Argument and Computation, 2013
Vol. 4, No. 2, 151–179, http://dx.doi.org/10.1080/19462166.2013.764927

Taylor & Francis
Taylor & Francis Group

Strategies for question selection in argumentative dialogues about plans

Rolando Medellin-Gasque[a]*, Katie Atkinson[a], Trevor Bench-Capon[a] and Peter McBurney[b]

[a]*Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK;*
[b]*Department of Informatics, King's College London, London, WC2R 2LS, UK*

(*Received 23 July 2012; final version received 2 January 2013*)

In this article, we present a proposal to enable agents to discuss the suitability of plans based on an argumentation scheme and associated critical questions. Agents coordinate their beliefs, intentions and preferences using a dialogue game based on this argumentation scheme and its critical questions. The detail encompassed by the argumentation scheme means that there is a large number of critical questions, and so dialogues may in principle be very lengthy. To improve the efficiency of dialogues, we present two strategies for selecting questions. We have implemented the system and, here, presented results showing how both strategies are effective in reducing the number of questions required to reach agreement, although their relative effectiveness is dependent on characteristics of the problem domain.

Keywords: argumentation-based dialogues; dialogue games; agent co-operation; argumentation schemes; critical questions

1. Introduction

The complexity of distributed systems limits the use of single-agent planning strategies in distributed problems because the local beliefs of an agent are often not sufficient to generate a satisfactory plan. A common assumption in traditional artificial intelligence (AI) planning is that the planner has accurate and complete knowledge of the world and the capabilities of other agents (Durfee 2001). Since this assumption is rarely satisfied when using multi-agent systems, structured argumentative dialogues have been proposed to co-ordinate plan-related tasks. In this article, we consider the problem of selecting from a large set of critical questions in dialogues about the refinement and selection of a plan. A strategy is necessary to provide autonomous agents with a mechanism to identify, prioritise and present relevant critical questions in an argumentative dialogue.

In the approach presented, agents coordinate their beliefs and intentions with the use of a strategy using a dialogue game based on an argumentation scheme and a set of related critical questions for plan proposals. We use the argumentation scheme for plan proposals presented in Medellin-Gasque, Atkinson, McBurney, and Bench-Capon (2011) and the related critical questions presented in Medellin-Gasque, Atkinson, and Bench-Capon (2012b).

The scheme has a large number of elements, and consequently the set of critical questions is necessarily large, so choosing an appropriate question in the dialogue becomes an important issue in terms of dialogue and cooperation efficiency. We will empirically demonstrate that selecting questions according to an appropriate strategy leads agents to cooperate more effectively. When selecting questions, agents consider several factors, including the context in which the dialogue develops and the nature of the questions. Our understanding of an appropriate question is one that is valid according to the protocol rules, and contributes either to the defeat of an existing

*Corresponding author. Email: medellin@liverpool.ac.uk argument or leads to new information needed to reach agreement. Using these principles, agents can pose and resolve critical questions in a descending order of priority to promote efficiency while maintaining focus and relevance.

The critical questions draw attention to potential inconsistencies in the proposal, and other alternative ways of reaching the goal. The analysis identified 65 critical questions that match the argumentation scheme where each question represents a way to question and/or attack the plan proposal. The large number of questions is necessary to cover the potentially many differing details (such as current circumstances, possible actions, their effects and timing, relation between actions and their timing, preferences) that make planning such an intricate, fine-grained process.

The need to cover all aspects of planning, especially durative actions and their combinations mean that the scheme needs many components and consequently there are many questions. In a given application, it is possible to make simplifying assumptions which will reduce the number of questions, but different applications will require different assumptions, and so the full list should be available.

Different questions become relevant at different times in the dialogue; for example, a question may presuppose a particular answer to a previous question. We identify here some of the factors that make questions relevant at some point in a dialogue and construct two strategies based on these factors. To establish the relevance of our approach, we implement two agents that engage in a dialogue where agents have different views of the world and use the strategies to select their critical questions.

The main contribution of this article is that we show how critical questions implemented in a dialogue game, together with a strategy to choose relevant questions, is beneficial both to the quality of the dialogue and the plan which results from it. In contrast to existing work that is largely theoretical, our novel implementation enables us to produce empirical results showing the benefits of our approach in teasing out the points of disagreement to come to an agreement on the best plan.

The remainder of the article is structured as follows: Section 2 describes a representative scenario where we believe this approach is applicable and the formal plan proposal used with examples of critical questions that match the scheme. Section 3 presents an overview of the dialogue game we have developed on the basis of this scheme with an example of the syntax and semantics of the protocol. Section 4 presents two strategies to select critical questions. Section 5 presents an example to illustrate our approach and gives details of the implementation of the dialogue game and the strategies. In Section 6, we present the results of our experiments together with an analysis of the results. Section 7 discusses some related work and we conclude in Section 8.

2. Argumentative approach to plan proposals

Complex, real-world domains require traditional approaches to AI planning to be rethought. Cooperative distributed planning focuses on how planning can be extended into a distributed environment, where the process of creating and executing a plan can involve actions and interactions of a number of participants (desJardins, Durfee, Ortiz, and Wolverton 2000). Co-operative distributed planning is typically carried out by agents which have shared objectives and representations of the environment but may differ in their beliefs and preferences.

An approach to distributed AI planning is presented in Figure 1, where agents first merge their knowledge bases through an inquiry or deliberation dialogue and then create a plan. Once the knowledge bases have been merged, the planning can be done as if there were a single agent until the point where the agents need to consider preferences over actions. Therefore, in this approach, distribution is not relevant to the specific task of plan creation and preferences are not placed in the dialogue in a natural way.

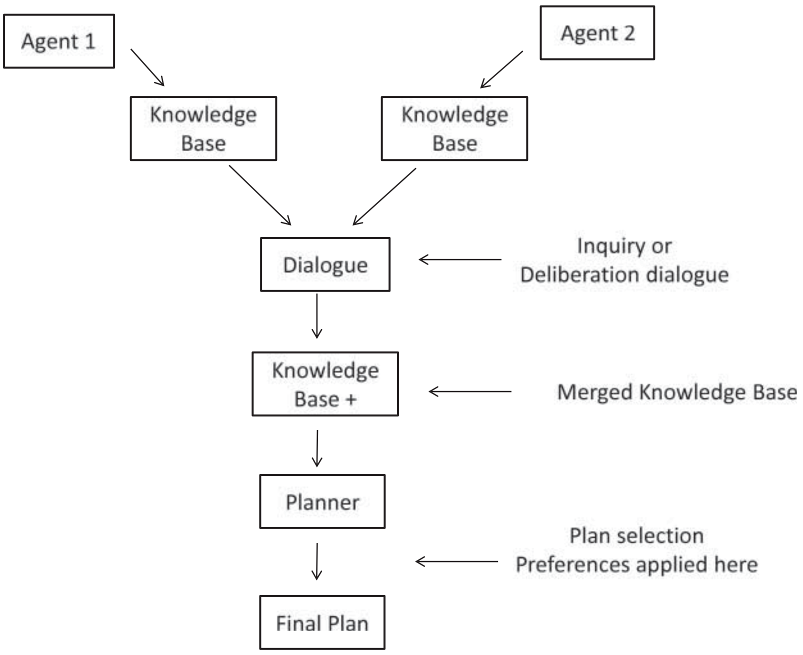

Figure 1. A distributed AI approach to planning.

Some research in the distributed planning has been focused on mechanisms for plan coordination (Durfee 2001); we propose here the use of argumentation-based dialogues to critique plans. Plan critique has been discussed in, for example, Wilkins and Myers (1998), where the critiques are part of a level in a hierarchical task network but preferences are not part of the critique.

Our dialogue approach focuses on critiquing plans, taking into account that agents have different beliefs about the world and different preferences. Preferences over plans are used to try to identify the best-possible plan taking into account the interests of both agents. In distributed planning techniques, there is no clear agreement as to where and when agents apply preferences over actions or plans. We believe that our approach presents an advantage because the preferences are applied in the dialogue together with the mechanism to evaluate and select the plans rather than as a separate process. Figure 2 presents a high-level schema of our approach in which the distribution remains important during the planning process itself.

AI has become increasingly interested in argumentation schemes due to their potential for making significant improvements in the reasoning capabilities of artificial agents and for automation of agent interactions (Bench-Capon and Dunne 2007) by guiding dialogue protocols. In essence, argumentation is a system for resolving conflicts in terms of the acceptability of the arguments that support the conflicting statements.

Argumentation schemes are stereotypical patterns of defeasible reasoning where arguments are presented as general inference rules from which, given a set of premises, a conclusion can be presumptively drawn (Prakken 2010). We extend research on practical reasoning using argumentation schemes and critical questions (Atkinson, Bench-Capon, and McBurney 2006) and extend it to plans. We build from the practical reasoning scheme for action proposals in Atkinson et al. (2006) to justify plan proposals. Based on this model, we develop a plan critique based on the elements of the scheme so as to give an added value to the process of selecting a plan. The added value is based on the fact that agents use their preferences to agree on a plan and the use of a

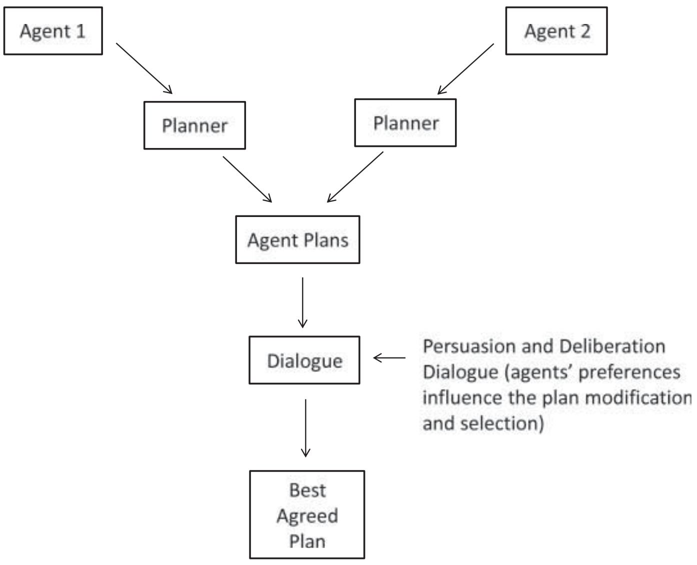
Agent 1
Agent 2
Planner
Planner
Agent Plans
Dialogue
Persuasion and Deliberation Dialogue (agents' preferences influence the plan modification and selection)
Best Agreed Plan

Figure 2. Argumentative approach to plan selection.

strategy to prioritise questions. Our plan proposal ASP can be expressed as an argumentation scheme as follows (full details of the argumentation scheme are given in Medellin-Gasque et al. (2011)):

Given a social context[1] and
current circumstances[2]
in which a set of preconditions hold,
a plan PL should be executed
to achieve new circumstances,
causing a set of postconditions to hold,
which will realize the plan-goal,
which will promote a set of values[3]

We use action-based alternating transition systems (AATS) as introduced in van der Hoek, Roberts, and Wooldridge (2007) as a semantic basis for our formalism to represent action and plan proposals. AATS models define joint-actions that may be performed by agents in a state and the effects of these actions. In particular, an AATS model defines semantic structures useful to represent joint-actions for multiple agents, their preconditions and the states that will result from the transition (Appendix 1).

