Autonomous Agents and Multi-Agent Systems (2018) 32:861–885
https://doi.org/10.1007/s10458-018-9396-x

A dialogue-based approach for dealing with uncertain and conflicting information in medical diagnosis

Chunli Yan[1] · Helena Lindgren[1] · Juan Carlos Nieves[1]

Published online: 29 September 2018

Abstract

In this paper, we propose a multi-agent framework to deal with situations involving uncertain or inconsistent information located in a distributed environment which cannot be combined into a single knowledge base. To this end, we introduce an inquiry dialogue approach based on a combination of possibilistic logic and a formal argumentation-based theory, where possibilistic logic is used to capture uncertain information, and the argumentation-based approach is used to deal with inconsistent knowledge in a distributed environment. We also modify the framework of earlier work, so that the system is not only easier to implement but also more suitable for educational purposes. The suggested approach is implemented in a clinical decision-support system in the domain of dementia diagnosis. The approach allows the physician to suggest a hypothetical diagnosis in a patient case, which is verified through the dialogue if sufficient patient information is present. If not, the user is informed about the missing information and potential inconsistencies in the information as a way to provide support for continuing medical education. The approach is presented, discussed, and applied to one scenario. The results contribute to the theory and application of inquiry dialogues in situations where the data are uncertain and inconsistent.

Keywords Multi-agent system · Inquiry dialogues · Possibilistic logic · Argumentation framework

This is a revised and extended version of the paper *A multi-agent system for nested inquiry dialogues* which appeared in Y. Demazeau, F. Zambonelli, J.M. Corchado, J. Bajo (eds.) Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection. PAAMS 2014. Lecture Notes in Computer Science, vol. 8473 pp. 303–314. Springer, Cham (2014).

✉ Chunli Yan
yan_chunli@yahoo.com

Helena Lindgren
helena.lindgren@umu.se

Juan Carlos Nieves
juan.carlos.nieves@umu.se

[1] Department of Computing Science, Umeå University, Umeå, Sweden

1 Introduction

Clinical decision-support systems (CDSSs) use Artificial Intelligence (AI) to help doctors reach diagnostic decisions and are playing an increasingly important role in clinical practice [29]. Knowledge bases are key elements in CDSSs. As the decisive factor for the precision in the decision-making process, they constitute the basis for the system. In practice, however, there are situations where two or more knowledge bases cannot be merged or efficiently combined due to security, privacy, or other concerns, but the knowledge stored therein is needed in the decision-making process. In such cases, a multi-agent system (MAS) [8] is a natural choice for a CDSS solution with practical usefulness. Moreover, a MAS can illustrate different viewpoints in a clinical teamwork situation, e.g., when a primary care practitioner treats a patient with progressing dementia without having much experience in this particular disease domain and may need to consult an expert physician during the diagnostic process. In sparsely populated areas, it is also common that experts may be located at specialist hospitals a long distance from the primary care center. In our work, we take this perspective a step further by building a decision-support system based on international consensus on diagnostic criteria [22], which may act as an expert agent in a MAS that can be consulted through the CDSS. We also choose dialogue-based systems to simulate the dialogue that a clinician could have with an expert physician in a patient case for educational purposes.

Walton and Krabbe defined several types of agent dialogues [34,42]. Among them, the *inquiry dialogue* allows agents to collaborate in order to find the best solution and new knowledge. However, the value of inquiry dialogue-based MASs in dealing with situations where knowledge bases could not be easily put together has only been explored to a limited extent. For instance, Black and Hunter describe an inquiry dialogue approach in [2]. However, to the best of our knowledge, their theories have not been implemented in practical applications. Actually, some of their theories are not straightforward to realize in applications for clinical situations.

Against this background, we have extended Black and Hunter's theoretical dialogue approach [2], which allows for a significantly simplified implementation. This modification is mainly in the structure. In Black and Hunter's structure, a top level *warrant inquiry* (*wi*) dialogue consists of several *argument inquiry* (*ai*) dialogues at different hierarchical levels, whereas in our structure, we have *wi* dialogue not only in the top level but also in all sublevels. The modified structure allows agents to reach partial conclusions within the nested dialogues, a method in which the strongest arguments are aggregated to serve the argument evaluation for deciding upon the major topic. Therefore, the structure of our system is clearer, and the system becomes time efficient, which is also in line with how humans reason. Details for generating *wi* and *ai* dialogues are provided, and a strategy for dealing with endless loops is given.

Moreover, since the agents have different roles and knowledge bases that include rules and behaviors, the data to be fused are often *inconsistent* and *uncertain*. In our earlier research [21,44], inquiry dialogues were explored, which were based on *defeasible logic* and integrated *preference levels*. The purpose was to integrate different preferences regarding knowledge sources (e.g., different potentially conflicting clinical practice guidelines). However, the ability of *defeasible logic programming* to deal with explicit uncertainty and fuzzy knowledge which often exist in the medical domain is fundamentally limited [4]. Thus, the approach presented in this article is built on the formal argumentation for dealing with inconsistent information introduced in [44]. We extend our earlier research by integrating *possibilistic logic* to capture uncertain information and degrees of confidence in sources of knowledge [6]. The reason for applying possibilistic logic in the clinical dialogue between agents is to extend defeasible logic programming capabilities for qualitative reasoning by incorporating the treatment of possibilistic uncertainty and vague knowledge [4]. It has applicability in specific individual patient cases. Reasoning with probabilities that require statistical information presents problems since the statistical distributions provided by evidence-based medicine (EBM) are not coherent; there are overlapping conditions [13], and the individual case may not be a standard case. In contrast, reasoning with possibilities gives a different starting point, where if knowing nothing at all, all potential diagnoses are possible, which may help the novice clinician to not jump to conclusions too soon [35]. This approach also follows the terminology applied in some diagnostic criteria where the medical community has tried to translate the uncertainty in EBM-based statistical information into formulations useful in clinical practice using terms like possible, probable, and unlikely (e.g., [28]). This approach allows to evaluate arguments based on the reliability of the source of generic knowledge (e.g., clinical diagnostic criteria) or the source of patient information (e.g., the patient vs. a relative).

Hence, using the methods we develop based on possibilistic logic and argumentation theory, traditionally difficult situations where the data are uncertain and inconsistent can now be properly dealt with transparently. This approach provides transparency so the clinician can follow the reasoning and decision-making process and make more well-founded medical decisions, and using the system will also provide continued medical education during everyday clinical practice. Consequently, our results contribute significantly to the research field of knowledge-based systems and practical application in the medical domain.

We implemented our methodologies in the CDSS that we developed for dementia diagnosis and management. The results are illustrated with representative diagnostic situations from the dementia domain. The domain knowledge related to dementia is a good example of a medical domain where uncertain and conflicting information is present [19,22] and where interactive decision support could increase the physician's knowledge [20].

The paper is organised as follows. The next section presents a short background of possibilistic logic. In Sect. 3, the developed methods for dialogue generation are described. Section 4 describes the details of the realisation of the MAS embedded in the clinical decision-support system. Next, one scenario is used to show how MAS is integrated into the clinical system and how it works. We compare our work with other related research in Sect. 6. The paper ends with conclusions and an outline of future work.

2 Background: possibilistic knowledge bases

Our approach is based on *Possibilistic logic* [6]. We begin by presenting the syntax of our rule-based knowledge bases. In the logic we used, formulas are expressed by classical propositional logic with attached numbers between 0 and 1. In the settings of possibilistic logic, we consider the framework of necessity formulas; hence, the numbers attached to logic formulas denote the degree of necessity of a formula. A *literal* denotes either an atom α or its negation $\neg\alpha$. The contradiction of a literal l is defined as c, such that $c(l) \equiv \neg l$ and $c(\neg l) \equiv l$. The symbols, such as binary connectives $\wedge$, implication $\rightarrow$, and negation $\neg$ are the same as in propositional logic.

A **rule** is denoted by $\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta$, such that α_i $(1 \leq i \leq n)$, and β are literals. α_i $(1 \leq i \leq n)$ is called the **premise** of the rule and β is called the **conclusion** of the rule. Given a rule $r = \alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta$, $concl(r) = \beta$. When $n = 0$, a rule is denoted by $\rightarrow \alpha$ and called a $fact$.

