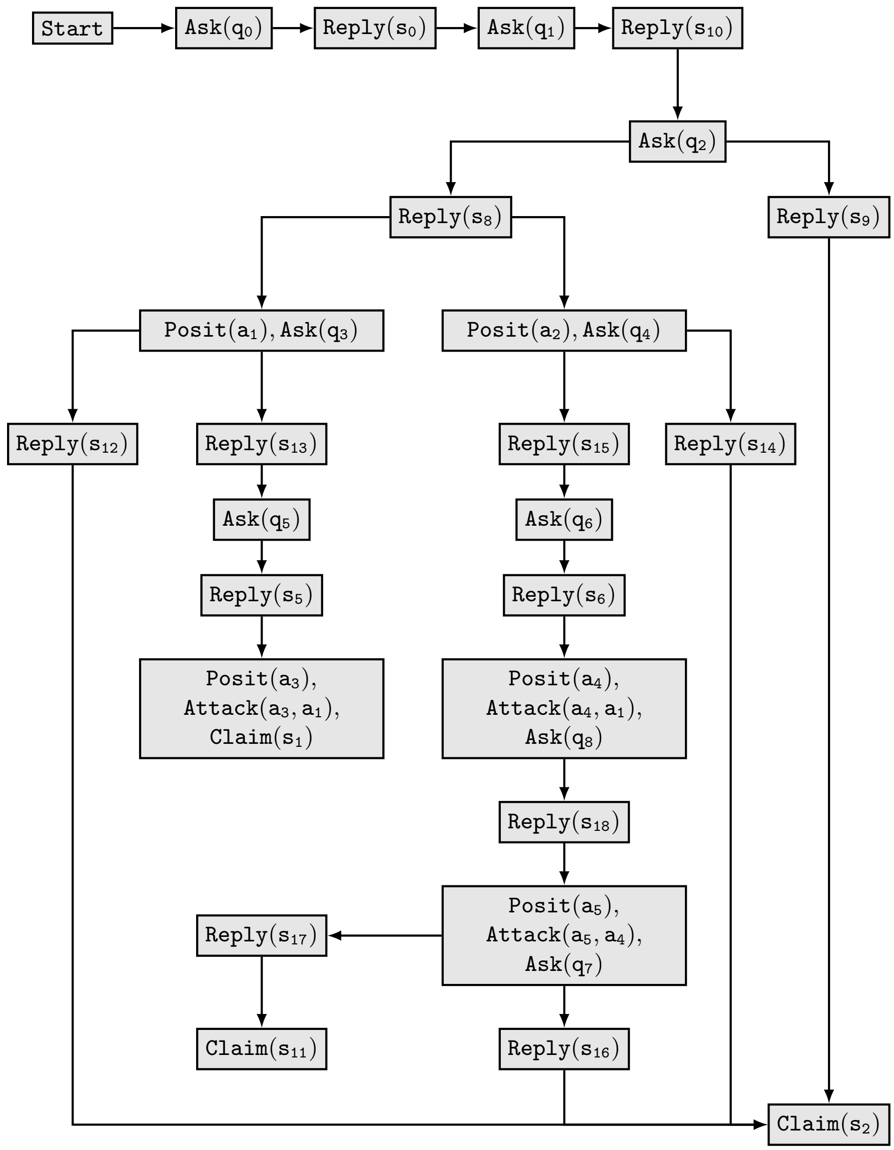

Fig. 3: State model for the protocol given in Figure 2. `start` gives the starting state. Each terminal state is a state without an exiting arc. Each state contains the moves that are made at that step of the dialogue. So multiple moves in a state means that multiple moves are made by the agent at that step in the dialogue. The `Reply` moves are made by the user. All other moves are made by the system.

unchanged. This idea of state-based execution of rules comes from a proposal for using executable logic for dialogical argumentation systems [52, 53]. Development of intelligent strategies based on opponent modelling does require further research, but there are promising proposals that could be further developed.

5 Discussions

There are many situations where bringing about behaviour change is potentially of great benefit. By making argumentation central to the persuasion, we may be able to deliver technology that is more persuasive and more useful than existing approaches to persuasion technologies. This is a valuable opportunity for the application of computational models of argument. Computational models of argument offer theories for understanding argumentation, and technologies for participating in argumentation, for the dialectical, logical, and dialogical levels. With recent developments in user modelling (based on probabilistic argumentation), and strategies for argumentation, there is progress in the rhetorical level. Taken together, these offer a range of possibilities for designing and implementing argument-centric persuasion technology for behaviour change.