In Table 1, we present our Argumentation Scheme for Plan proposals (ASP) and its expression in the AATS model representation. A valid instantiation of the scheme presupposes the existence of a regulatory environment or a social context X in which the proponent-agent has some rights to engage in a dialogue with the cooperating agent. Current circumstances are represented by the initial state $q_0 \in Q$, where Q is a set of states. An agent could instantiate the scheme to propose plan PL as a finite set of related action combinations partially ordered with respect to time. The plan leads to a state q_y in which postconditions hold and the plan-goal G is achieved (where G is an assignment of truth values to a set of propositions $p_G \subseteq \Phi$) and a non-empty set of values V_G is promoted/demoted. An example of how our plan proposal could be instantiated is as follows:

- Given the problem that two robot agents need to reach a zone travelling together with constraints in the paths,

Table 1. Plan proposal ASP and the AATS model representation.

Plan proposal	Using an AATS model
Given a social context X, in the current circumstances q_x in which preconditions $\pi(q_x)$ hold plan PL should be performed to achieve new circumstances q_y in which postconditions $\pi(q_y)$ will hold. This will realise the plan-goal G which will promote value(s) V_G.	Given social context X, In the initial state $q_0 = q_x \in Q$, where $\pi(q_0)$, agents $a, b \in Ag$ should execute plan PL, where PL is a finite set of related joint-actions j_n partially ordered with relation to time such that PL $= \{j_0, \ldots, j_n\}$ and $\{j_0, \ldots, j_n\} \in J_{Ag}$ and $j_n = \{\alpha_a, \ldots, \alpha_b\}$ with transition given by $\tau(q_x, \mathrm{PL})$ is q_y, where $\tau(q_0, \{j_1, \ldots, j_n\}) = \tau(\tau(q_0, j_1), (j_2, \ldots, j_n))$ and $\tau(q_x, \{\}) = q_x$ such that $p_G \in \pi(q_x)$ and $p_G \notin \pi(q_y)$, where $G = p$ and ($V_G \subseteq V$ such that $v_1 \in V_G$ iff $\delta(q_x, q_y, v_1)$ is $+$) and $V_G \neq \emptyset$

- agent 1 has the authority to propose a plan (given by the *social context*),
- agent 1 believes the agents are in zone A (*current circumstances*),
- agents are ready to leave at time *t* (*preconditions*),
- agents need to perform together plan PL_n consisting of a sequence of joint actions assigning actions to both of them,
- to reach zone D (*new circumstances, goal*),
- to promote efficiency and safety (*values*).

2.1. *Critical questions for the plan proposal scheme*

In Walton (2005), Walton explains *…arguments need to be examined within the context of an ongoing investigation in dialogue in which questions are being asked and answered.* Critical questions provide a way to examine the acceptability of arguments instantiating schemes. Different critical questions are used to critique different aspects of the argument. A benefit of having critical questions associated with an argument scheme is that the questions enable dialogue participants to identify points of challenge to an argument or locate premises and assumptions in the argument that can be recognised as questionable. Questions can take the form of attacks if the agents provide evidence to back up the claim. A weak question challenges an assumption rather than denying a claim. This is a request for more information and does not incur any commitments (unlike a strong attack).

Critical questions can be used as a basis on which to create rule-governed interaction protocols called ‘Dialogue Games’ for agents where the participants put forward arguments (instantiating the argumentation scheme) and opponents of the argument challenge it (by instantiating critical questions) as in, for example, Atkinson, Bench-Capon, and McBurney (2005) and Heras, Navarro, Botti, and Julián (2010). Essentially, the moves of the game correspond to the critical questions. Argumentation-based dialogues are used then to formalise dialogues between autonomous agents based on theories of argument exchange. We classify our set of 65 critical questions for the plan proposal scheme ASP into seven layers according to the aspect of the proposal which is challenged:

Layer 1: An action and its elements (lowest level).
Layer 2: The timing of a particular action.
Layer 3: The way actions are combined.

Layer 4: The plan proposal overall.
Layer 5: The timing of the plan proposal.
Layer 6: Side effects.
Layer 7: Alternative options (highest level).

We now discuss each layer and present some examples of questions. The full list is presented in Medellin-Gasque et al. (2012b).

Layer 1. An action and its elements (*16 questions*).
In this layer, questions aim to find inconsistencies for a particular action, challenging the validity and possibility of the action elements. The validity of an *action* and an *action element* is given by the existence of the action definition in the agent's action repertoire, thus an agent challenging the validity of an action or an action element says that the action has no definition in the agents' representation of the world. We use the 'durative action' representation from the Planning Domain Definition Language (PDDL) 2.1 planning specification in Fox and Long (2003) to question action elements from a different perspective from the one used in Atkinson et al. (2006). The PDDL 2.1 specification is an extension to PDDL[4] for expressing temporal planning domains. Durative actions are defined with elements such as action duration, invariant conditions, termination conditions, start effects and end effects. Further details on the formalisation of the durative action in our plan proposal could be found in Medellin-Gasque et al. (2011). Some example questions at this layer are as follows:

- *CQA-01. Is the action possible?*
- *CQA-02. Are the action preconditions as stated by proponent?*
- *CQA-04. Are the action invariants conditions as stated by proponent?*
- *CQA-07. Are the termination conditions as described possible?*

Layer 2. The timing of an action (*nine questions*).
Here, questions focus on the possibility of the action with respect to a particular time point. This layer includes the following questions:

- *CQAT-02. Is the action possible with the specified duration?*
- *CQAT-06. What is the earliest time the action can start?*
- *CQAT-08. Is it possible for the action to finish at the specified time?*
- *CQAT-09. What is the earliest time the action can end?*

Layer 3. The way actions are combined (*10 questions*).
In some types of plan (e.g. partial-plans), actions can be interleaved, so this layer presents questions that focus the way two actions are combined in the plan, for example:

- *CQAC-01.* (*For sequential actions*) *Could actions α_i and α_j be performed concurrently?*
- *CQAC-02.* (*For sequential actions*) *Can the order of the actions be changed?*
- *CQAC-03.* (*For concurrent actions*) *Is there a conflict in any of the invariant conditions of the actions?*
- *CQAC-06.* (*For concurrent actions*) *Is there a maximum duration for actions to perform concurrently?*

Layer 4. The plan proposal (*11 questions*).
The questions in this layer challenge the plan as a single entity with the elements that support it. Examples include

- *CQPP-01. Is the plan possible?*
- *CQPP-04. Are the current circumstances as stated by proponent?*

- *CQPP-12. Assuming the believed preconditions are true, will the plan bring about the desired state?*
- *CQPP-14. Can the desired goal G be realised?*
- *CQPP-16. Are the values in V_G legitimate values?*

We assume that values are subjective and represent a social interest of the agent, but agents should have a common ontology regarding values, so questions about the legitimacy of values are relevant to align the ontology.

Layer 5. The timing of the plan proposal (*10 questions*).
Here, questions focus on the plan possibility given the times specified. This layer includes the following questions:

- *CQPPT-01. Is the starting point for the plan fixed? If not, what is the range allowed?*
- *CQPPT-05. Can the plan duration be longer?*
- *CQPPT-06. Is the plan possible with the specified duration?*
- *CQPPT-16. Is the plan possible at the specified start time?*

Layer 6. Side effects[5] (*five questions*).
Here, questions in this layer consider plan side effects not previously considered. These include the following questions:

- *CQSE-01. Does performing the plan have a side effect which demotes the value v_n?*
- *CQSE-04. Does performing the plan preclude doing some other action which would promote some other value v_u?*

Layer 7. Alternative options (*four questions*).
Here, questions consider other possibly better alternatives, such as:

- *CQAO-03. Is there an alternative plan that promotes the same value v_n?*
- *CQAO-05. Is there an alternative plan to realise the same goal G?*
- *CQAO-09. Is there another agent Ag_i that could perform action α?*

Whilst we leave open the possibility for further questions to be added to our categories, we have generated the list from a systematic analysis of the various elements of our argumentation scheme and hence believe that it can be taken as complete for our current purposes. We believe a comprehensive dialogue about plans should cover the plan at different levels and enable all aspects of the plan to be questioned therefore the need of 65 questions to critique plans in detail at several levels. In particular implementations, assumptions can, if desired, be made which will place certain aspects beyond question and so reduce the number of questions. By providing the full range of questions our scheme leaves the choice to the implementation in the light of their particular needs. In the next section, we present the details of a dialogue game protocol based on these critical questions.

3. Dialogue game protocol

To define our dialogue game, we use the elements presented in McBurney and Parsons (2009), where the authors describe the elements of a dialogue game:

- Commencement rules: rules which define the circumstances under which the dialogue can start.
- Locution rules: rules that indicate which utterances are allowed.
- Combination rules: rules which define the dialogical context under which particular locutions are permitted or not, or obligatory or not.

- Commitment rules: rules which define the circumstances under which participants incur dialogical commitments by their utterances, and thus alter the contents of the participants' associated commitment stores.
- Combination of commitment rules: rules which define how commitments are combined or manipulated when utterances incurring conflicting or complementary commitments are made.
- Speaker order rules: rules which define the order in which speakers may pose utterances and when the current speaker changes.
- Termination rules: rules that define the circumstances under which the dialogue ends.

We based our dialogue game protocol on these elements focusing on the locutions and the rules for the combination of locutions. Our protocol is also inspired by the protocol presented in Atkinson et al. (2005), where a persuasion dialogue is used to enable agents to argue about proposals for action with a common goal and different preferences. The elements of our protocol are as follows:

Commencement rules: the social context determines which agents can initiate and participate in dialogues, so that some agents have the power to start dialogues by creating a dialogue thread which others can join.

Locution rules: the protocol uses mainly the locutions in the *Fatio* protocol in McBurney and Parsons (2004), where the authors extended the agent communications language FIPA ACL (FIPA 2002) locutions to handle rational argument dialogues. The *Fatio* locutions are: *assert, question, challenge, justify and retract*. We add the locutions *accept, reject,* that allow to accept or reject specific plan proposals or actions. We assume that the language syntax comprises two layers as presented in the FIPA ACL specification. The outer (wrapper) layer comprises the locutions which express the illocutionary force of the inner content (the speech acts) and the inner layer related to the topic of the discussion.

Commitment rules: each proposal and assertion results in a dialogical commitment for the agent.

Combination of commitments rules: specific locutions are available to drop commitments.

Speaker order rules: order and turn taking are determined in the postconditions of each locution.

Termination rules: the dialogue finishes when a proposal is accepted by all the participants, when no more proposals are available for evaluation or when all the participants leave the dialogue.

Our protocol is divided into six stages that group locutions together and help to define the semantics of the protocol. The stages are based on those presented in McBurney, Hitchcock, and Parsons (2007), where dialogue stages for a deliberation dialogue are specified as a part of a formal framework. Hulstijn uses a similar five-stage model for negotiation dialogues in Hulstijn (2000). Our dialogue game stages are (details of our dialogue game protocol are presented in Medellin-Gasque, Atkinson, and Bench-Capon (2012a)) given as follows:

(1) *Opening stage*: this stage is to ensure that all the agents that join the dialogue commit to cooperate towards agreeing on a plan. We assume that the permissions to participate in the dialogue are given by the social context.
(2) *Plan proposal stage*: this stage is where an agent takes the proponent role and puts forward a plan to reach a goal. Agents can adopt one of two roles in the dialogue, the *proponent-agent* or the *respondent-agent*. It is important to mention that although each proposal is evaluated separately, the protocol enables participants to present several proposals until one is generally accepted.

Table 2. Protocol syntax and informal semantics.