A rule together with an attached number p form a possibilistic belief. p is not a probability (like it is in probability theory), but it induces a certainty (or confidence) scale. This value is determined by the expert providing the knowledge base.

Definition 1 A **possibilistic belief**, denoted by B, is a tuple of the form (ϕ, p) where ϕ is a rule and $p \in (0, 1]$ is the lower bound of the belief in terms of necessity measures, which means the formula ϕ is certain at least to the level p. Given a belief (ϕ, p), if ϕ is a fact, we name this belief **state belief**; otherwise, we name it **domain belief**.

In this paper, unless otherwise stated, a *possibilistic belief* is called a *belief* since all beliefs in this paper are possibilistic. We name the rule's premise as the **belief's premise** and the rule's conclusion as the **belief's conclusion**. Given a belief $B = (\phi, p)$, $N(\phi) \geq p$ where N is a necessity measure modeling the possibly incomplete state knowledge [6,31], and $B^* = \phi$. (l, p) is called a possibilistic literal where l is a literal, and p is its possibilistic value.

A **belief base** of an agent x, denoted by Σ_{x}, is a finite set of beliefs. The set of all rules in Σ_{x} is denoted by $\Sigma_{\mathrm{x}}^* = \{\phi | (\phi, p) \in \Sigma_{\mathrm{x}}\}$.

To project a set of rules from a belief base of a given agent x regarding a particular conclusion, we define the concept of *related belief base* as follows:

Definition 2 The **related belief base** about literal α regarding agent x, denoted by $\Sigma_{\mathrm{x}}^{\alpha}$ is defined as
$\Sigma_{\mathrm{x}}^{\alpha} = \{(\phi, p) | (\phi, p) \in \Sigma_{\mathrm{x}} \text{ and } (concl(\phi) = \alpha \text{ or } concl(\phi) = \neg\alpha)\}$
We use the function $relatedBeliefBase_x(\alpha)$ to return $\Sigma_{\mathrm{x}}^{\alpha}$ from Σ_{x}.

Following Ref. [6], the possibilistic inference rule, under the framework of necessity formulas, is $(\phi_1, p_1), (\phi_1 \to \phi_2, p_2) \vdash (\phi_2, min(p_1, p_2))$. The rule's necessity number is given by the minimal necessity numbers of all the participant formulas. We use $\vdash_{pl}$ to denote the inference in possibilistic logic. From this rule, if we know (ϕ, p_1) is true, we can infer that all beliefs $(\phi, p_2)(p_2 \leq p_1)$ are true.

An *argument* is usually defined as a set of propositions such that a set of premises supports a conclusion. Depending on the goal of a given dialogue, an argument can be used, for instance, to persuade someone of something or present reasons for accepting a conclusion.

From the knowledge representation perspective, we only consider a sublanguage of possibilistic logic, which is based on possibilistic beliefs. This sublanguage of possibilistic logic has been tailored to express the modelled knowledge by the knowledge elicitation tool implemented and presented in the application Dementia Diagnosis and Management Support System—Web version (DMSS-W) (see Sect. 4). Since the knowledge base is based on possibilistic beliefs that only have a literal in each of their heads, the conclusion of an argument will be a possibilistic literal.

Definition 3 An **argument** is a tuple of the form $\langle \Phi, (l, p) \rangle$ constructed from a belief base Σ, where Φ is a set of beliefs, and (l, p) is a possibilistic literal and called the **claim** of the argument, so that the following conditions hold:

1. $\Phi \subseteq \Sigma$ and $\Phi \vdash_{pl} (l, p)$,
2. Φ^* is consistent.
3. There is no $\Phi_1^* \subset \Phi^*$, such that item 1 and item 2 are satisfied.

In this definition, item 2 describes an argument as being derived from a consistent set of beliefs. Consistency is understood in the standard way as it does by classical logic which means that one cannot infer a and $\neg a$ from the same logic theory. Therefore, no conflicting arguments can be derived from the same set. Item 3 states that the consistent set is a minimally set (i.e. no subset of this consistent set can derive the argument with the same fact or its contradiction).

When we say the argument $A_1 = \langle \Phi_1, B_1 \rangle$ is a **subargument** of the argument $A_2 = \langle \Phi_2, B_2 \rangle$, we mean $\Phi_1 \subset \Phi_2$.

Definition 4 Let $A_1 = \langle \Phi_1, (l_1, p_1) \rangle$ and $A_2 = \langle \Phi_2, (l_2, p_2) \rangle$. A_1 **rebuts** A_2 iff $c(l_1) \equiv l_2$. A_1 **undercuts** A_2 iff there exists an argument $A_3 = \langle \Phi_3, (l_3, p_3) \rangle$ and A_3 is a subargument of A_2, such that $c(l_1) \equiv l_3$.

Rebut and undercut are also known as *counterargument* in literature [1].

Beliefs in a belief base are sometimes inconsistent, leading to conflicts between arguments constructed from this belief. In the following section, we introduce an argumentation-based approach to resolve conflicts and calculate the acceptability of a literal.

3 Modeling dialogues

In this section, we present our approach to generate dialogues and evaluate arguments between agents.

3.1 Dialogue representation

The *inquiry dialogue*, among other types of dialogues, was defined by Walton and Krabbe [42], with the purpose to collaboratively build new knowledge. The goal of an inquiry dialogue is to prove or contradict and possibly falsify the hypothesis in the proof process of collaborative reasoning. In our approach, there are always exactly two agents taking part in an inquiry dialogue. In the following sections, we use x to represent one agent and $\hat{x}$ to represent the other. That is, if $x = 1$ then $\hat{x} = 2$ and vice versa. Each agent has its own belief base Σ_x. However, they cannot prove the truth of the hypothesis by themselves; hence, they need to collaborate to find and verify the evidence regarding the given hypothesis (topic).

To formalize our dialogue system, we follow the dialogue style introduced by Black and Hunter [2]. Two participating agents use *moves* to communicate with each other in our argumentation system. Three types of moves are allowed: *open*, *assert*, and *close*. An *open* move means an agent opens a new dialogue. An *assert* move means an agent believes a given belief is true. A *close* move indicates an agent wants to close the current dialogue; however, if another agent does not agree, this dialogue will not be closed.

In the same way as in [2], we use two kinds of inquiry dialogues in our framework: *wi* and *ai* dialogues. Ai dialogue generates the argument that can be used in a wi dialogue (i.e. if all the premises of a rule can be proven true, an argument is generated). Wi dialogue contains 0 to n ai dialogues. Wi dialogue generates new knowledge by comparing these arguments. The two types of dialogues above are nested within each other.

A move m is a tuple of the form $\langle agent, move\ type, dialogue\ type, topic \rangle$, where *agent* is the agent that makes this move; *move type* denotes the kind of move (*open*, *assert* or *close*); *dialogue type* means the type of the dialogue (*wi* or *ai*); and *topic* is the content of the move. If the move is an open/close *wi* move, the $topic$ is a literal; if it is an open/close *ai* move, the $topic$ is a domain belief; otherwise, the $topic$ is a state belief. The topic of the dialogue is determined by the topic of the first move made in this dialogue, so the topic of a

Table 1 Move format

Move	Dialogue	Format
Open	wi	$\langle x,\ open,\ wi,\ \alpha\rangle$
Close	wi	$\langle x,\ close,\ wi,\ \alpha\rangle$
Open	ai	$\langle x,\ open,\ ai,\ (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta, p)\rangle$
Close	ai	$\langle x,\ close,\ ai,\ (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta, p)\rangle$
Assert	wi	$\langle x,\ assert,\ wi,\ (\rightarrow \alpha, p)\rangle$
Assert	ai	$\langle x,\ assert,\ ai,\ (\rightarrow \alpha, p)\rangle$

wi dialogue is a literal and the topic of an *ai* dialogue is a domain belief. Since we have two types of inquiry dialogues and three types of moves, there are a total of six types of move formats (Table 1). More details are listed below.