References

1. Cialdini, R.: Influence: The Psychology of Persuasion. HarperCollins (1984)
2. Fogg, B.: Persuasive computers. In: Proceedings of the SIGCHI Conference on Human Factors in Computings Systems, CHI (1998) 225–232
3. Lehto, T., Oinas-Kukkonen, H.: Persuasive features in six weight loss websites: A qualitative evaluation. In: Persuasive technology 2010. Volume 6137 of LNCS., Springer (2010) 162–17
4. Langrid, S., Oinas-Kukkonon, H.: Less fizzy drinks: A multi-method study of persuasive reminders. In: Persuasive Technology 2012. Volume 7284 of LNCS., Springer (2012) 256–261
5. Zwinderman, M., Shirzad, A., Ma, X., Bajracharya, P., Sandberg, H., Kaptein, M.: Phone row: A smartphone game designed to persuade people to engage in moderate-intensity physical activity. In: Persuasion Technology 2012. Volume 7822 of LNCS., Springer (2012) 55–66
6. Kaipainen, K., Mattila, E., Kinnunen, M., Korhonen, I.: Facilitation of goal-setting and follow-up in internet intervention for health and wellness. In: Persuasion Technology 2010. Volume 7822 of LNCS., Springer (2010) 238–249
7. Dung, P.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. Artificial Intelligence **77** (1995) 321–357
8. Besnard, P., Hunter, A.: Constructing argument graphs with deductive arguments: a tutorial. Argument and Computation **5**(1) (2014) 5–30
9. S.Modgil, Prakken, H.: The aspic+ framework for structured argumentation: a tutorial. Argument and Computation **5**(1) (2014) 31–62
10. F.Toni: A tutorial on assumption-based argumentation. Arument and Computation **5**(1) (2014) 89–117

11. Garcia, A., Simari, G.: Defeasible logic programming: Delp-servers, contextual queries, and explanations for answers. Argument and Computation **5**(1) (2014) 63–88
12. Amgoud, L., Maudet, N., Parsons, S.: Arguments, dialogue and negotiation. In: Fourteenth European Conference on Artifcial Intelligence (ECAI 2000), IOS Press (2000) 338–342
13. Black, E., Hunter, A.: An inquiry dialogue system. Autonomous Agents and Multi-Agent Systems **19**(2) (2009) 173–209
14. Dignum, F., Dunin-Keplicz, B., Verbrugge, R.: Dialogue in team formation. In: Issues in Agent Communication. Springer (2000) 264–280
15. Fan, X., Toni, F.: Assumption-based argumentation dialogues. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'11). (2011) 198–203
16. Hamblin, C.: Mathematical models of dialogue. Theoria **37** (1971) 567–583
17. Mackenzie, J.: Question begging in non-cumulative systems. Journal of Philosophical Logic **8** (1979) 117–133
18. McBurney, P., Parsons, S.: Games that agents play: A formal framework for dialogues between autonomous agents. Journal of Logic, Language and Information **11** (2002) 315–334
19. McBurney, P., van Eijk, R., Parsons, S., Amgoud, L.: A dialogue-game protocol for agent purchase negotiations. Journal of Autonomous Agents and Multi-Agent Systems **7** (2003) 235–273
20. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of some formal inter-agent dialogues. J. of Logic and Comp. **13**(3) (2003) 347–376
21. Prakken, H.: Coherence and flexibility in dialogue games for argumentation. J. of Logic and Comp. **15**(6) (2005) 1009–1040
22. Walton, D., Krabbe, E.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. SUNY Press (1995)
23. Hunter, A.: Making argumentation more believable. In: Proceedings of AAAI 2004, MIT Press (2004) 269–274
24. Hunter, A.: Towards higher impact argumentation. In: Proceedings of AAAI 2004, MIT Press (2004) 275–280
25. Amgoud, L., Prade, H.: Formal handling of threats and rewards in a negotiation dialogue. In: Proceedings of AAMAS. (2005) 529–536
26. Hunter, A.: Reasoning about the appropriateness of proponents for arguments. In: Proceedings of AAAI. (2008) 89–94
27. Bench-Capon, T.: Persuasion in practical argument using value based argumentationframeworks. Journal of Logic and Computation **13**(3) (2003) 429–448
28. Bench-Capon, T., Doutre, S., Dunne, P.: Audiences in argumentation frameworks. Artificial Intelligence **171**(1) (2007) 42–71
29. Oren, N., Atkinson, K., Li, H.: Group persuasion through uncertain audience modelling. In: Proceedings of the International Comference on Computational Models of Argument (COMMA'12). (2012) 350–357
30. Prakken, H.: Formal sytems for persuasion dialogue. Knowledge Engineering Review **21**(2) (2006) 163–188
31. Prakken, H.: Coherence and flexibility in dialogue games for argumentation. Journal of Logic and Computation **15**(6) (2005) 1009–1040
32. Caminada, M., Podlaszewski, M.: Grounded semantics as persuasion dialogue. In: Computational Models of Argument (COMMA'12). (2012) 478–485
33. Haenni, R.: Cost-bounded argumentation. International Journal of Approximate Reasoning **26**(2) (2001) 101–127