Locution	Preconditions	Postconditions
$open\_dialogue(d, ag)$	Agent *ag* has permission to open the dialogue	Dialogue *d* created Dialogue stage *opening*
$enter\_dialogue(d, ag)$	Dialogue stage: *open* Agent *ag* has permission to join the dialogue *d*	Agent *ag* in dialogue *d* Dialogue stage: *proposing*
$propose(d, ag, asp_n)$	Dialogue *d* created Agent *ag* in dialogue *d* Dialogue stage: *proposing*	Proposal for plan asp_n created Agent *ag* committed to the elements in proposal ASP Dialogue stage: *evaluation* Agent *ag* takes proponent role
$question(d, ag, asp_n, q_n)$	Proposal asp_n asserted Dialogue stage: *evaluating* Agent *ag* is in dialogue *d* Dialogue stage: *evaluating*	Element related challenged by question q_n Agent *ag* takes respondent role Dialogue stage: *evaluating*
$assert(d, ag, asp_n, \phi, q_n)$	Proposal asp_n asserted Question q_n asserted Dialogue stage: *evaluating*	Agent *ag* committed to element ϕ The assertion represents an answer for question q_n Dialogue stage: *evaluating*
$challenge(d, ag, asp_n, q_n)$	Proposal asp_n asserted Dialogue stage: *evaluating*	Element challenged by q_n Agent *ag* takes respondent role Dialogue stage: *evaluating*
$justify(d, ag, asp_n, \phi)$	Proposal asp_n asserted Challenge for φ asserted	Agent *ag* committed to ϕ Dialogue stage: *evaluating*
$retract(d, ag, asp_n)$	Proposal asp_n asserted	Agent *ag* not committed to asp_n
$agreel(d, ag, asp_n)$	Proposal (asp_n) asserted by agent *ag*	Proposal asp_n accepted by *ag* Dialogue stage: *selecting*
$reject\_proposal(d, ag, asp_n)$	Proposal asp_n open Dialogue stage: *selecting*	Dialogue stage: *proposing* Proposal asp_n closed
$leave\_dialogue(d, ag)$	Dialogue *d* created Agent *ag* in dialogue *d*	Agent *ag* out of dialogue *d* Agents' *ag* not committed to any element

(3) *Evaluation stage*: in this stage, the respondent agent uses our set of critical questions to pose attacks on a proponent agent's argument in the proposal. The proponent can either assert information to provide evidence or retract the element or proposal.
(4) *Refinement stage*: we use this stage to refine the plan by proposing new actions. This stage differs from the evaluation stage in the sense that the evaluation is performed at the action level. This stage enables deliberation dialogue where agents can engage in a dialogue about how to achieve their goals.
(5) *Selection stage*: the evaluation of accepted proposals is done in this stage. The dialogue has two outcomes, either a consensus for a plan execution is reached or not.
(6) *Closing stage*: at this stage, agents finish their participation in the dialogue following an acceptance or rejection of the proposal.

Table 2 presents examples of the syntax and the informal axiomatic semantics of the protocol. We next define our strategies to select critical questions to be used in the *Evaluation* stage of our protocol.

4. Strategies to select critical questions

In open environments it is, in principle, desirable that agents engaged in a dialogue have the freedom to pose any question, but in some scenarios, agents may be restricted by preconditions imposed by the domain, the social context or the dialogue protocol. While this restricts the freedom of the agents, it typically has benefits in terms of the efficiency and coherence of the dialogue. We focus on the process of selecting from a set of critical questions once the communicative act, in this case *question*(), has been selected but the content remains to be determined. A different problem is to select the communicative act itself and different strategies could be applied as in Amgoud and Hameurlain (2007), where the authors define a strategy for dialogue move selection. In this article, we consider a strategy to be a process with two steps: (1) identify the differences between the respondent's representation of the world and the proposal presented by the proponent to generate potentially useful questions and (2) prioritise the questions and select which one to pose. We now discuss these two steps in more detail.

4.1. *Belief representation alignment*

Our respondent agent identifies questions to present based on the information in the proposal. When finding these questions we are verifying the plan presented against the local specification of the respondent. The process used to identify questions compares the information of the proposal which is: the goal, the initial state, the action specification, the social context and the values involved against the agent's local beliefs about the world. If an inconsistency is detected, the related question is added to a list of potentially useful questions. We distinguish this problem from an ontology alignment problem. Of course, there may be an element of both problems in a scenario such as this but we assume that the ontologies are the same for agents or that any differences have been resolved before the dialogue starts.[6] We use the pseudo-code in Table 3 to obtain questions comparing the information in the proposal with its beliefs.

In a cooperative dialogue scenario such as the one we are considering agents need to agree on their beliefs about the world. In a continuously changing environment, this may be very difficult. Even if agents agree at some point on a set of circumstances, a change could happen that invalidates several coordination agreements between agents. This means the protocol semantics should allow questions about the domain to be posed more than once. The process to create questions identifies and helps to resolve conflicts in the world representation of the agents after which the process can be continued by selecting the best-possible plan based on their preferences. The process of identifying questions takes into account the fact that some questions depend on the outcome of

Table 3. World representation alignment pseudo-code.

	Step	Comment
1.	IdentifyConflicts()	Information taken from the plan proposal
2.	CheckPlanGoal()	Inconsistencies in the plan goal
3.	If inconsistency detected	
4.	AddQuestions()	Creating the sublist with possible attacks
5.	CheckInitialState()	Inconsistencies in the initial state
6.	If inconsistency detected	
7.	AddQuestions()	
8.	For each Action in the plan	
9.	CheckActionSpecification()	Inconsistencies in the actions specification
10.	If inconsistency detected	
11.	AddQuestions()	
12.	Next Action	
13.	CheckSocialContext()	Inconsistencies in the social context
14.	If inconsistency detected	
15.	AddQuestions()	
16.	CheckValues()	Inconsistencies in the values
17.	If inconsistency detected	
18.	AddQuestions()	
19.	End	

others. For example, when checking for the validity of an action element, if the action is not valid, there is no point in considering the question of whether the action is possible.

4.2. *Question prioritising and selection*

Critical questions were classified into seven layers in Section 2. We now present a more detailed analysis of the critical questions to further classify and order them taking into account this finer grained description. When categorising the questions our aim is to identify their intrinsic purpose in the dialogue, which we use to give the questions a priority in our strategies. From a general perspective, in a planning scenario, critical questions may refer to the domain, to the plan, or to the scheduling of the actions. We take this categorisation as our first-ordering criterion. Intuitively, we want first to resolve inconsistencies of beliefs about the domain (to create valid plans), then focus on the plan itself and finally, focus on the scheduling elements of the plan. A standard AI planning process in fact follows the same order: a valid domain and problem representation are the input of a generic planner algorithm and once the plan is created, a scheduling process can be applied to it.

The next categorisation refers to the way a proposal could be questioned depending on the nature of the critical question. From the set of our questions, a question can challenge:

- The *suitability* of an element (*e.g. Are there any side effects when executing action α?*)
- The *validity* of an element (*e.g. Do the termination conditions of action α hold?*)
- The *possibility* of an element (*e.g. Are the current circumstances possible?*)
- The *possibility in time* of an element (*e.g. Is it possible to execute action α at time t?*)
- Other possible better *alternatives* (*e.g. Is there an alternative plan to reach the goal?*)

This categorisation provides an order in which questions can be posed. We first want to establish that the plan proposed is *suitable* for the context based on the motives that it satisfies. We include in the suitability questions, for example, *Do the new circumstances already pertain?* or *Does the goal promote value v?* It may not make sense to argue about the validity of an element if the intrinsic motives to perform the plan are not fully agreed by all parties. Once agreed as to the suitability, we can question the validity of the elements to resolve any conflicts about the beliefs of the world that the agents have (*e.g. Is action α valid?*) Once agents agree on the validity of the elements, the possibility of the plan can be addressed. An element can be valid but not 'possible' for the current state of the world when an agent has the action in its repertoire but the preconditions are not currently satisfied, necessitating another step in the plan to enable the action. For example, the action *takeTrain*() may be recognised by the agents as a *valid* action to perform in the world but may not be *possible* to perform it at a certain train station or at a certain time. When we deal with possibility questions, a finer grain of detail may be addressed leading us to consider the actions that are possible for the plan. With the actions we can follow the same order, first suitability questions, then validity questions and finally possibility questions. Finally, we can focus on alternative plans as a direct attack on the critiqued plan. The plan proposed may be valid and acceptable for all the involved agents but still another plan may be a better option given their preferences. If we combine both categorisations, we obtain a priority order to consider critical questions in a dialogue. We designate this strategy $s1$, and also consider a different priority order placing validity questions before the suitability questions to create strategy $s2$ (Table 4 presents the two strategies). Each one of layers presented in Section 2.1 can have questions of the types presented above. We believe that this categorisation is better to create a priority order for questions in a dialogue than simply progressing through the critical questions layers. In the next section, we present an implementation where agents use these strategies in a dialogue simulation with various scenarios and analyse the results together with a random question selection approach, which provides a base line for comparison.

Table 4. Order in which questions are considered for both strategies.

Strategy $s1$ question order	Strategy $s2$ question order
1. Plan suitability	1. Actions' validity
2. Actions' suitability	2. Plan elements' validity
3. State of the world suitability	3. Plan suitability
4. Plan elements' validity	4. Actions' suitability
5. Actions' validity	5. Norms suitability
6. State of the world validity	6. Plan elements' possibility
7. Actions' possibility	7. Plan elements' possibility in time
8. Actions' possibility in time	8. Actions' possibility
9. Plan elements' possibility	9. Actions' possibility in time
10. Plan elements' possibility in time	10. State of the world validity
11. Alternative actions	11. Alternative actions
12. Alternative plans	12. Alternative plans

5. Implementation

In this section, we describe experiments that show the effect of following our strategies on the effectiveness and efficiency of the dialogues. The strategies guide the process of selecting moves in a dialogue between two agents. In the example problem, two agents (John and Paul), who are in Inverness and need to attend a conference in Paris, have to choose between different possible routes travelling together. The actions that can be combined to reach the goal are: *takeTrain*(), *takeFlight*() and *takeCoach*() through different cities (Inverness, Manchester, London, Paris).[7] Each city has restrictions on the availability of the train, plane and coach connections. Several values are used: value v_1 = *money*, the cheapest option, value v_2 = *time*, the fastest option, value v_3 = *friendship*, travelling with a friend, and value v_4 = *comfort*, the most comfortable way to travel.

The purpose of the implementation is to apply the selection strategies in a scenario where agents engage in a persuasive dialogue to select the best plan. We designed four test cases where agents have different plans and information about the world. Then, we give agents a set of preferences and a strategy and run a dialogue simulation where agents propose plans using our dialogue game to discuss with one another and select the best-possible plan for both. The strategy is applied to the questioning process that rejects or accepts the plan based on the validity of the information the agents present. We use our two strategies and also one in which questions are posed randomly to provide a comparison point. We now describe briefly the dialogue protocol implementation, which is fully set out and described in Medellin-Gasque et al. (2012a).

5.1. *Dialogue protocol implementation*

To implement our protocol, we use *TuCSoN* (Tuple Centres Spread over the Network), a software platform for *tuple-centre* applications (Omicini and Zambonelli 1999). Tuple-based coordination models originate from the field of paralleling programming, but their features are also useful for the coordination of distributed systems (Gelernter and Carriero 1992). A *tuple-centre* is basically an enhancement of a *tuple space*, where agents synchronise and cooperate over information available in a shared data space through a behaviour specification. A *tuple-centre* is thus a *tuple-space* enhanced with a behaviour specification that defines the responses or reactions to communication events (Omicini and Denti 2001). These responses are specified in terms of a reaction specification language: *TuCSoN* uses *ReSpecT* (Reaction Specification Tuples) (Denti, Natali, and Omicini 1998), which adopts a tuple language based on first-order logic to define logic tuples that are accessible via standard communication operations. The protocol syntax and semantic rules are preloaded in the *tuple-centre* as *ReSpecT* rules. The *TuCSoN* infrastructure permits *tuple-centres* to be saved making them persistent. We use this feature to load the protocol each time a dialogue is started. Valid locutions are in the form of persistent tuples and protocol semantics are specified using *ReSpecT* reactions. Table 5 presents the *ReSpecT* tuples used to define the syntax of our protocol. The parameters used in the locutions are (*Di*) dialogue identifier, (*Ag*) agent identifier, (*Pr*) proposal identifier, (*Ac*) action identifier, (*CqN*) critical question and (*Ev*) evidence.