- $\langle x,\ open/close,\ wi,\ \alpha\rangle$ means that agent x opens/closes a wi dialogue, and the topic of the dialogue is α.
- $\langle x,\ open/close,\ ai,\ (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta, p)\rangle$ means that agent x opens/closes an ai dialogue and the topic of the dialogue is $(\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta, p)$.
- $\langle x,\ assert,\ wi,\ (\rightarrow \alpha, p)\rangle$ means that this move is within a wi dialogue whose topic is α or $\neg\alpha$, and agent x asserts that $(\rightarrow \alpha, p)$ is true.
- $\langle x,\ assert,\ ai,\ (\rightarrow \alpha, p)\rangle$ means this move is within an *ai* dialogue, and agent x asserts that $(\rightarrow \alpha, p)$ is true and α or $\neg\alpha$ is one of the *ai* topic's premises. Let us observe that $(\rightarrow \alpha, p)$ is not from the agent's belief base but from the *result store* (which will be described in next section in Definition 11) that has already been proved to be *true*, *false*, or *unknown* by the two agents.

To formalize our dialogue system, we use the definitions of *dialogue, sub-dialogue* and *well-formed dialogue* in [2]. Here, we only give a brief description. For detailed descriptions of these concepts, we refer to [2].

Definition 5 A **dialogue** D_r^t (r, t $\in \mathbb{N}$ and r$\leq$ t) is a sequence of moves $[m_r, \ldots, m_t]$ with two agents participating, in which: (1) the first move of the dialogue is an open move, and (2) each agent takes its turn to make moves. A **sub-dialogue** is a sub-sequence of another dialogue. A **well-formed dialogue** is a dialogue with the following conditions: (1) the last two moves must be close moves made by two agents successively, which means both agents agree to close the dialogue; (2) the dialogue only terminates once; and (3) all the sub-dialogues are also well-formed and terminate before their parent dialogue. We use the function $Topic(D_r^t)$ to return the topic of the dialogue D_r^t.

3.2 Generating dialogues

We first define some notations which are needed to generate the dialogues and then provide specific protocols for generating the two different types of dialogues: *wi* and *ai*.

In a wi dialogue, we use a *Possible Beliefs Queue* (*PBQ*) to store the *relatedBeliefBase* i.e., Definition 2, of a belief according to the dialogue's topic; consequently, it can pick up the first belief from this queue when it needs to make a move.

Definition 6 A **PBQ** is a queue of beliefs that the agent can legally use for selecting the next move from the current wi dialogue. Let D_r^t be the current dialogue and $\mathscr{I}$ be the set of participants. For all $x \in \mathscr{I}$,

Table 2 Meaning of each atom

Atom	Symptom/disease
a	Extrapyramidal symptoms are present
b	Sensitivity to neuroleptica is present
c	Dementia is present
d	Vascular symptoms are present
f	Lewy body dementia

$$PBQ_x^t(\alpha) = \begin{cases} \mathit{relatedBeliefBase}_x(\alpha), & \text{if } m_t = \langle x, open, wi, \alpha\rangle \text{ or} \\ & \quad m_{t-1} = \langle \hat{x}, open, wi, \alpha\rangle, \\ \mathit{relatedBeliefBase}_x(\alpha)\backslash\{(\phi, p)\}, & \text{if } m_t = \langle x, open, ai, (\phi, p)\rangle, \\ \mathit{relatedBeliefBase}_x(\alpha)\backslash\{(\rightarrow \alpha, p)\}, & \text{if } m_t = \langle x, assert, wi, (\rightarrow \alpha, p)\rangle, \\ PBQ_x^{t-1}(\alpha), & \text{otherwise.} \end{cases}$$

PBQ is used by an agent to select the next explicit move in a given wi dialogue. Each agent has its own PBQ where state beliefs and domain beliefs are both stored. When agent x opens a wi dialogue with topic α, it updates its PBQ, and at the next move, agent $\hat{x}$ updates its PBQ. Within the wi dialogue, the agent retrieves and deletes the first belief in its PBQ (since PBQ is a First-In-First-Out queue data structure) and uses this belief for its next move. If the belief is a state belief, the agent makes an *assert wi* move (see step 3.3.1 in Table 3); else, if it is a domain belief, the agent makes an *open ai* move (see step 3.3.2 in Table 3); else it makes a *close wi* move (see step 3.2 in Table 3) since the queue is empty.

Example 1 The example comes from a medical practice scenario. There are two agents: professional agent (PA) simulates a novice physician, and domain agent (DA) simulates a medical domain expert. PA investigates a patient regarding possible Lewy body dementia. However it does not have enough knowledge to reach a decision. Therefore, it consults DA who has more experience diagnosing Lewy body dementia. The belief bases associated with DA and PA are:

$\Sigma_{\text{DA}} = \{(a \wedge b \wedge c \rightarrow f, p_1), (d \rightarrow \neg f, p_1), (c, 1)\}$ and
$\Sigma_{\text{PA}} = \{(a \wedge c \rightarrow f, p_2), (a, 1), (d, 1)\}$.
p_1 and p_2 mean $probable$ and $possible$, respectively, and $probable$ is more credible than $possible$. The meaning of the atoms represent the symptoms or the disease that are listed in Table 2.

If PA makes the move $\langle PA, open, wi, f\rangle$ at t = 1, the PBQ for each agent is (also shown in Fig. 1):

$$PBQ_{PA}^1(f) = \{(a \wedge c \rightarrow f, p_2)\}$$
$$PBQ_{DA}^2(f) = \{(a \wedge b \wedge c \rightarrow f, p_1), (d \rightarrow \neg f, p_1)\}$$

When an agent opens an ai dialogue with the topic (Φ, p), a *Query Store* (*QS*) associated with this topic is created and shared between the agents. Within an ai dialogue, if an agent needs to make a move, it can consult the QS w.r.t. a specific topic, retrieve the first fact, and then make an open wi move.

Table 3 The wi dialogue protocol

1, Agent x starts a wi dialogue D_r^t with the move $m_r = \langle x, open, wi, \alpha\rangle$ and let $t = r$.

2, Both agents x and $\hat{x}$ update their PBQs according to Definition 6.

3, Loop

  3.1, Set x' = the opposite agent and t=t+1.

  3.2, If $PBQ^t_{x'}(\alpha)$ is empty, $m_t = \langle x', close, wi, \alpha\rangle$. If $m_{t-1} = \langle x, close, wi, \alpha\rangle$, the wi dialogue closes and goes to 5.

  3.3, Else get the first belief (ϕ, p) in $PBQ^t_{x'}(\alpha)$ and delete it.

    3.3.1, If (ϕ, p) is a state belief, $m_t = \langle x', assert, wi, (\phi, p)\rangle$. The CS of the agent x is updated according to Definition 8.

    3.3.2, Else $m_t = \langle x', open, ai, (\phi, p)\rangle$

4, End loop.

5, When the dialogue closes, the outcome of the wi dialogue is calculated according to Algorithm 1 or 2; and the RS is updated according to Definition 11.

Definition 7 For the current ai dialogue with topic (Φ, p), a **QS**, denoted by QS^t_{Φ}, is a queue with a finite number of literals, such that