34. Dung, P., Thang, P.: Towards (probabilistic) argumentation for jury-based dispute resolution. In: Computational Models of Argument (COMMA'10), IOS Press (2010) 171–182
35. Li, H., Oren, N., Norman, T.: Probabilistic argumentation frameworks. In: Proceedings of the First International Workshop on the Theory and Applications of Formal Argumentation (TAFA'11). (2011) 1–16
36. Thimm, M.: A probabilistic semantics for abstract argumentation. In: Proceedings of the European Conference on Artificial Intelligence (ECAI'12). (2012) 750–755
37. Hunter, A.: Some foundations for probabilistic argumentation. In: Proceedings of the International Comference on Computational Models of Argument (COMMA'12). (2012) 117–128
38. Hunter, A.: A probabilistic approach to modelling uncertain logical arguments. International Journal of Approximate Reasoning **54**(1) (2013) 47–81
39. Hunter, A., Thimm, M.: Probabilistic argumentation with incomplete information. In: Proceedings of ECAI. (2014) (in press).
40. Rienstra, T., Thimm, M., Oren, N.: Opponent models with uncertainty for strategic argumentation. In: Proceedings of IJCAI'13, IJCAI/AAAI (2013)
41. Hunter, A.: Modelling uncertainty in persuasion,. In: Proceedings of the International Conference on Scalable Uncertainty Management (SUM'13). Volume 8078 of LNCS., Springer (2013) 57–70
42. Rahwan, I., Larson, K.: Pareto optimality in abstract argumentation. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008), AAAI Press (2008)
43. Riveret, R., Prakken, H., Rotolo, A., Sartor, G.: Heuristics in argumentation: A game theory investigation. In: Computational Models of Argument (COMMA 2008). Volume 172 of Frontiers in Artificial Intelligence and Applications., IOS Press (2008) 324–335
44. Matt, P., Toni, F.: A game-theoretic measure of argument strength for abstract argumentation. In: Logics in A.I. Volume 5293 of LNCS. (2008) 285–297
45. Oren, N., Norman, T.: Arguing using opponent models. In: Argumentation in Multi-agent Systems. Volume 6057 of LNCS. (2009) 160–174
46. Hunter, A., Thimm, M.: Probabilistic argument graphs for argumentation lotteries. In: Computational Models of Argument (COMMA'14), IOS Press (2014)
47. Rahwan, I., Larson, K.: Mechanism design for abstract argumentation. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'08, IFAAMAS (2008) 1031–1038
48. Rahwan, I., Larson, K., Tohmé, F.: A characterisation of strategy-proofness for grounded argumentation semantics. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09). (2009) 251–256
49. Fan, X., Toni, F.: Mechanism design for argumentation-based persuasion. In: Computational Models of Argument (COMMA'12). (2012) 322–333
50. Narita, T., Kitamura, Y.: Persuasive conversational agent with persuasion tactics. In: Persuasive 2010. Volume 6137 of LNCS. (2010) 1526
51. Hadjinikolis, C., Siantos, Y., Modgil, S., Black, E., McBurney, P.: Opponent modelling in persuasion dialogues. In: Proceedings of IJCAI. (2013)
52. Black, E., Hunter, A.: Executable logic for dialogical argumentation. In: European Conf. on Artificial Intelligence (ECAI'12), IOS Press (2012) 15–20
53. Hunter, A.: Analysis of dialogical argumentation via finite state machines. In: Scalable Uncertainty Methods. Volume 8078 of LNCS., Springer (2013) 1–14