The semantics of the protocol are embedded in the tuple-centre as *ReSpecT* reactions. As an example, the semantics for the *propose* locution as a *ReSpecT* reaction are given in Table 6.

5.2. *Agent and system architecture*

An AI planning task requires a description of the initial state, a set of action capabilities and a set of private goals. We base our world representation in the PDDL originally introduced in Ghallab et al. (1998) and revised in Fox and Long (2003) (PDDL 2.1) to handle durative actions. Typically, a PDDL specification consists of a set of predicates, a set of actions with parameters, preconditions and effects. PDDL 2.1 introduced the concept of durative actions as explained in Section 2 and permits the duration together with more conditions and effects to be included in the action. A condition could be labelled as *at_start*, *over_all* and *at_end*, and effects can be specified as *at_start* and *at_end*. We use Java classes to represent the action elements and the state of the world. Each of our agents has a 'Dialogue Manager' in charge of the communication with the tuple-centre and a configurable strategy to select critical questions. Figure 3 gives an overview of the agent components and the system architecture.

Table 5. Protocol syntax and *ReSpecT* format.

Locution	*ReSpecT* format	Description
$open\_dialogue()$	$loc(open(Di, Ag))$	Agent opening a dialogue
$enter\_dialogue()$	$loc(enter(Di, Ag))$	Agent entering the dialogue
$propose\_plan()$	$loc(propose(Di, Ag, Pr))$	Agent proposing a plan in proposal *Pr*
$accept\_proposal()$	$loc(accept\_proposal(Di, Ag, Pr))$	Agent accepting the proposal
$retract\_proposal()$	$loc(retract\_proposal(Di, Ag, Pr))$	Agent retracting the proposal
$propose\_action()$	$loc(propose\_action(Di, Ag, Pr, Ac))$	Agent proposing action for a plan
$accept\_action()$	$loc(accept\_action(Di, Ag, Pr, Ac))$	Agent accepting an action
$retract\_action()$	$loc(retract\_action(Di, Ag, Pr, Ac))$	Agent retracting an action
$question()$	$loc(question(Di, Ag, Pr, CqN))$	Agent questioning an element in *Pr*
$assert()$	$loc(assert(Di, Ag, Pr, CqN, Ev))$	Agent asserting evidence to question *CqN*
$leave\_dialogue()$	$loc(leave\_dialogue(Di, Ag))$	Agent leaving the dialogue

Table 6. *ReSpecT* semantics for the *propose* locution.

ReSpecT reaction	Description
$reaction(out(propose(Di, Ag, Pr)),$	Reaction tuple for the proposed plan
Preconditions	
$(rd\_r(loc(propose\_plan(\_,\_,\_))),$	Check the locution is in the protocol syntax
$rd\_r(dState(proposing)),$	Check the dialogue state
$rd\_r(participant(Di, Ag)),$	Check participant *Ag* is in the dialogue
Postconditions	
$out\_r(dhistory(propose\_plan(Di, Ag, Pr))),$	Insert dialogue history tuple
$in\_r(dState(open)),$	Delete previous dialogue state
$out\_r(dState(evaluating)),$	Insert new dialogue state
$out\_r(committment(Ag, Pr)),$	Insert dialectical commitment to proposal *Pr*
$out\_r(role(Di, rolePr, Ag)),$	Assign role *proponent* to the agent
$in\_r(propose\_plan(\_,\_,\_)))).$	Clean auxiliary tuples

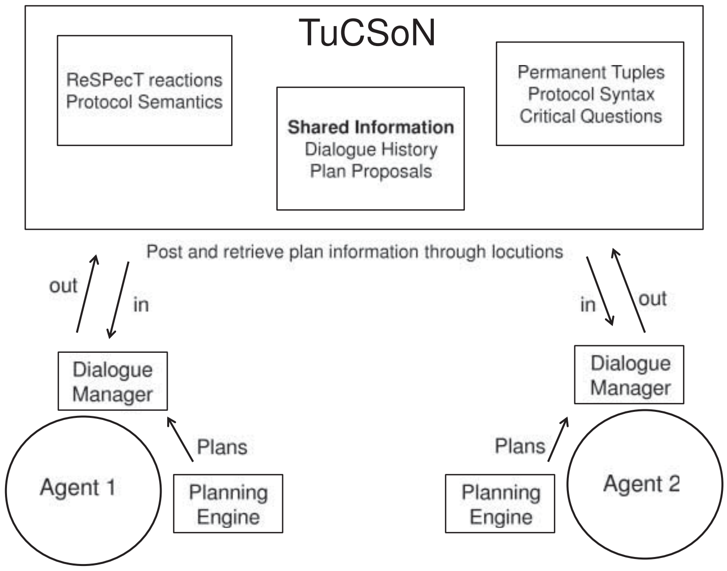

Figure 3. System architecture.

We use the 'Dialogue Manager' concept from the TRAINS implementation presented in Allen et al. (1995), where a conversational planning agent engages in a dialogue to create a plan, using feedback received from the interaction. The Dialogue Manager in our system has the following main tasks:

(1) Identify questions to pose (validating local information).
(2) Apply the critical question selection strategy.
(3) Create the proposal tuples.
(4) Provide the interface to communicate with the tuple-centre to post and retrieve tuples; tuple centre tasks comprise:
   - Validate the protocol syntax.
   - Validate the protocol semantics.
   - Retrieve critical questions.
   - Retrieve dialogue participants.
   - Retrieve the dialogue history.

5.3. *Dialogue runs*

A dialogue-run in our experiments consists of the following steps:

(1) A proponent agent (PRO) initiates the dialogue.
(2) The agent's planning engine selects the preferred plan according to its value preference and current beliefs.
(3) The dialogue manager transforms the plan into a proposal object.
(4) The dialogue manager creates a valid tuple using the proposal object.
(5) The protocol in the tuple centre validates the locution.
(6) The respondent agent (RES) acknowledges the proposal and starts the questioning process, applying the strategy.

(7) The questioning process involves RES questioning PRO over the elements in the proposal until acceptance, retraction or rejection.
(8) When the RES agent poses the question: *CQAO-03. Is there an alternative plan to realise the same goal*? the agent roles change. RES gets a turn to propose and evaluate its preferred plan (note that we just give RES one chance to put forward its preferred plan[8]).
(9) Once the questioning process finishes, if the preferred plan for both is the same the dialogue finishes. Where we have two valid plans at the end of the evaluation PRO selects the plan which promotes more values, or when these are equal, the plan that demotes fewer values. Further details on how the dialogue runs are implemented are given in the next section when we analyse the results.

Since we use a Java implementation, the communication between modules is done through objects. The pseudo-code in Figure 4 shows how the agents create and interchange objects for a dialogue run from the point when the proposal needs to be posted.

```
Proponent Agent
 ProposePlan()
      \\Agent selects the preferred plan
     PlanningEngine.IdentifyBestPreferredPlan()
      \\Create proposal from object plan
     DialogueManager.CreateProposal()
      \\Post proposal to the tuple centre
     Dialogue Manager.PostTuple()
     TucsonContext.PostTuple()
       \\Proponent waits for questions
     DialogueThread.WaitforQuestions()
     If PlanQuestioned()
        For each question
            \\Search for evidence related to question
            DialogueManager.SearchEvidence()
          If EvidenceFound()
                DialogueManager.ProvideEvidence()
          else
                DialogueManager.RetractProposal()
          End if
      Next
   End If
End

Respondent Agent
DialogueThread.AnswerProposal()
   DialogueManager.ApplyStrategy()
     \\Identify valid questions
      DialogueManager.IdentifyConflicts()
    \\Order questions based on strategy
      DialogueManager.Orderquestions()
      While QuestionList.haselements()
     \\Post question to the tuple centre
          DialogueManagerPostQuestion()
          \\Wait from proponent answer
           DialogueThread.WaitforAnswer()
         If EvidenceNotProvided()
            \\Question defeats proposal
                  break()
               End If
      End While
  End
End
```

Figure 4. Pseudo-code for the proponent and respondent agents' functions in a dialogue simulation.

5.4. *Scenarios used in the experiments*

The plans used by our agents in our experiments are presented in Table 7 together with the status of their values (promoted(+), demoted(−) or neutral(=)). Although each individual action could be associated with a value, for the sake of simplicity here we will only consider values related to the plan as a whole.

We use 2 agents and 20 test cases presented in Tables 8 and 9 to generate dialogue runs. Test cases are formed by providing the agents with:

(1) Information about the world:
- A set of constraints that represent the *social context.*
- A belief about the initial state.
- A set of action specifications.

(2) A set of plans.
(3) A set of values.
(4) A preference order over values.

In the different test cases, we change the validity of some elements in the plans and/or world representation for each agent to create different runs. We give agents four different sets of information about the world and plans (presented in Table 8) and we combine them with five different preference orders (Table 9) to generate the 20 test cases. In Table 8, a check mark (✓) indicates the validity of the element and a cross (×) indicates some problem in the specification. The validity of elements (actions conditions, action effects) is represented using a 'token attribute' associated with each element. That the 'token' is *false* represents that the element validity against the context has expired.

In test case A, both agents have valid plans and their beliefs about the world are aligned. This test case generates only questions about alternative plans. In test case B, John's plans p_1, p_2, p_3 are

Table 7. Agents' plans.

Plan	Actions	Values	
p_1 – Coach	$j_1 = takeCoach(Inverness, Manchester)$	$v_1 = money$	$+$
	$j_2 = takeCoach(Manchester, London)$	$v_2 = duration$	$-$
	$j_3 = takeCoach(London, Paris)$	$v_3 = friendship$	$=$
		$v_4 = comfort$	$-$
p_2– Trains	$j_4 = takeTrain(Inverness, Manchester)$	$v_1 = money$	$=$
	$j_5 = takeTrain(Manchester, Paris)$	$v_2 = duration$	$=$
		$v_3 = friendship$	$+$
		$v_4 = comfort$	$+$
p_3 – Flight	$j_6 = takeFlight(Inverness, Paris)$	$v_1 = money$	$-$
		$v_2 = duration$	$+$
		$v_3 = friendship$	$-$
		$v_4 = comfort$	$=$
p_4 – Coach–train	$j_1 = takeCoach(Inverness, Manchester)$	$v_1 = money$	$=$
	$j_5 = takeTrain(Manchester, Paris)$	$v_2 = duration$	$-$
		$v_3 = friendship$	$+$
		$v_4 = comfort$	$=$
p_5 – Train–flight	$j_7 = takeTrain(Inverness, London)$	$v_1 = money$	$-$
	$j_8 = takeFlight(London, Paris)$	$v_2 = duration$	$=$
		$v_3 = friendship$	$=$
		$v_4 = comfort$	$=$
p_6 – Coach–train–flight	$j_1 = takeCoach(Inverness, Manchester)$	$v_1 = money$	$-$
	$j_9 = takeTrain(Manchester, London)$	$v_2 = duration$	$=$
	$j_{10} = takeTrain(London, Paris)$	$v_3 = friendship$	$=$
		$v_4 = comfort$	$-$

Table 8. Test cases.