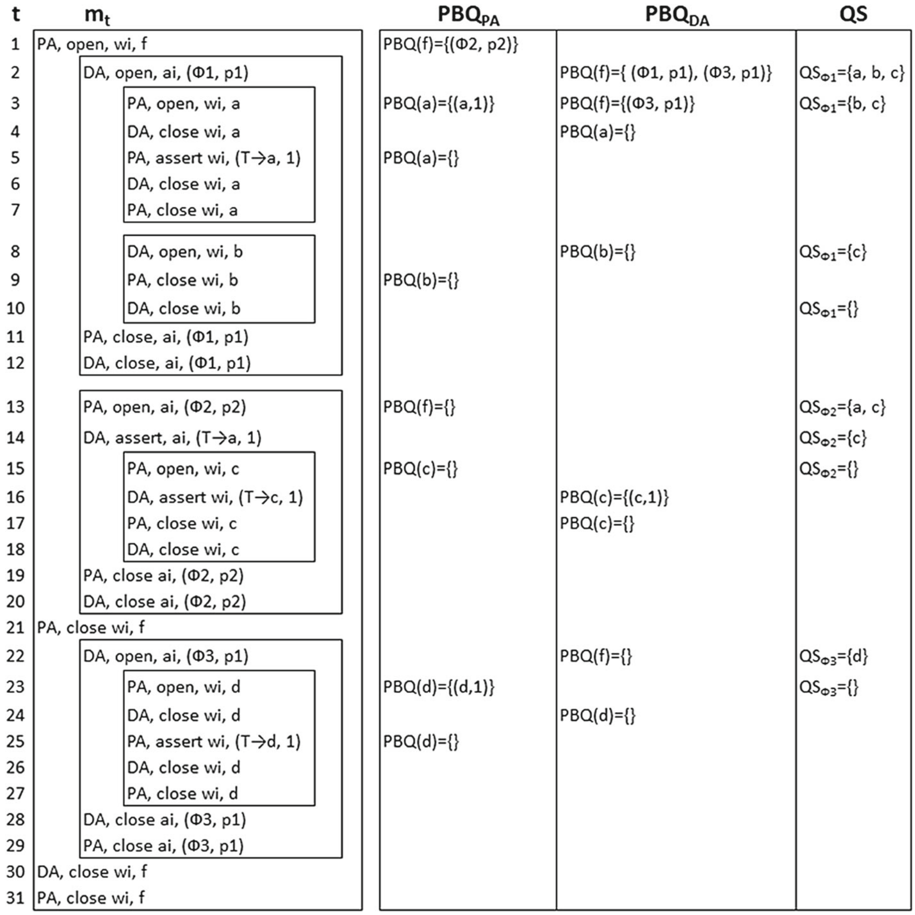
$$QS^t_{\Phi} = \begin{cases} \{\alpha_1, \ldots, \alpha_n\}, & \text{if } m_t = \langle x, open, ai, (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta, p)\rangle\,, \\ QS^{t-1}_{\Phi} - \{\alpha\}\,, & \text{if } (m_t = \langle x, open, wi, \alpha\rangle \text{ or } m_t = \langle x, assert, ai, (\rightarrow \alpha, p)\rangle) \\ & \quad \text{and } \alpha \in QS^{t-1}_{\Phi}, \\ \emptyset, & \text{if } (m_t = \langle x, assert, ai, (\rightarrow \neg\alpha, p)\rangle \text{ and } \alpha \in QS^{t-1}_{\Phi}), \\ QS^{t-1}_{\Phi}, & \text{otherwise.} \end{cases}$$

Both agents share the same QS, which only stores premises. The QS is used by the agents to select the next explicit move in a given ai dialogue. When an agent makes an *open ai* move, the premises of its topic rule are stored in the QS. Within this ai dialogue, if the QS is empty, the agent makes a *close ai* move (see step 3.2 in Table 4); otherwise, it makes an *open wi* or *assert ai* move. If the move $\langle x, open, wi, \alpha_k\rangle$ (see step 3.3.3 in Table 4) or $\langle x, assert, ai, (\rightarrow \alpha_k, p)\rangle$ is made (see step 3.3.1 in Table 4), the QS removes α_k. If the move $\langle x, assert, ai, (\rightarrow \neg\alpha_k, p)\rangle$ is made (see step 3.3.2 in Table 4), the QS resets $\emptyset$ directly since this premise is already proved to be false, and the ai dialogue can then be closed without consulting the QS anymore. Hence, the QS of a well-formed dialogue is empty in the last move of the dialogue.

Example 2 Continuing the running example started in Example 1, if DA makes the move $\langle DA, open, ai, (\Phi 1, p_1)\rangle$ where $\Phi 1 = (a \wedge b \wedge c \rightarrow f)$, then $QS^2_{\Phi 1} = \{a, b, c\}$ (also shown in Fig. 1 at t=2).

To manage committed data that becomes public to other agents, a *Commitment Store* (*CS*) is used, which is a set of possibilistic literals and initiated as an empty set. To identify the state of the CS of each agent which participates in a given dialogue D_r^t, CS_x^t denotes the CS of Agent x, and t denotes the timepoint in the dialogue D_r^t.

The updates of a CS, outcome of an ai dialogue ($Outcome_{ai}$), and outcome of a wi dialogue ($Outcome_{wi}$) are obtained recursively. To update CS, we need to get $Outcome_{ai}$. To get $Outcome_{ai}$, we need to calculate $Outcome_{wi}$. To calculate $Outcome_{wi}$, we need to know CS. The update of the CS of each agent is done as follows ($Outcome_{ai}$ and $Outcome_{wi}$ will be defined in Definition 9 and Definition 10, respectively).



t	m$_t$	PBQ$_{PA}$	PBQ$_{DA}$	QS
1	PA, open, wi, f	PBQ(f)={(Φ2, p2)}		
2	DA, open, ai, (Φ1, p1)		PBQ(f)={ (Φ1, p1), (Φ3, p1)}	QS$_{Φ1}$={a, b, c}
3	PA, open, wi, a	PBQ(a)={(a,1)}	PBQ(f)={(Φ3, p1)}	QS$_{Φ1}$={b, c}
4	DA, close wi, a		PBQ(a)={}	
5	PA, assert wi, (T→a, 1)	PBQ(a)={}		
6	DA, close wi, a			
7	PA, close wi, a			
8	DA, open, wi, b		PBQ(b)={}	QS$_{Φ1}$={c}
9	PA, close wi, b	PBQ(b)={}		
10	DA, close wi, b			QS$_{Φ1}$={}
11	PA, close, ai, (Φ1, p1)			
12	DA, close, ai, (Φ1, p1)			
13	PA, open, ai, (Φ2, p2)	PBQ(f)={}		QS$_{Φ2}$={a, c}
14	DA, assert, ai, (T→a, 1)			QS$_{Φ2}$={c}
15	PA, open, wi, c	PBQ(c)={}		QS$_{Φ2}$={}
16	DA, assert wi, (T→c, 1)		PBQ(c)={(c,1)}	
17	PA, close wi, c		PBQ(c)={}	
18	DA, close wi, c			
19	PA, close ai, (Φ2, p2)			
20	DA, close ai, (Φ2, p2)			
21	PA, close wi, f			
22	DA, open, ai, (Φ3, p1)		PBQ(f)={}	QS$_{Φ3}$={d}
23	PA, open, wi, d	PBQ(d)={(d,1)}		QS$_{Φ3}$={}
24	DA, close wi, d		PBQ(d)={}	
25	PA, assert wi, (T→d, 1)	PBQ(d)={}		
26	DA, close wi, d			
27	PA, close wi, d			
28	DA, close ai, (Φ3, p1)			
29	PA, close ai, (Φ3, p1)			
30	DA, close wi, f			
31	PA, close wi, f			

Fig. 1 The nested dialogues generated by PA and DA in the running example. The values of PBQ, QS, and RS in each timepoint are given on the right side where only the updated or newly added information are shown. Here, $\Phi 1 = (a \wedge b \wedge c \rightarrow f)$, $\Phi 2 = (a \wedge c \rightarrow f)$ and $\Phi 3 = (d \rightarrow \neg f)$

Definition 8 Let D_r^t be the current dialogue and $\mathscr{I}$ be the set of participants. For all $x \in \mathscr{I}$,

$$CS_x^t = \begin{cases} \emptyset, & \text{if } t \leq 1, \\ CS_x^{t-1} \cup \{(\alpha, p)\}, & \text{if } m_t = \langle x, assert, wi, (\rightarrow \alpha, p)\rangle \text{ or} \\ & \quad Outcome_{ai}(D_r^t) = (\alpha, p), \\ CS_x^{t-1}, & \text{otherwise.} \end{cases}$$

The CS of each agent is initiated as an empty set. It is updated whenever the agent performs an *assert wi* move to assert a state belief or when the ai dialogue closes and a state belief is calculated according to $Outcome_{ai}$, which is defined in Definition 9. An important consequence of this update is that the information added to the CS is public to the other agents taking part in the given dialogue.

When an ai dialogue terminates, its outcome is calculated. If all the premises of its topic are considered to be true (i.e. $Outcome_{wi} = \langle T, p\rangle$, $Outcome_{wi}$ is given in Definition 10), the

Table 4 The ai dialogue protocol

1, Agent x starts an ai dialogue D_r^t with the move $m_r = \langle x, open, ai, (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \alpha, p)\rangle$ and let $t = r$.

2, The QS is updated according to Definition 7.

3, Loop

3.1, Set x' = the opposite agent and t = t + 1.

3.2, If $QS^t(D_r^t)$ is empty, $m_t = \langle x', close, ai, (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \alpha), p)\rangle$. If $m_{t-1} = \langle x, close, ai, (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \alpha), p)\rangle$, the ai dialogue closes and goes to 5.