Test case	John's plans		John's beliefs		Paul's plans		Paul's beliefs	
A	p_1	✓	Social context constraints	✓	p_4	✓	Norms	✓
	p_2	✓	Initial state	✓	p_5	✓	Initial state	✓
	p_3	✓	Action specification	✓			Action specification	✓
	p_6	✓						
B	p_1	×	Social context constraints	✓	p_4	✓	Norms	×
	p_2	×	Initial state	✓	p_5	✓	Initial state	×
	p_3	×	Action specification	✓			Action specification	✓
	p_6	✓						
C	p_1	✓	Social context constraints	✓	p_5	✓	Norms	✓
	p_2	×	Initial state	✓	p_4	×	Initial state	✓
	p_3	×	Action specification	✓			Action specification	✓
	p_6	✓						
D	p_1	✓	Social context constraints	✓	p_4	✓	Norms	✓
	p_2	×	Initial state	×	p_5	✓	Initial state	✓
	p_3	✓	Action specification	✓			Action specification	✓
	p_6	✓						

Table 9. Test case agents' preferences.

	John's preference orders		Paul's preference orders	
Run	Values	Plan	Values	Plan
1	$v_1 > v_4 > v_3 > v_2$	v_1 in p_1	$v_3 > v_4 > v_1 > v_2$	v_3 in p_4
2	$v_3 > v_2 > v_1 > v_4$	v_3 in p_2	$v_3 > v_4 > v_1 > v_2$	v_3 in p_4
3	$v_2 > v_1 > v_3 > v_4$	v_2 in p_1	$v_2 > v_4 > v_1 > v_3$	v_2 in p_5
4	$v_3 > v_1 > v_4 > v_2$	v_3 in p_2	$v_1 > v_2 > v_4 > v_3$	v_1 in p_4
5	$v_1 > v_4 > v_3 > v_2$	v_1 in p_1	$v_2 > v_4 > v_3 > v_1$	v_3 in p_4

not valid and induce questions about the validity and possibility of action elements (any preference over these plans has to be re-evaluated after the dialogue). Some constraints and initial state of the world believed by Paul are not valid, and so this situation generates questions regarding the validity and possibility of the elements. In test case C, John's plans p_2, p_3 are not valid and Paul's plan p_4 is also not valid. In test case D, John's initial state is not valid nor is his plan p_2. Dialogue runs for this test case aim to question the proposals at the suitability level.

Table 9 presents the five sets of preference orders we use for the two agents. Agents may change their preferred plan once the dialogue finishes, depending on the outcome of the questioning process. More details on how the agents' preferred plan changes after the dialogue is given in the results presented in the next section. To exercise the different combinations of the strategy used, we ran each test case six times combining the strategy used (strategies $s1$, $s2$ and a random approach) with the agent that starts the dialogue (two agents), to determine whether which agent goes first has a significant influence on the dialogue. In this approach, critical questions cannot be on their turn attacked (as is the case of attacks using arguments in a dialogue game) and should be answered.

From the complete list of 65 critical questions, we have implemented 28 questions for these experiments.[9] We implemented critical questions related to validity, possibility and suitability for the action and plan specifications, alternate plans and side effect questions. We intend to investigate the characteristics as part of future work. An annotated example of a dialogue run outcome is given in Appendix 1.

6. Evaluation

To analyse the results, we record for each run the number of proposals, the number of questions and the outcome of the dialogue. In Tables 10–17, we present the results of the dialogue runs. We discuss the results presented in each table and conclude with an overall analysis of the results. The tables present the results for each test case separately, each run (characterised by the agents' preference over plans) is executed three times, one for each of our strategies and one random approach. Results for each test case are presented in two tables, depending on the agent that starts the dialogue. The results present the number of proposals and the order in which they were evaluated, together with the outcome of the plan evaluation (a check mark (✓) for an accepted proposal, and a cross ($\times$) for a rejected proposal). Finally, we present the overall number of questions evaluated (Qs) and the selected plan. We analyse now each test case separately.

For test case A (Tables 10 and 11), we can observe the following:

- Neither plans nor agents' beliefs have inconsistencies and in A1–A4 both strategies just pose one question: *CQAO-05. Is there an alternative better plan to reach the goal*? In the random approach, the respondent agent has to go through all the questions, because the random approach has no process to identify relevant questions.
- For this test case, where no inconsistencies were found, there is no difference in the results when we change the agent that starts the dialogue.
- In run A1, after plans p_1 and p_4 are evaluated, the proponent selects plan p_4, which demotes fewer values than plan p_1 following step (9) from Section 5.3. This selection is used in runs A1–A4 in Tables 10 and 11.
- In run A5 from Table 10, plan p_1 is discarded because of its side effects identified by question: *CQSE-01. Does the plan p have a side effect which demotes the value v_n*? Plan p_1 demotes value $v_2 = duration$, which is Paul's highest ranked value. Similarly, in run A5 from Table 11, plan p_5 demotes value $v_1 = money$, John's highest ranked value.

From test case B results (Tables 12 and 13), we can observe the following:

- When John: starts the dialogue, plan proposal p_6 is always selected since it is the only valid plan.
- In run B1 from Table 12, agents evaluate five proposals. It is worth mentioning that even though plans p_4 and p_5 are valid plans, Paul: has incorrect beliefs about the initial state and that is why the proposals are rejected.

Table 10. Test case A when John starts the dialogue.

Run	Preference before dialogue	After dialogue	Strategy	Proposals	Qs	Plan selected
A1	John: p_1	John: p_1	$s1$	$2 - p_1(\checkmark), p_4(\checkmark)$	1	p_4
	Paul: p_4	Paul: p_4	$s2$	$2 - p_1(\checkmark), p_4(\checkmark)$	1	p_4
			Random	$2 - p_1(\checkmark), p_4(\checkmark)$	119	p_4
A2	John: p_2	John: p_2	$s1$	$2 - p_2(\checkmark), p_4(\checkmark)$	1	p_2
	Paul: p_4	Paul: p_4	$s2$	$2 - p_2(\checkmark), p_4(\checkmark)$	1	p_2
			Random	$2 - p_1(\checkmark), p_4(\checkmark)$	169	p_2
A3	John: p_3	John: p_3	$s1$	$2 - p_3(\checkmark), p_5(\checkmark)$	1	p_3
	Paul: p_5	Paul: p_5	$s2$	$2 - p_3(\checkmark), p_5(\checkmark)$	1	p_3
			Random	$2 - p_3(\checkmark), p_5(\checkmark)$	91	p_3
A4	John: p_2	John: p_2	$s1$	$2 - p_2(\checkmark), p_4(\checkmark)$	1	p_2
	Paul: p_4	Paul: p_4	$s2$	$2 - p_2(\checkmark), p_4(\checkmark)$	1	p_2
			Random	$2 - p_2(\checkmark), p_4(\checkmark)$	169	p_2
A5	John: p_1	John: p_2	$s1$	$3 - p_1(\times), p_2(\checkmark), p_5(\times)$	3	p_2
	Paul: p_5	Paul: –	$s2$	$3 - p_1(\times), p_2(\checkmark), p_5(\times)$	3	p_2
			Random	$3 - p_1(\times), p_2(\checkmark), p_5(\times)$	148	p_2

Table 11. Test case A when Paul starts the dialogue.

Run	Preference before dialogue	After dialogue	Strategy	Proposals	Qs	Plan selected
A1	John: p_1	John: p_1	$s1$	$2 - p_4(\checkmark), p_1(\checkmark)$	1	p_4
	Paul: p_4	Paul: p_4	$s2$	$2 - p_4(\checkmark), p_1(\checkmark)$	1	p_4
			Random	$2 - p_4(\checkmark), p_1(\checkmark)$	119	p_4
A2	John: p_2	John: p_2	$s1$	$2 - p_4(\checkmark), p_2(\checkmark)$	1	p_2
	Paul: p_4	Paul: p_4	$s2$	$2 - p_4(\checkmark), p_2(\checkmark)$	1	p_2
			Random	$2 - p_4(\checkmark), p_2(\checkmark)$	169	p_2
A3	John: p_3	John: p_3	$s1$	$2 - p_5(\checkmark), p_3(\checkmark)$	1	p_3
	Paul: p_5	Paul: p_5	$s2$	$2 - p_5(\checkmark), p_3(\checkmark)$	1	p_3
			Random	$2 - p_5(\checkmark), p_3(\checkmark)$	91	p_3
A4	John: p_2	John: p_2	$s1$	$2 - p_4(\checkmark), p_2(\checkmark)$	1	p_2
	Paul: p_4	Paul: p_4	$s2$	$2 - p_4(\checkmark), p_2(\checkmark)$	1	p_2
			Random	$2 - p_4(\checkmark), p_2(\checkmark)$	169	p_2
A5	John: p_1	John: –	$s1$	$3 - p_5(\times), p_4(\checkmark), p_1(\times)$	3	p_4
	Paul: p_5	Paul: p_4	$s2$	$3 - p_5(\times), p_4(\checkmark), p_1(\times)$	3	p_4
			Random	$3 - p_5(\times), p_4(\checkmark), p_1(\times)$	148	p_4

- In run B2, agents evaluate three proposals. First, plan proposal p_2 is rejected by Paul. The next value preferred for John: is v_2, so John: picks p_6 which is neutral to v_2 (plan p_1 is also neutral to $v_2 = \textit{duration}$). Finally, Paul proposes plan p_4 through the alternate question option; the proposal is rejected and the proposal process stops.
- For test case B, strategy $s2$ performs better than strategy $s1$ in terms of the number of questions evaluated because most of the inconsistencies were induced by the possibility for action conditions in the plan representations and strategy $s2$ considers these aspects first. As stated above, Strategy $s1$ puts forward suitability questions first, followed by validity and possibility questions.
- When Paul starts the dialogue (Table 13), there is no outcome since none of the plans presented is valid according to Paul's beliefs. Although plan p_4 and plan p_5 are valid plans, Paul has incorrect beliefs about the initial state and this is the reason why the proposals are rejected. Because Paul starts, John: does not get the chance to present plan p_6 because the plan is not preferred in any run. The dialogue implementation is thus not fair in that there can be a difference in the outcome depending on who starts the dialogue. We give the respondent agent just one opportunity to pose its best-preferred plan, and if that plan turns out to be not valid, there is no second chance, whereas the proponent agent has the opportunity to propose all its plans until one is accepted. The protocol can handle multiple proposals for different agents, but the way the dialogue examples were designed does not have the flexibility required to allow a fair dialogue independent of the agent that starts the dialogue. The problem could be addressed by re-starting the dialogue with the other agent going first and having a procedure for adjudicating cases where the outcomes differ.
- In all the runs for this test case, agents do not have a preference after the dialogue since the plan selected does not promote their preferred option. Nevertheless, the best plan according to the next value in the agent's preference order is selected. In this test case, this is not evident since only one plan is valid in the final evaluation.

From test case C results (Tables 14 and 15), we can observe the following:

- Again strategy $s2$ performs better than strategy $s1$ in terms of number of questions evaluated.
- When Paul starts the dialogue in run C1 Table 15, the preferences after the dialogue are: John plan p_1 and Paul plan p_5, John: selects p_1 because between these two plans, p_1 is the plan that promotes more values.

Table 12. Test case B when John starts the dialogue.

Run	Preference before dialogue	After dialogue	Strategy	Proposals	Qs	Plan selected
B1	John: p_1	John: –	$s1$	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_4(\times)$	23	p_6
	Paul: p_4	Paul: –	$s2$	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_4(\times)$	12	p_6
			Random	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_4(\times)$	184	p_6
B2	John: p_2	John: –	$s1$	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	15	p_6
	Paul: p_4	Paul: –	$s2$	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	9	p_6
			Random	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	150	p_6
B3	John: p_3	John: –	$s1$	$3 - p_3(\times), p_6(\checkmark), p_5(\times)$	13	p_6
	Paul: p_5	Paul: –	$s2$	$3 - p_3(\times), p_6(\checkmark), p_5(\times)$	8	p_6
			Random	$3 - p_3(\times), p_6(\checkmark), p_5(\times)$	168	p_6
B4	John: p_2	John: –	$s1$	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	15	p_6
	Paul: p_4	Paul: –	$s2$	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	9	p_6
			Random	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	150	p_6
B5	John: p_1	John: –	$s1$	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_5(\times)$	15	p_6
	Paul: p_5	Paul: –	$s2$	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_5(\times)$	23	p_6
			Random	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_5(\times)$	139	p_6

Table 13. Test case B when Paul starts the dialogue.