3.3, Else

3.3.1, If $Result(\alpha_k)^t = T$ and $PL(\alpha_k)^t = p'$, $m_t = \langle x', assert, ai, (\rightarrow \alpha_k, p')\rangle (1 \leq k \leq n)$.

3.3.2, Else if $(Result(\alpha_k)^t = F$ or $Result(\alpha_k)^t = U)$ and $PL(\alpha_k)^t = p'$, $m_t = \langle x', assert, ai, (\rightarrow \neg\alpha_k, p')\rangle (1 \leq k \leq n)$.

3.3.3, Else $m_t = \langle x', open, wi, \alpha_k\rangle (1 \leq k \leq n)$.

3.3.4, Update QS according to Definition 7.

4, End loop.

5, When the ai dialogue terminates, the outcome of the dialogue is calculated according to Definition 9; and the CS is updated according to Definition 8.

outcome is a belief constructed with the rule's conclusion and a necessity value p; otherwise, the outcome is null.

Definition 9 Let D_r^t be a well-formed ai dialogue and $(\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta, p)$ be its topic. **Outcome of an ai dialogue** is defined as follows:

$$Outcome_{ai}(D_r^t) = \begin{cases} (\beta, p'), & \text{if } \forall \alpha_i (i \in \{1, \ldots, n\}), ((\ Outcome_{wi}(D_{r_i}^{t_i}) = \langle T, p_i\rangle \\ & \text{and } \ Topic(D_{r_i}^{t_i}) = \alpha_i) \text{ or} \\ & m_j = \langle x, assert, ai, (\rightarrow \alpha_i, p_i)\rangle (r < j < t)) \\ & \text{and } p' = min(p_i, p), \\ null, & \text{Otherwise.} \end{cases}$$

Within a wi dialogue, several arguments defending or attacking the topic α may be generated. When this wi dialogue terminates, its outcome is calculated. The outcome is a tuple $\langle r, p\rangle$ where $r \in \{T, F, U\}$. If the defending arguments win, $r = T$, meaning α is $True$. If the attacking arguments win, $r = F$, meaning α is $False$. In both cases, p is a necessity value which can be calculated from the Algorithms 1 or 2. However if the two sides are well matched, $r = U$, which means the result is *unknown* and p is null.

Definition 10 Let D_r^t be a well-formed ai dialogue and α be its topic. **Outcome of a wi dialogue** is the outcome of the function $Outcome_{wi}(D_r^t)$.

To meet different users' requirements, we provide two alternative algorithms (Algorithms 1 and Algorithms 2) to obtain $Outcome_{wi}$. The following notations are used in the two algorithms:

Let Λ be a set of possibilistic literals, α be a propositional atom and $p \in (0, 1]$,

- $F_d(\Lambda, \alpha) = \{(l, p) | (l, p) \in \Lambda$ and $c(l) \equiv \alpha\}$; F_d denotes all the possibilistic literals that are the topic α from the set Λ.
- $F_a(\Lambda, \alpha) = \{(l, p) | (l, p) \in \Lambda$ and $c(l) \equiv \neg\alpha\}$; F_a denotes all the possibilistic literals that are $\neg\alpha$ from Λ.

- $GLB(\Lambda) = max\{p | (l, p) \in \Lambda\}$; GLB denotes the greatest lower bound from a nonempty set Λ.
- $F_p(\Lambda, p) = \{(l, p) | (l, p) \in \Lambda\}$; F_p denotes all the possibilistic literals with a particular p from Λ.
- $Cardinality(\Lambda) = \{i | i$ is the quantity of the possibilistic literals in $\Lambda\}$; $Cardinality$ retrieves the number of the possibilistic literals in Λ.

The main idea of Algorithm 1 is as follows: (1) classify beliefs from the union of two CSs into two sets, which are Λ_d and Λ_a; (2) make a copy of the two sets to show all the arguments in the set to the user when this set wins in the end; (3) retrieve the GLB from each set and compare these two numbers; (4) the set with larger GLB wins; (5) if they have the same GLB, the algorithm will count the quantity of beliefs whose possibilistic value equals the GLB in each set; (6) if they are equal, these beliefs will be removed from each set (this operation is usually called *cut* in possibilistic logic), and two new sets are retrieved; (7) the new sets are compared until one set wins or both become empty.

Algorithm 1 Get $Outcome_{wi}(D_r^t)$

```
Require: a well-formed wi dialogue $D_r^t$ with $\alpha$ as its topic
Ensure: $\langle r, p \rangle$
1: Set $\Lambda_{d1} = \Lambda_d = F_d(CS_x^t \cup CS_{\hat{x}}^t, \alpha)$ and $\Lambda_{a1} = \Lambda_a = F_a(CS_x^t \cup CS_{\hat{x}}^t, \alpha)$,
2: loop
3:     If $\Lambda_d = \emptyset$ and $\Lambda_a = \emptyset$, return $\langle U, null \rangle$.
4:     If $\Lambda_d \neq \emptyset$ and $\Lambda_a = \emptyset$, return $\langle T, GLB(\Lambda_d) \rangle$
5:     If $\Lambda_d = \emptyset$ and $\Lambda_a \neq \emptyset$, return $\langle F, GLB(\Lambda_a) \rangle$
6:     If $GLB(\Lambda_d) > GLB(\Lambda_a)$, return $\langle T, GLB(\Lambda_d) \rangle$.
7:     If $GLB(\Lambda_d) < GLB(\Lambda_a)$, return $\langle F, GLB(\Lambda_a) \rangle$.
8:     Set $\Lambda_{dt} = F_p(\Lambda_d, GLB(\Lambda_d))$ and $\Lambda_{at} = F_p(\Lambda_a, GLB(\Lambda_a))$,
9:     If $Cardinality(\Lambda_{dt}) > Cardinality(\Lambda_{at})$, return $\langle T, GLB(\Lambda_d) \rangle$.
10:    If $Cardinality(\Lambda_{dt}) < Cardinality(\Lambda_{at})$, return $\langle F, GLB(\Lambda_a) \rangle$.
11:    Set $\Lambda_d = \Lambda_d - \Lambda_{dt}$ and $\Lambda_a = \Lambda_a - \Lambda_{at}$.
12: end loop
```

The main idea of Algorithm 2 is similar to Algorithm 1. However, it omits the *cut* and loop parts (i.e. step 2, 11, and 12 in Algorithm 1 are omitted) so that only the arguments with the highest weight in both sets are compared. If it cannot make a comparison from these arguments, $\langle U, null \rangle$ will be returned. The motivation of Algorithm 2 is grounded in the medical domain where clinicians typically evaluate the strongest evidence that supports a conclusion.

Algorithm 2 Get $Outcome_{wi}(D_r^t)$

```
Require: a well-formed wi dialogue $D_r^t$ with $\alpha$ as its topic
Ensure: $\langle r, p \rangle$
Set $\Lambda_d = F_d(CS_x^t \cup CS_{\hat{x}}^t, \alpha)$ and $\Lambda_a = F_a(CS_x^t \cup CS_{\hat{x}}^t, \alpha)$,
If $\Lambda_d = \emptyset$ and $\Lambda_a = \emptyset$, return $\langle U, null \rangle$.
If $\Lambda_d \neq \emptyset$ and $\Lambda_a = \emptyset$, return $\langle T, GLB(\Lambda_d) \rangle$
If $\Lambda_d = \emptyset$ and $\Lambda_a \neq \emptyset$, return $\langle F, GLB(\Lambda_a) \rangle$
If $GLB(\Lambda_d) > GLB(\Lambda_a)$, return $\langle T, GLB(\Lambda_d) \rangle$.
If $GLB(\Lambda_d) < GLB(\Lambda_a)$, return $\langle F, GLB(\Lambda_a) \rangle$.
Set $\Lambda_{dt} = F_p(\Lambda_d, GLB(\Lambda_d))$ and $\Lambda_{at} = F_p(\Lambda_a, GLB(\Lambda_a))$,
If $Cardinality(\Lambda_{dt}) > Cardinality(\Lambda_{at})$, return $\langle T, GLB(\Lambda_d) \rangle$.
If $Cardinality(\Lambda_{dt}) < Cardinality(\Lambda_{at})$, return $\langle F, GLB(\Lambda_a) \rangle$,
Else return $\langle U, null \rangle$.
```

It is common that the same atom or its negation can be used in different rules as a premise. If the premise has already been proved before (i.e. a wi dialogue with this premise as topic has already been terminated), the system should not prove it twice. We use *Result Store* (*RS*) to store the intermediate result in order to avoid repetitive work.

Definition 11 A **RS** is a set of tuples of the form $\left\langle \alpha, Outcome_{wi}(D_r^t)\right\rangle$ where D_r^t is a well-formed wi dialogue, and α is its topic.

When a wi dialogue opens, the RS is added with a new pair $\langle \alpha, \langle U, null\rangle\rangle$ where α is the topic of the dialogue. When it closes, $\langle U, null\rangle$ part is updated with $Outcome_{wi}(D_r^t)$.

$$RS^t = \begin{cases} RS^{t-1} \cup \{\langle \alpha, \langle U, null\rangle\rangle\}, & \text{iff } m_t = \langle x, open, wi, \alpha\rangle, \\ RS^{t-1} \cup \{\langle \alpha, \langle r, p\rangle\rangle\}, & \text{iff } Topic(D_r^t) = \alpha \text{ and } Outcome_{wi}(D_r^t) = \langle r, p\rangle, \\ RS^{t-1}, & \text{Otherwise.} \end{cases}$$

The outcome of the wi dialogue with topic α is a tuple $\langle r, p\rangle$. r and p are returned by the functions $Result(\alpha)$ and $PL(\alpha)$, respectively, which are defined as follows:

$$Result(\alpha)^t = \begin{cases} r, & \text{iff } \langle \alpha, \langle r, p\rangle\rangle \in RS^t, \\ null, & \text{Otherwise.} \end{cases}$$

$$PL(\alpha)^t = \begin{cases} p, & \text{iff } \langle \alpha, \langle r, p\rangle\rangle \in RS^t, \\ null, & \text{Otherwise.} \end{cases}$$

When we get Result $(\alpha)^t$, $Result(\neg\alpha)^t$ can also be inferred as follows:

$$Result(\neg\alpha)^t = \begin{cases} T, & \text{if } Result(\alpha)^t = F, \\ F, & \text{if } Result(\alpha)^t = T, \\ U, & \text{if } Result(\alpha)^t = U, \\ null, & \text{Otherwise.} \end{cases}$$

If the wi dialogue with topic α has been opened, $Result(\alpha)^t$ and $PL(\alpha)^t$ should be U and $null$, respectively. This, however, is different from the case when the dialogue has not been opened at all. In the latter case, $Result(\alpha)^t$ and $PL(\alpha)^t$ should both return $null$. In fact, when the wi dialogue opens, $\langle \alpha, \langle U, null\rangle\rangle$ is added to the RS as an indicator so we can distinguish if the dialogue has opened or not.

When $Result(\alpha)^t$ is U, it can be further divided into two cases. The first is more common, where the dialogue has already been terminated before time t and the result is U (see step 3 in Algorithm 1). The second is special, where the dialogue remains open until t. Here, in case an ai dialogue with topic ($\alpha \rightarrow \beta$) exists inside the wi dialogue with topic α, then an endless loop may occur if the agent now opens a wi dialogue with topic α inside this ai dialogue. To address this problem, we develop a *fast end strategy* so that the agent makes an assert ai move for both cases.

Definition 12 The **fast end strategy** for an agent x participating in an ai dialogue is that the ai dialogue terminates directly if one of the premises of the ai dialogue's topic is a wi dialogue's topic which has not been closed yet. Specifically, $Result(\alpha)$ returns U when the wi dialogue with topic α is not terminated. Then, the agents will do a *close ai* move in turn in the following two steps and, the QS regarding this ai dialogue is reset to $\emptyset$.

In agent dialogue frameworks, protocols are usually presented to determine the agents' next moves [2,15,26,36]. In this paper, we regard a protocol as a function that, given a particular type of dialogue, a specific move is returned according to its belief base and the previous moves they have already made. This is somewhat different with respect to the approaches described in [2,26], where their protocols only return a set of legal moves. To select exactly one move, they need additional computation. However, using our protocols, exactly one move can be selected at each time point.

If the current dialogue is a wi dialogue, the wi dialogue protocol (presented in Table 3) should be followed by each agent to determine its next move. When Agent x opens a wi dialogue at time r (i.e. $m_r = \langle x, open, wi, \alpha\rangle$), Agent x updates its PBQ according to Definition 6. At the next time (i.e. $r + 1$), $\hat{x}$ updates its PBQ. Then, within this wi dialogue, an agent makes its next move according to its PBQ. If PBQ is empty, the agent makes a *close wi* move (see step 3.2 in Table 3); else if the first element in PBQ is a state belief, the agent asserts this belief (see step 3.3.1 in Table 3); else the agent opens an ai dialogue since it is a domain belief (see step 3.3.2 in Table 3).

Similar to the wi dialogue, if the current dialogue is an ai dialogue, the ai dialogue protocol (presented in Table 4) should be followed by each agent. When Agent x opens an ai dialogue at time r (i.e. $m_r = \langle x, open, ai, (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \alpha, p)\rangle$), the QS is updated according to Definition 7. Within this ai dialogue, the agents make the next move according to the QS and the RS. If QS is empty, the agent makes a *close ai* move (see step 3.2 in Table 4); else the next move is determined by the RS. If the premise α_k is not in the QS (i.e. the wi dialogue with topic α_k has never been opened yet), the agent makes an *open wi* dialogue (see step 3.3.3 in Table 4); else, it makes an *assert ai* move. However, there are two cases about the *assert ai* move. If $Result(\alpha_k) = T$, $\langle x, assert, (\rightarrow \alpha_k, p')$ is performed (see step 3.3.1 in Table 4) and the ai dialogue can continue; else $\langle x, assert, (\rightarrow \neg\alpha_k, p')$ is performed (see step 3.3.2 in Table 4) and the ai dialogue will be closed in the following two steps since the agents cannot prove the current premise.

Example 3 Continuing the running example, the whole dialogue generated by PA and DA are shown in Fig. 1.

In this example, eight dialogues are generated. Inside the top wi dialogue D_1^{31}, there are three ai dialogues: D_2^{12}, D_{13}^{20}, and D_{22}^{29}. Inside each ai dialogue, some wi dialogues are also generated. To determine the outcome of the outer layer dialogue, the inner layer should be determined first.

To diagnose whether the patient has f or not, PA opens the wi dialogue with topic f. Meanwhile, $\langle f, \langle U, null\rangle\rangle$ is added to the RS. Then, PA and DA both update their PBQs. DA has two domain beliefs in its PBQ and opens an ai dialogue with the first one as its topic. The premises of the topic is added to the QS. Next, PA opens another wi dialogue with the first literal a in the QS as its topic. Meanwhile, $\langle a, \langle U, null\rangle\rangle$ is added to RS. DA has no beliefs regarding a. Therefore, it tries to close the dialogue. However, since PA has a state belief, it asserts $(\rightarrow a, 1)$, and the dialogue is still open. $(a, 1)$ is added to CS_{PA}. When the wi dialogue successfully terminates at $t = 7$, its outcome is calculated according to Algorithm 1 or 2. Since there is only one argument $\langle((\rightarrow a, 1)), (a, 1)\rangle$ supporting the topic a, we have $Outcome_{wi}(D_3^7) = \langle T, 1\rangle$. Meanwhile, $\langle a, \langle T, 1\rangle\rangle$ is updated to RS.

$Outcome_{wi}(D_8^{10}) = \langle U, null\rangle$ is obtained since no beliefs related to b are found in both belief bases. Therefore, $Outcome_{ai}(D_2^{12}) = null$ since one of the topic's premises cannot be proven to be true.

At $t = 13$, PA opens the ai dialogue with topic $(a \wedge c \rightarrow f, p_2)$. In the next step, DA makes the assert ai move $\langle DA, assert, ai, (\rightarrow a, 1)\rangle$ instead of opening a wi dialogue with a because the wi dialogue with a as the topic has already opened before and $Result^{13}(a) = T$. Similarly, $(c, 1)$ is added to CS_{DA} at $t = 16$. $Outcome_{wi}(D_{15}^{18}) = \langle T, 1\rangle$ and $Outcome_{ai}(D_{13}^{20}) = (f, p_2)$ are reached, and (f, p_2) is added to CS_{DA}. The argument $\langle((\rightarrow a, 1), (\rightarrow c, 1), (a \wedge c \rightarrow f, p_2)), (f, p_2)\rangle$ is built.