Run	Preference before dialogue	After dialogue	Strategy	Proposals	Qs	Plan selected
B1	John: p_1	John: –	$s1$	$3 - p_4(\times), p_5(\times), p_1(\times)$	10	–
	Paul: p_4	Paul: –	$s2$	$3 - p_4(\times), p_5(\times), p_2(\times)$	4	–
			Random	$3 - p_4(\times), p_5(\times), p_2(\times)$	157	–
B2	John: p_2	John: –	$s1$	$3 - p_4(\times), p_5(\times), p_2(\times)$	14	–
	Paul: p_4	Paul: –	$s2$	$3 - p_4(\times), p_5(\times), p_2(\times)$	6	–
			Random	$3 - p_4(\times), p_5(\times), p_2(\times)$	171	–
B3	John: p_3	John: –	$s1$	$3 - p_5(\times), p_4(\times), p_3(\times)$	12	–
	Paul: p_5	Paul: –	$s2$	$3 - p_5(\times), p_4(\times), p_3(\times)$	5	–
			Random	$3 - p_5(\times), p_4(\times), p_3(\times)$	167	–
B4	John: p_2	John: –	$s1$	$3 - p_4(\times), p_5(\times), p_2(\times)$	14	–
	Paul: p_4	Paul: –	$s2$	$3 - p_4(\times), p_5(\times), p_2(\times)$	6	–
			Random	$3 - p_4(\times), p_5(\times), p_2(\times)$	171	–
B5	John: p_1	John: –	$s1$	$3 - p_4(\times), p_5(\times), p_2(\times)$	6	–
	Paul: p_5	Paul: –	$s2$	$3 - p_4(\times), p_5(\times), p_2(\times)$	10	–
			Random	$3 - p_4(\times), p_5(\times), p_2(\times)$	137	–

- In run C5 Table 15, plan p_5 and plan p_1 are rejected because of their side effects as explained for test case A5. Plan p_5 proposed by Paul demotes $v_1 = money$, which is John's highest ranked value in run C5.

From test case D results (Tables 16 and 17), we can observe that Strategy $s1$ performs better in this test case because most of the inconsistencies are in the initial state. Since suitability questions are considered first (e.g *CQPP-05. Do the new circumstances already pertain?*), the dialogue is completed more quickly.

6.1. *Summary of experiments*

We have implemented agents that engage in a dialogue to select the best valid plan possible taking the preferences of both agents into account. The dialogue takes a persuasion approach and makes use of critical questions to evaluate the plan proposal at several levels. In general, the outcome of the dialogue does not change when we change the strategy but the number of questions is always different. The implementation of our two-step strategy shows that the number

Table 14. Test case C when John starts the dialogue.

Run	Preference before dialogue	After dialogue	Strategy	Proposals	Qs	Plan selected
C1	John: p_1	John: p_1	$s1$	$2 - p_1(\checkmark), p_4(\times)$	6	p_1
	Paul: p_4	Paul: –	2	$2 - p_1(\checkmark), p_4(\times)$	4	p_1
			Random	$2 - p_1(\checkmark), p_4(\times)$	78	p_1
C2	John: p_2	John: –	$s1$	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	8	p_6
	Paul: p_4	Paul: –	$s2$	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	5	p_6
			Random	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	106	p_6
C3	John: p_3	John: –	$s1$	$3 - p_3(\times), p_4(\times), p_5(\checkmark)$	7	p_5
	Paul: p_5	Paul: p_5	$s2$	$3 - p_3(\times), p_4(\times), p_5(\checkmark)$	5	p_5
			Random	$3 - p_3(\times), p_4(\times), p_5(\checkmark)$	169	p_5
C4	John: p_2	John: –	$s1$	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	8	p_6
	Paul: p_4	Paul: –	$s2$	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	5	p_6
			Random	$3 - p_2(\times), p_6(\checkmark), p_4(\times)$	106	p_6
C5	John: p_1	John: –	$s1$	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_5(\times)$	10	p_6
	Paul: p_5	Paul: –	$s2$	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_5(\times)$	12	p_6
			Random	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_5(\times)$	149	p_6

Table 15. Test case C when Paul starts the dialogue.

Run	Preference before dialogue	After dialogue	Strategy	Proposals	Qs	Plan selected
C1	John: p_1	John: p_1	$s1$	$3 - p_4(\times), p_5(\checkmark), p_1(\checkmark)$	7	p_1
	Paul: p_4	Paul: –	$s2$	$3 - p_4(\times), p_5(\checkmark), p_1(\checkmark)$	5	p_1
			Random	$3 - p_4(\times), p_5(\checkmark), p_1(\checkmark)$	138	p_1
C2	John: p_2	John: –	$s1$	$3 - p_4(\times), p_5(\checkmark), p_2(\times)$	8	p_5
	Paul: p_4	Paul: –	$s2$	$3 - p_4(\times), p_5(\checkmark), p_2(\times)$	5	p_5
			Random	$3 - p_4(\times), p_5(\checkmark), p_2(\times)$	129	p_5
C3	John: p_3	John:–	$s1$	$2 - p_5(\checkmark), p_3(\times)$	6	p_5
	Paul: p_5	Paul: p_5	$s2$	$2 - p_5(\checkmark), p_3(\times)$	4	p_5
			Random	$2 - p_5(\checkmark), p_3(\times)$	84	p_5
C4	John: p_2	John: –	$s1$	$3 - p_4(\times), p_5(\checkmark), p_2(\times)$	8	p_5
	Paul: p_4	Paul: –	$s2$	$3 - p_4(\times), p_5(\checkmark), p_2(\times)$	5	p_5
			Random	$3 - p_4(\times), p_5(\checkmark), p_2(\times)$	126	p_5
C5	John: p_1	John: –	$s1$	$3 - p_5(\times), p_4(\times), p_1(\times)$	7	–
	Paul: p_5	Paul: –	$s2$	$3 - p_5(\times), p_4(\times), p_1(\times)$	9	–
			Random	$3 - p_5(\times), p_4(\times), p_1(\times)$	105	–

Table 16. Test case D when John starts the dialogue.

Run	Preference before dialogue	After dialogue	Strategy	Proposals	Qs	Plan selected
D1	John: p_1	John: p_4	$s1$	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\times), p_4(\checkmark)$	13	p_4
	Paul: p_4	Paul: p_4	2	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\times), p_4(\checkmark)$	15	p_4
			Random	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\times), p_4(\checkmark)$	205	p_4
D2	John: p_2	John: p_4	$s1$	$5 - p_2(\times), p_6(\times), p_1(\times), p_3(\times), p_4(\checkmark)$	13	p_4
	Paul: p_4	Paul: p_4	2	$5 - p_2(\times), p_6(\times), p_1(\times), p_3(\times), p_4(\checkmark)$	15	p_4
			Random	$5 - p_2(\times), p_6(\times), p_1(\times), p_3(\times), p_4(\checkmark)$	182	p_4
D3	John: p_3	John: p_5	$s1$	$5 - p_3(\times), p_6(\times), p_2(\times), p_1(\times), p_5(\checkmark)$	13	p_5
	Paul: p_5	Paul: p_5	2	$5 - p_3(\times), p_6(\times), p_2(\times), p_1(\times), p_5(\checkmark)$	15	p_5
			Random	$5 - p_3(\times), p_6(\times), p_2(\times), p_1(\times), p_5(\checkmark)$	198	p_5
D4	John: p_2	John: p_4	$s1$	$5 - p_2(\times), p_6(\times), p_1(\times), p_3(\times), p_4(\checkmark)$	13	p_4
	Paul: p_4	Paul: p_4	2	$5 - p_2(\times), p_6(\times), p_1(\times), p_3(\times), p_4(\checkmark)$	15	p_4
			Random	$5 - p_2(\times), p_6(\times), p_1(\times), p_3(\times), p_4(\checkmark)$	182	p_4
D5	John: p_1	John: –	$s1$	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_5(\times)$	8	–
	Paul: p_5	Paul: –	2	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_5(\times)$	12	–
			Random	$5 - p_1(\times), p_2(\times), p_3(\times), p_6(\checkmark), p_5(\times)$	145	–

Table 17. Test case D when Paul starts the dialogue.

Run	Preference before dialogue	After dialogue	Strategy	Proposals	Qs	Plan selected
D1	John: p_1	John: p_4	$s1$	$2 - p_4(\checkmark), p_1(\times)$	10	p_4
	Paul: p_4	Paul: p_4	$s2$	$2 - p_4(\checkmark), p_1(\times)$	12	p_4
			Random	$2 - p_4(\checkmark), p_1(\times)$	120	p_4
D2	John: p_2	John: p_4	$s1$	$2 - p_4(\checkmark), p_2(\times)$	10	p_4
	Paul: p_4	Paul: p_4	$s2$	$2 - p_4(\checkmark), p_2(\times)$	12	p_4
			Random	$2 - p_4(\checkmark), p_2(\times)$	74	p_4
D3	John: p_3	John: p_5	$s1$	$2 - p_5(\checkmark), p_3(\times)$	10	p_5
	Paul: p_5	Paul: p_5	$s2$	$2 - p_5(\checkmark), p_3(\times)$	12	p_5
			Random	$2 - p_5(\checkmark), p_3(\times)$	81	p_5
D4	John: p_2	John: p_5	$s1$	$2 - p_5(\checkmark), p_3(\times)$	10	p_5
	Paul: p_4	Paul: p_5	$s2$	$2 - p_5(\checkmark), p_3(\times)$	12	p_5
			Random	$2 - p_5(\checkmark), p_3(\times)$	182	p_5
D5	John: p_1	John: p_4	$s1$	$3 - p_5(\times), p_4(\checkmark), p_1(\times)$	13	p_4
	Paul: p_5	Paul: p_4	$s2$	$3 - p_5(\times), p_4(\checkmark), p_1(\times)$	17	p_4
			Random	$3 - p_5(\times), p_4(\checkmark), p_1(\times)$	146	p_4

of questions decreases considerably when compared with the random approach in all of the runs. This is the most important feature that we wanted to show when running these experiments. Asking questions about the main issues will naturally help converge to a solution much faster than asking random questions where there is no previous question identifying and the priority of questions is constructed randomly.

When posing random questions, in the worst case the respondent agent has to go through all the questions, which is not desirable. Now, when using a strategy, the dialogue length changes depending on the type of conflict the agents have. If it is possible to anticipate which sort of problems are likely in a particular setting, the appropriate strategy can be chosen accordingly.

When the agents' preferences change, the number of questions does not change considerably but sometimes the quality of the outcome may be affected. This is because the best plan might not be considered if one acceptable to both agents is considered before the best plan is reached. When an agent prefers a plan, it tries to put it forward first and so accelerates the process determining whether it is accepted or rejected. We believe that the order of the questions in the strategy could be further tailored for a particular scenario with information of previous dialogues, which would provide information about the other agents' preference as in Black and Atkinson (2011).

7. Related research

This article contributes to an active area of research that uses argumentation for practical reasoning and provides autonomous agents with a way to communicate and cooperate when selecting and executing a plan in a non-deterministic environment.