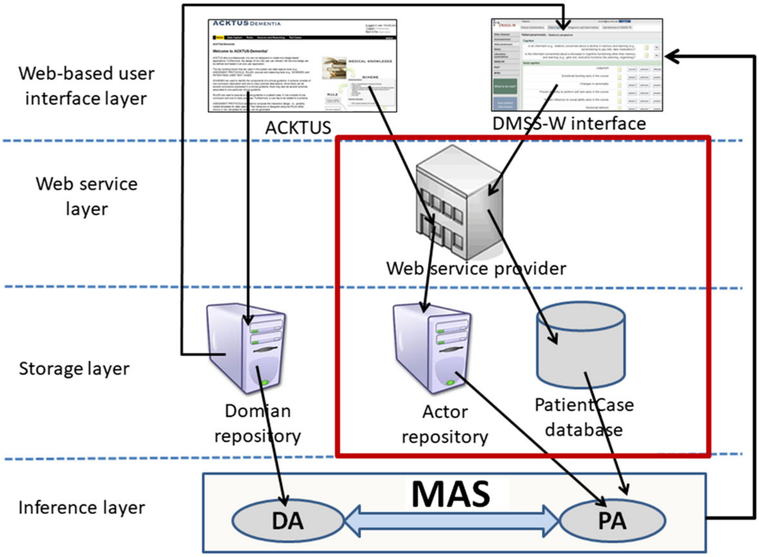


Fig. 2 Architecture of the whole system. The modules within the red frame are physically located on the client side

$Outcome_{wi}(D^{27}_{23}) = \langle T, 1\rangle$ and $Outcome_{ai}(D^{29}_{22}) = (\neg f, p_1)$ are achieved and (f, p_1) is added to CS_{PA}. The argument $\langle((\rightarrow d, 1), (d \rightarrow \neg f, p_1)), (\neg f, p_1)\rangle$ is built. This argument conflicts with the previous argument and $p_1 > p_2$, thus $Outcome_{wi}(D^{31}_{1}) = \langle F, p_1\rangle$ is reached and $\langle f, \langle F, p_1\rangle\rangle$ is updated to the RS.

4 Implementation

The argumentation-based approach presented in Sects. 2 and 3 has been implemented and integrated into the application *DMSS-W* (Dementia Diagnosis and Management Support System—Web version) [22], which is a clinical decision-support system for diagnosing dementia diseases. DMSS-W utilizes data from the ACKTUS platform (Activity-Centered Modeling of Knowledge and Interaction Tailored to Users) [23,24], which is a web-based tool for modeling medical knowledge into rules and claims in natural language. We used *Java Agent Development Framework* (*JADE*) to develop the MAS and *Java* language platform to integrate it into DMSS-W as one of its inference engines. In the MAS, DA and PA are defined as two agents.

In this section, we will explain how the whole system works, where the knowledge comes from, and how it is mapped as facts and rules and used by DA and PA. First, the architecture of the whole system will be given.

4.1 Architecture

The whole system can be divided into four layers: *interface layer*, *web service layer*, *storage layer*, and *inference layer* as depicted in Fig. 2. It includes two systems: ACKTUS and DMSS-W; one web service provider; two ontology repositories: *domain repository* and *actor*

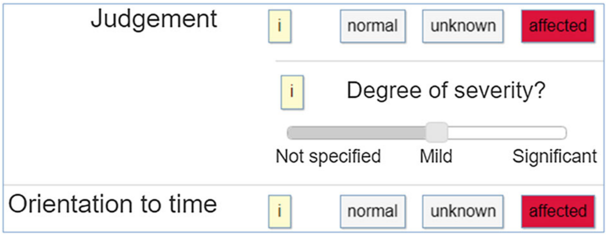


Fig. 3 Example of IOs and the scales in the interface

repository; one kind of relational database *patientCase*; and two kinds of agents: *DA* and *PA*. There are two kinds of users: *domain experts* and *the regular professionals* in this architecture. We will detail each of them in the following paragraphs.

Domain experts in a medical domain model the evidence-based domain knowledge into formal machine-interpretable representations using ACKTUS. The knowledge is stored in the *domain repository*, which is used by the DA. The *domain repository* is located on the same server as ACKTUS and DMSS-W. Consequently, DA has access to a large set of rules based upon clinical guidelines, which are consensus documents involving a large international community of domain experts [12,43].

A regular professional can also formulate his or her knowledge using the same structures; however, since the knowledge is not official and evidence-based, it is stored in the *actor repository*, which is used by PA. PA can also retrieve factual data (state beliefs, symptoms observed in the patient) from the *PatientCaseDB*. These factual data are inputted by the user through DMSS-W user interface, transferred via the web service, and stored in the *PatientCaseDB*.

PA initiates a dialogue, and DA responds. Subsequently, they generate a dialogue with nested sub-dialogues using the approach presented in Sect. 3. When the top dialogue terminates, a result about a hypothesis is reached. The dialogue with all the moves and the result are then presented in the DMSS-W user interface to the end user (see Fig. 5). The diagnostic result is achieved.

4.2 Construction and design of knowledge bases

Below we show how to interpret the knowledge as beliefs and use them in the reasoning process.

4.2.1 State beliefs

The domain experts model *interaction objects* (IOs) through ACKTUS and store them in the *domain repository*. The *domain repository* is used for elaborated knowledge such as symptom manifestations, syndromes and diseases, and the evaluated observations obtained from laboratory examinations. Each IO contains one or more *scales* with different scale values. Figure 3 is a screenshot of DMSS-W showing two IOs (*Judgement* and *Orientation to time*) with their scale values. The end user answers the questions posted in the user interface by clicking the scale values. IOs with inputted scale values are used as *state beliefs* in the reasoning on a topic.

The scale can be simple with two or three possible values. For this type, if the scale value is a *certain* knowledge (e.g., normal, affected), we use an atom, such as α, to represent *affected* and its opposite $\neg\alpha$ to represent *normal*; also, we assign the possibilistic value as 1. If the knowledge is *uncertain*, we map this data to two opposite possibilistic beliefs with a necessity value less than 0.5 (the assignment of it is the user's task). We give a concrete example to demonstrate our method. The IO *Orientation to time* in Fig. 3 has a simple scale type with three values (normal/unknown/affected). We use OOT to represent "Orientation to time-affected" and $\neg OOT$ to represent "Orientation to time-normal". Therefore from the interface,

1. If "Orientation to time-normal" is clicked, the state belief $(\rightarrow \neg OOT, 1)$ is generated;
2. If "Orientation to time-unknown" is clicked, both $(\rightarrow OOT, p)$ and $(\rightarrow \neg OOT, p)$ are generated, such that $(0 < p < 0.5)$;
3. If "Orientation to time-affected" is clicked, $(\rightarrow OOT, 1)$ is generated.

The presence of a symptom can, in some cases, lead to the possibility to choose an additional scale value, such as the severity level. Such successor values are treated as follows. We also use a concrete example to show our method. The IO *Judgement* in Fig. 3 contains a reliability scale [*normal*, *unknown*, *affected*], and the scale value "affected" triggers the severity scale [*not specified*, *mild*, *Significant*]. In this example, the atom $Judge$ is used for representing "Judgement-affected" and $\neg Judge$ as "Judgements-normal". Therefore:

1. If "Judgement-normal" is clicked, $(\rightarrow \neg Judge, 1)$ is generated;
2. If "Judgement-unknown" is clicked, both $(\rightarrow Judge, p)$ and $(\rightarrow \neg Judge, p)$ are generated, such that $(0 < p < 0.5)$;
3. If "Judgement-affected" is clicked,
   (a) If "not specified" is chosen, both $(\rightarrow Judge, 1)$ and $(\rightarrow Judge\_n, p)$ are generated, such that $(0 < p < 0.5)$;
   (b) If "mild" is chosen, both $(\rightarrow Judge, 1)$ and $(\rightarrow Judge\_m, 1)$ are generated;
   (c) If "significant" is chosen, both $(\rightarrow Judge, 1)$ and $(\rightarrow Judge\_s, 1)$ are generated.