The practical reasoning scheme of Atkinson et al. (2006) together with the AATS semantics of Atkinson and Bench-Capon (2007) forms the foundation of our scheme but has been extended especially with respect to time, duration and sequencing of actions. Thus plans, rather than single actions, can be considered. Although Atkinson and Bench-Capon (2007) reason about plans as single monolithic super-actions, our extension allows us to get inside the plans and consider their particular components. Our current account also examines the temporal aspects more thoroughly.

In Dunin-Keplicz and Verbrugge (2003), the authors define *planning steps* in relation to dialogue. The first step is the discussion of proposals as a subtype of persuasion dialogue. The outcome of this dialogue should be a set of sub-goals and a common belief about this goals. The next step required is an *inquiry* or *deliberation* dialogue that matches actions with subtasks. *Persuasion* and *information seeking* dialogues are then applicable during the allocation of these tasks. Even negotiation dialogues may be required because agents may have a conflict of interests during action allocation. Finally, a collectively trusted team member may conclude action allocation.

In Tang, Norman, and Parsons (2009), a model for individual and joint actions of agents for describing the behaviour of multi-agent teams is presented. The model uses policies to generate plans and at the same time, the communication needs for the execution stage are embedded in the policy algorithm. Thus, the communication needs are considered before the plan is executed. In our approach, agents propose plans taken from a plan library and then engage in a dialogue to justify the plans and possibly refine them. Agents do not create plans in our approach; we focus instead on the mutual acceptability of plans. We consider a refinement of pre-formulated plans and assume that agents have a plan library. Although agents creating the plans would not change the perspective of the article. Tang et al.'s approach could be combined with ours to generate a comprehensive planning process that includes the creation of the plan, its justification and eventually its execution.

In Belesiotis, Rovatsos, and Rahwan (2010), the authors develop an argumentation mechanism for reconciling conflicts between agents over plan proposals. The authors extend a protocol where argument-moves enable discussion about planning steps in iterated dispute dialogues as presented in Dunne and Bench-Capon (2003). The approach identifies relevant conflicts in agents' beliefs and discusses algorithms for argument generation based on the characteristics of the planning domain. Our approach also considers conflicts in the agents' beliefs and these are transformed into the critical questions used in dialogue game. Furthermore, we consider conflicts in the plan itself and the social context to evaluate more aspects of the proposal.

In Toniolo, Sycara, and Norman (2011), the authors define a mechanism to enable agreements to be reached regarding a shared plan using argumentation schemes. The main difference with our approach is that we use a critiquing dialogue, where Toniolo et al. (2011) use a deliberative dialogue that focuses on resolving conflicts in the action representation and existing agent commitments. A set of rules allow agents to formulate arguments in the dialogue (arguments for plan constraints, norms and goals), whereas in our approach, these guidelines generate arguments given by the set of critical questions and the strategy used.

Another related approach is presented in Onaindia, Sapena, and Torreño (2010), where the authors present the problem of solving cooperative distributed planning tasks through an argumentation-based model. The model allows agents to exchange partial solutions, express opinions on the adequacy of candidate solutions and adapt their own proposals for the benefit of the overall task. The argumentation-based model is designed in terms of argumentation schemes and critical questions whose interpretation is given through the semantic structure of a partial order planning paradigm. The approach assumes a lack of uncertainty and deterministic planning actions, and so, focuses only on questions concerned with the choice of actions. The argumentation scheme, based on the scheme for action proposal from Atkinson et al. (2005) is of the form: *in the current circumstances and considering the current base plan* Π_i*, agent* ag_i *should perform the refinement step* Π'*, which will result in a new partial plan* Π_j*, which will realise some sub-goals G, which will promote some values V.* Our work is similar to this work in the sense that plans are entities treated at a detailed level when arguing about them. We go further, however, and consider plan proposals in even more detail referring to action elements and combinations of actions. We believe that this gives agents more ways (more questions) to critique plans so they are able to evaluate plans more thoroughly and therefore select better plans but we are not in the position (it is not our implementation objective) to ensure that the approach gives better plans. A comparative evaluation is needed to make such a claim. Our argumentation scheme is related to a more comprehensive set of critical questions, that together with a strategy to select critical questions, gives agents more options to critique a proposal. Specifically, the approach in Onaindia et al. (2010) is based on refinement steps created from instantiations of an argumentation scheme, whereas our approach is that we map these refinements into specific questions at the dialogue level, giving the agents the possibility to argue over specific problems that would not be detected with other approaches. Therefore, the elements presented allow an agent to question and/or attack the argument in a more targeted fashion, facilitating the modification of more types of plans and specific identification of differences between participants.

8. Conclusions

Planning is known to be a highly complex and detailed problem due to the need to represent and reason about a large number of elements. We have shown how an argumentation-based approach can capture these elements but at the cost of needing to select from a very large number of moves when critiquing a plan proposal, placing a high premium on an effective strategy for move selection. Our experiments confirm that this is the case and how even a simple strategy can greatly assist in the move selection process.

The approach to plan selection presented in this article provides means for agents to cooperate, while allowing the agents to reach agreements that reflect their individual preferences. We showed that the use of a strategy when selecting a question in a dialogue regarding plans is beneficial, although different strategies performed better in different cases. We identified the characteristics which influence the performance of the strategies. The strategy where possibility questions are put first in the questioning order performs better for most cases because inconsistencies are normally found mainly in the plan representation. We believe that the strategy should be tailored to the context in which the dialogue develops and modified as the dialogue develops to reduce the exchange of information.

To summarise our approach on the strategies, the alignment of belief relies on the 'identification' of relevant critical questions about the validity of elements but the resolution of these inconsistencies (introduced by questions) is influenced by the order in which validity questions are put forward in the dialogue. The order in which questions are put forward in a dialogue does make a difference as shown in the strategies comparison. Furthermore, strategies to select critical questions reduce to some extent the overhead in communication and this could help in some multi-agent environments where communication is more costly than internal computation.

One of the main differences of our approach from standard distributed planning approaches is when and how agents discuss the best course of action to take. In our approach, we use a persuasion dialogue to critique plans between agents, assuming that the plans are already defined. The dialogue can then focus on the evaluation of plans considering agents' potentially different beliefs about the world and different preferences. The preferences are applied in the dialogue to choose the best-possible plan respecting the preferences of both agents. We believe that this approach presents an advantage over distributed planning approaches where knowledge-bases are first merged, then plans are created, and there is no clear indication as to where and when agents apply preferences over actions or plans. Furthermore, in our approach, we map these refinements to the plan into specific questions giving the agents the possibility to argue over specific problems in a more targeted fashion. Therefore, the elements presented allow an agent to question and/or attack the argument, facilitating the modification of plans and identifying problems that would not be detected with other approaches.

With the strategy, we want an approach to critical question selection that helps to reach agreements more quickly and in a more natural way. We showed that when using a strategy, agents reach an agreement more quickly in particular scenarios. In addition, the plan selection could be done in a more natural way as a result of the structured exchange of arguments using argumentation schemes and critical questions. A future research direction is related to the fact that in some of the test cases (A5, C5), there is no outcome because the plans gets rejected based on their side effects. This suggests that selecting an appropriate question to question or challenge a proposal based on the other agent's preference is relevant. In Black and Atkinson (2011), the authors present a dialogue system that allows agents to exchange arguments in order to come to an agreement on how to act using a model of what is important to the recipient agent. Assuming an agent can reason about or engage in a dialogue to ask the other agent's preference, our strategy could take into account this factor when choosing which question to pose.

Furthermore, we intend to extend our evaluation to look at other strategies that further change the question ordering to see if this has any effect on the outcome of a dialogue and provide new benchmarks against which our strategies can be compared. Finally, in future work, we also plan to consider issues connecting time with action combinations, so as to better understand the potentially complex inter-relationships of durative actions and their combinations.

Acknowledgements

Rolando Medellin-Gasque is grateful to CONACYT of Mexico for financial support.

Notes

1. The 'social context' was an extension to the argumentation scheme presented in Atkinson et al. (2006) and introduced in Atkinson, Girle, McBurney, and Parsons (2009), where agents rely on a social structure to issue valid commands in a command dialogue scenario.
2. We use the terms 'initial state' and 'current circumstances' interchangeably in this article.
3. 'Values' are qualitative social interests of agents following Atkinson et al. (2006).
4. The PDDL is an attempt to standardise planning domain and problem description languages developed for the International Planning Competitions.
5. A side effect is an outcome of the action that was unintended, and could in principle promote or demote a value, though our implementation in Section 5 currently considers only negative side effects.
6. In Torreño, Onainda, and Sapena (2010), the ontology alignment problem is discussed for a similar scenario.
7. In our example, we assume that actions are meant to be executed simultaneously by both agents, although often this is not the case in multi-agent planning where task allocation can be an important feature.
8. We may revisit this restriction in future work.
9. Certain questions are not required because of our assumption that actions are executed by both agents simultaneously.

References

Allen, J.F., Schubert, L.K., Ferguson, G., Heeman, P., Hwang, C.H., Kato, T., Light, M., Martin, N.G., Miller, B.W., Poesio, M., and Traum, D.R. (1995), 'The TRAINS Project: A Case Study in Building a Conversational Planning Agent', *Journal of Experimental and Theoretical Artificial Intelligence*, 7, 7–48.

Amgoud, L., and Hameurlain, N. (2007), 'Argumentation in Multi-Agent Systems', in *An Argumentation-Based Approach for Dialog Move Selection* (Vol. 4766), eds. N. Maudet, S. Parsons, and I. Rahwan, Berlin, Heidelberg: Springer, pp. 128–141.

Atkinson, K., and Bench-Capon, T. (2007), 'Practical Reasoning as Presumptive Argumentation Using Action Based Alternating Transition Systems', *Artificial Intelligence*, 171, 855–874.

Atkinson, K., Bench-Capon, T., and McBurney, P. (2005), 'A Dialogue Game Protocol for Multi-Agent Argument over Proposals for Action', *Autonomous Agents and Multi-Agent Systems*, 11, 153–171.

Atkinson, K., Bench-Capon, T., and McBurney, P. (2006), 'Computational Representation of Practical Argument', *Synthese*, 152, 157–206.

Atkinson, K., Girle, R., McBurney, P., and Parsons, S. (2009), ‘Command Dialogues’, in *Argumentation in Multi-Agent Systems*, eds. I. Rahwan and P. Moraitis, Fifth International Workshop, Berlin, Heidelberg: Springer-Verlag, pp. 93–106.

Belesiotis, A., Rovatsos, M., and Rahwan, I. (2010), ‘A Generative Dialogue System for Arguing About Plans in Situation Calculus’, in *Proceedings of the 6th International Conference on Argumentation in Multi-Agent Systems*, Budapest, Hungary, ArgMAS’09, Berlin, Heidelberg: Springer-Verlag, pp. 23–41.

Bench-Capon, T., and Dunne, P.E. (2007), ‘Argumentation in Artificial Intelligence’, *Artificial Intelligence*, 171, 619 – 641.

Black, E., and Atkinson, K. (2011), ‘Choosing Persuasive Arguments for Action’, in *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, Taipei, Taiwan, AAMAS ’11, Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, pp. 905–912.

Denti, E., Natali, A., and Omicini, A. (1998), ‘On the Expressive Power of a Language for Programming Coordination Media’, in *Proceedings of the 1998 ACM Symposium on Applied Computing (SAC98)*, ACM, New York, NY, USA, pp. 169–177.

Dunin-Keplicz, B., and Verbrugge, R. (2003), ‘Dialogue in Teamwork’, in *Enhanced Interoperable Systems. Proceedings of the 10th ISPE International Conference on Concurrent Engineering (ISPE CE 2003)*, eds. R. Jardim-Gonçalves, J. Cha and A. Steiger-Garção, 26–30 July 2003, Madeira, Portugal: A.A. Balkema Publishers, pp. 121–128.

Dunne, P.E., and Bench-Capon, T. (2003), ‘Two Party Immediate Response Disputes: Properties and Efficiency’, *Artificial Intelligence*, 149, 221–250.

Durfee, E.H. (2001), ‘Multi-Agents Systems and Applications’, in *Distributed Problem Solving and Planning* (Vol. 2086), eds. J.G. Carbonell and J. Siekmann, Berlin: Springer-Verlag, pp. 118–149.

FIPA, (2002), ‘Communicative Act Library Specification’, Standard SC00037J, Foundation for Intelligent Physical Agents.

Fox, M., and Long, D. (2003), ‘PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains’, *Journal Artificial Intelligence Research*, 20, 61–124.

Gelernter, D., and Carriero, N. (1992), ‘Coordination Languages and Their Significance’, *Communications of the ACM*, 35, 97–107.

Ghallab, M., Isi, C.K., Penberthy, S., Smith, D.E., Sun, Y., and Weld, D. (1998), ‘PDDL – The Planning Domain Definition Language’, Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.

Heras, S., Navarro, M., Botti, V., and Julián, V. (2010), ‘Argumentation in Multi-Agent Systems’, in *Applying Dialogue Games to Manage Recommendation in Social Networks* (Vol. 6057), eds. P. McBurney, I. Rahwan, S. Parsons, and N. Maudet, Berlin, Heidelberg: Springer, pp. 256–272.

van der Hoek, W., Roberts, M., and Wooldridge, M. (2007), ‘Social Laws in Alternating Time: Effectiveness, Feasibility, and Synthesis’, *Synthese*, 156, 1–19.

Hulstijn, J. (2000), ‘Dialogue Models for Inquiry and Transaction’, PhD Thesis, Universiteit Twente, Enschede, The Netherlands.

desJardins, M.E., Durfee, E.H., Ortiz, C.L., Jr., and Wolverton, M.J. (2000), ‘A Survey of Research in Distributed, Continual Planning’, *AI Magazine*, 20(4), 13–20.

McBurney, P., Hitchcock, D., and Parsons, S. (2007), ‘The Eightfold Way of Deliberation Dialogue’, *International Journal of Intelligent Systems*, 22, 95–132.

McBurney, P., and Parsons, S. (2004), ‘Locutions for Argumentation in Agent Interaction Protocols’, in *Agent Communication. Revised Proceedings of the International Workshop on Agent Communication (AC2004)*, eds. R.M. van Eijk, M.P. Huget, and F. Dignum, July, Vol. 3396, Berlin: Springer, pp. 209–225.

McBurney, P., and Parsons, S. (2009), ‘Dialogue Games for Agent Argumentation’, in *Argumentation in Artificial Intelligence* (chap. 13) eds. I. Rahwan and G. Simari, Berlin: Springer, pp. 261–280.

Medellin-Gasque, R., Atkinson, K., McBurney, P., and Bench-Capon, T. (2011), ‘Arguments Over Co-operative Plans’, in *Theory and Applications of Formal Argumentation. First International Workshop, TAFA 2011*, July, Lecture Notes in Computer Science (LNCS) 7132, Berlin: Springer, pp. 50–66.

Medellin-Gasque, R., Atkinson, K., and Bench-Capon, T. (2012a), ‘Dialogue Game Protocol for Co-operative Plan Proposals’, Technical Report ULCS-12-003, Department of Computer Science, University of Liverpool, UK.

Medellin-Gasque, R., Atkinson, K., and Bench-Capon, T. (2012b), 'An Analysis Over Critical Questions for Plan Proposals', Technical Report ULCS-12-002, Department of Computer Science, University of Liverpool, UK.

Omicini, A., and Denti, E. (2001), 'From Tuple Spaces to Tuple Centres', *Science of Computer Programming*, 41, 277–294.

Omicini, A., and Zambonelli, F. (1999), 'Coordination for Internet Application Development', *Autonomous Agents and Multi-Agent Systems*, 2, 251–269.

Onaindia, E., Sapena, O., and Torreño, A. (2010), 'Cooperative Distributed Planning through Argumentation', *International Journal of Artificial Intelligence*, 4, pp. 118–136, CESER Publications.

Prakken, H. (2010), 'An Abstract Framework for Argumentation with Structured Arguments', *Argument and Computation*, 1, 93–124.

Tang, Y., Norman, T.J., and Parsons, S. (2009), 'A Model for Integrating Dialogue and the Execution of Joint Plans', in *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, Budapest, Hungary, Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, pp. 883–890.

Toniolo, A., Sycara, K., and Norman, T. (2011), 'Argumentation Schemes for Policy-Driven Planning', in *Proceedings of the First International Workshop on the Theory and Applications of Formal Argumentation (TAFA)*, July, Barcelona, Spain.

Torreño, A., Onaindía, E., and Sapena, O. (2010), 'Reaching a Common Agreement Discourse Universe on Multi-Agent Planning', in *5th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2010)*, Vol. 6077, Springer, pp. 185–192.

Walton, D.N. (2005), 'Justification of Argumentation Schemes', *Australasian Journal of Logic*, 3, 1–13.

Wilkins, D.E., and Myers, K.L. (1998), 'A Multiagent Planning Architecture', in *Proceedings 4th International Conference on Artificial Intelligence Planning Systems*, Pittsburgh, PA: AAAI Press, pp. 154–162.

Appendix 1. AATS definition

An *AATS* is an $(n + 7)$-tuple of the form:

$$S = \langle Q, q_0, Ag, Ac_1, \ldots, Ac_n, \rho, \tau, \Phi, \pi \rangle,$$

where

- Q is a finite non-empty set of states;
- $q_0 \in Q$ is the initial state;
- $Ag = \{1, \ldots, n\}$ is a finite non-empty set of agents;
- Ac_i is a finite, non-empty set of actions, for each $i \in Ag$, where $Ac_i \cap Ac_j = \emptyset$ for all $i \neq j \in Ag$; now we can say that a joint action j_{Ag} for the set of agents Ag is a tuple $(\alpha_i, \ldots, \alpha_n)$, where for each $\alpha_j (j \leq n)$, there is some $i \in Ag$ such that $\alpha_j \in Ac_i$. We denote the set of all joint-actions J_{AG}. Given an element j of J_{AG} and an agent $i \in Ag$, $i's$ action in j is denoted by j_i. If we have specific actions for more than one agent, we call this an *agent combination*, e.g. $\{j_i, j_j\}$ will represent an action combination for agents i and j.
- $\rho : Ac_{Ag} \rightarrow 2^Q$ is an action precondition function, which for each action $\alpha \in Ac_{Ag}$ defines the set of states $\rho(\alpha)$ from which α may be executed;
- $\tau : Q \times J_{Ag} \rightarrow Q$ is a partial system transition function, which defines the state $\tau(q, j)$ that would result by the performance of j from state q, note that, as this function is partial, not all joint actions are possible in all states (cf. the precondition function above);
- Φ is a finite, non-empty set of atomic propositions; and
- $\pi : Q \rightarrow 2^\Phi$ is an interpretation function, which gives the set of atomic propositions satisfied in each state: if $p \in \pi(q)$, then this means that the propositional variable p is satisfied (equivalently, true) in state q.

In Atkinson and Bench-Capon (2007), the authors extended this transition system to enable representation of a theory of practical reasoning related to arguments about action through values were added to the system. The extensions are as follows:

- Av_i, is a finite, non-empty set of values $Av_i \subseteq V$, for each $i \in Ag$.
- δ: $Q \times Q \times Av_{Ag} \rightarrow \{+, -, =\}$ is a valuation function which defines the status (promoted(+), demoted(−) or neutral (=)) of a value $v_u \in Av_{Ag}$ ascribed by the agent to the transition between two states: $\delta(q_x, q_y, v_u)$ labels the transition between q_x and q_y with one of $\{+, -, =\}$ with respect to the value $v_u \in Av_{Ag}$.

Appendix 2. Dialogue simulation run

```
\\------Run C2-----
\\Agent Proponent John (AP) presents argument for plan p1
\\Agent Respondent Paul (AR) challenges the proposal
\\with critical questions.

Loading Protocol\ldots
[PCADA] Open Dialogue - Agent Proponent
[PCADA] Enter Dialogue - Agent Respondent

\\Proponent selects preferred plan and presents it as a plan
\\proposal in the Tuple centre
[Test] PLAN NAME: p1 - flight
[PCADA] Agent Proponent - Agent 1- Set Active Plan
[PCADA] Agent Proponent Create Proposal Tuple
[PCADA] Post Proposal Tuples AP

\\Respondent acknowledges proposal , identify questions
\\and apply strategy
[PCADA] Identify Questions AR
[PCADA] Applying Ordering Strategy
[PCADA] Posting Questions AR

\\Respondent challenges current circumstance(5234) possibility
[PCADA] Question: cQPP05
[PCADA] Question question(10,8,proposal5503,4,cQPP-05,5234,null,false,
currentCirc_POSSIBILITY)

\\Respondent agent provides evidence and rejects the attack.
[PCADA] Proponent provides evidence
[PCADA] Assert assert(10,7,proposal5503,cQPP-05,evidence,ValidToken)

\\Respondent challenges norm validity
[PCADA] Question: cQPP02
[PCADA] Question question(10,8,proposal5503,1,cQPP-02,0,null,false,
norm_VALIDITY)
[PCADA] Proponent provides evidence
[PCADA] Assert assert(10,7,proposal5503,cQPP-04,evidence,ValidToken)

\\Respondent challenges current circumstance validity
[PCADA] Question: cQPP04
[PCADA] Question question(10,8,proposal5503,4,cQPP-04,5236,null,false,
currentCirc_VALIDITY)
[PCADA] Proponent provides evidence
[PCADA] Assert assert(10,7,proposal5503,cQPP-04,evidence,ValidToken)

\\No more questions identified for this proposal
\\Respondent accepts the proposal
```

```
[PCADA] Accept Proposal AR

\\Roles between agents change when alternative plan
\\ question is considered
\\Agent Proponent Paul
\\Agent Respondent John

[PCADA] Question question(10,8,proposal5503,6,cQAO-01,p4 -coach-train
,null,false,alternateOption_Plan)

\\Evaluating alternative plan p4
[Test] Alternative PLAN NAME: p4 - coach-train

[PCADA] Agent Proponent - Agent 2 - Set Active Plan
[PCADA] Agent Proponent Create Proposal Tuple
[PCADA] Post Proposal Tuples AP
[PCADA] Identify Questions AR
[PCADA] Applying Ordering Strategy
[PCADA] Posting Questions AR

\\Respondent agent challenges action tale train possibility in time
[PCADA] Question: cQAT01
[PCADA] Question question(10,8,proposal5503,2,cQAT-01,5361,
takeTrain_LP,false,action_POSSIBILITY_TIME)
[PCADA] Proponent provides evidence
[PCADA] Assert assert(10,7,proposal5503,cQAT-01,evidence,ValidToken)

\\Respondent challenges end effect possibility.
[PCADA] Question: cQA12
[PCADA] Question question(10,8,proposal5503,1,cQA-12,5350,5359,true,
endEffect_POSSIBILITY)
[PCADA] Retract Evidence Proponent
[PCADA] Retract -retract(10,7,proposal5503)
[PCADA] Retract plan because of question cQA-12

\\Two plan proposals evaluated. Six questions in total
Total number of proposals: 2
Total number of questions: 6
Outcome Agent -7 p1 - flight
------------------------------------
[PCADA] Leave Dialogue Agent Questioner
[PCADA] Leave Dialogue Agent Proponent
[------Dialogue Finished-----Run2c-A1-S2-----
[------------------------------------------------------------------
```