Except these, there are additional domain beliefs that are applied during the inference procedure.

1. If we know "affected-not specified" is true, we can infer the following: $(Judge\_n \rightarrow Judge, 1)$, $(Judge\_n \rightarrow \neg Judge\_m, p)$ and $(Judge\_n \rightarrow \neg Judge\_s, p)$, such that $(0 < p < 0.5)$;
2. If we know "affected-mild" is true, we can infer the following rules: $(Judge\_m \rightarrow Judge, 1)$, $(Judge\_m \rightarrow \neg Judge\_s, 1)$ and $(Judge\_m \rightarrow \neg Judge\_n, 1)$;
3. If we know "affected-significant" is true, we can infer $(Judge\_s \rightarrow Judge, 1)$, $(Judge\_s \rightarrow \neg Judge\_n, 1)$ and $(Judge\_s \rightarrow \neg Judge\_m, 1)$;
4. If we know "normal" is true, we can infer: $(\neg Judge \rightarrow \neg Judge\_n, 1)$, $(\neg Judge \rightarrow \neg Judge\_m, 1)$ and $(\neg Judge \rightarrow \neg Judge\_s, 1)$.

4.2.2 Domain beliefs

An IO associated with one of its scale values composes a premise or a conclusion. Based on the premises and conclusion, a rule can be created. The domain expert who models the knowledge in ACKTUS selects a scale for a certain conclusion that mirrors the underlying medical guideline, and this scale is used for extracting the rule's possibilistic value. Typically, this scale contains the values *present*, *absent* or *unknown*, where *present* represents pathology, and *absent* represents a normal condition. The values of a scale can also represent a broader range of possibilities, such as [*excluded*, *unlikely*, *possible*−, *possible*, *possible*+, *probable*−, *probable*, *probable*+]. The rules are assigned with their possibilistic values based on the scale and values the domain expert applies to symptoms and diseases. In this way, we can map the knowledge base into the logic framework and use the logic-based method to conduct a reasoning process.

The following is an example of the construction of a domain belief. The knowledge base contains a structure with the premise "Episodic memory is normal" and the conclusion "Alzheimer's disease is excluded". This information can be formalised as a domain belief $(\neg\alpha \rightarrow \neg\beta, excluded)$, where $\neg\alpha$ is the premise, $\neg\beta$ is the conclusion and $excluded$ is the possibilistic value of the belief.

4.2.3 Arguments

An *argument* is constructed based on a set of beliefs to deduct a conclusion. Its p is calculated based on the beliefs forming the argument. We continue the previous example. If *Episodic memory is normal* is proved to be true, an argument can be formalized as $\langle((\rightarrow \neg\alpha, 1),\ (\neg\alpha \rightarrow \neg\beta, excluded)),\ (\neg\beta, excluded)\rangle$.

Since the data in the knowledge bases can be uncertain and inconsistent, conflicting arguments could be deducted. Our strategy to solve the conflict is based on Algorithm 1 or 2, depending on the user's preference.

5 Dealing with uncertain and inconsistent data via an inquiry dialogue

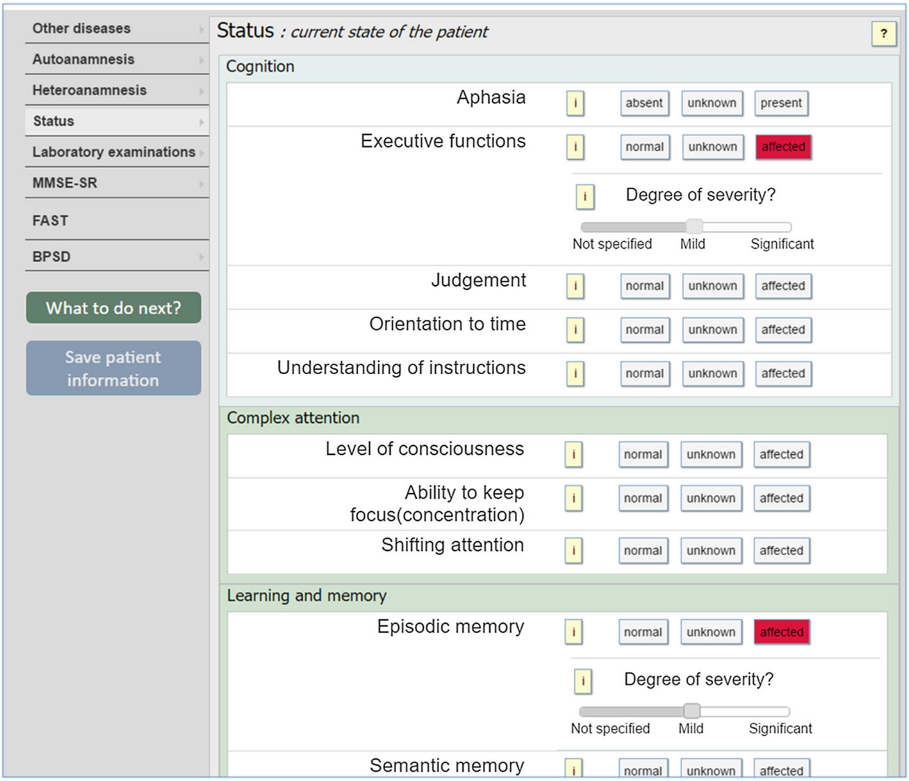
In this section, we show one scenario of how the MAS is integrated into the DMSS-W application and demonstrate how it works.

A physician diagnoses a potential dementia patient. The physician uses different methods (e.g., observation, asking the patient or the relatives, laboratory examination) to collect information about the patient's condition and inputs the information into DMSS-W. A part of the system interface is shown in Fig. 4. The symptoms used in this scenario are the following:

- *Status shows mild executive dysfunction (the red button in the upper right corner in Fig.* 4). Based on this, information $(\rightarrow a, 1)$ and $(\rightarrow a\_m, 1)$ are retrieved.
- *Status shows mild episodic memory dysfunction (the red button in the lower right corner in Fig.* 4). Based on this, $(\rightarrow b, 1)$ and $(\rightarrow b\_m, 1)$ are retrieved.
- *Heteroanamnesis shows mild memory dysfunction.* Based on this, $(\rightarrow c, 1)$ and $(\rightarrow c\_m, 1)$ are retrieved.
- *Heteroanamnesis shows mild executive dysfunction.* Based on this, $(\rightarrow d, 1)$ and $(\rightarrow d\_m, 1)$ are retrieved.
- *Proven no disability to perform self care.* Based on this, $(\rightarrow \neg e, 1)$ is retrieved.

The symptoms are stored in the PatientCaseDB and used as state beliefs by the PA in the reasoning process. The hypothesis *mild cognitive impairment (MCI) is present* is the topic that the physician selects for the dialogue between PA and DA.

The DA, representing an expert, has a large set of domain beliefs in its belief base, which are related to the topic of the dialogue and retrieved from the *domain repository*. It should be noted that these domain beliefs are retrieved from different, sometimes conflicting sources (in this case two different guidelines for diagnosis). A subset of domain beliefs are applied in our examples for readability purposes. We apply five domain beliefs that are associated to



Status : *current state of the patient*

Cognition

Aphasia: absent | unknown | present

Executive functions: normal | unknown | [✓ affected]

Degree of severity? Not specified | [✓ Mild] | Significant

Judgement: normal | unknown | affected

Orientation to time: normal | unknown | affected

Understanding of instructions: normal | unknown | affected

Complex attention

Level of consciousness: normal | unknown | affected

Ability to keep focus(concentration): normal | unknown | affected

Shifting attention: normal | unknown | affected

Learning and memory

Episodic memory: normal | unknown | [✓ affected]

Degree of severity? Not specified | [✓ Mild] | Significant

Semantic memory: normal | unknown | affected

Fig. 4 Interface of DMSS-W

two different conflicting diagnostic criteria for MCI. These domain beliefs have a conclusion that leads to a hypothesis about *MCI*, as listed below. The DA has no state beliefs about the patient in its belief base since it does not have the management responsibility for patients like the PA has. Although, in our practice, DA only has domain beliefs; we note that the theory presented in Sect. 3 does not have such a restriction on the agents.

All of the first three domain beliefs' conclusions are *MCI is present* which support the hypothesis. Both of the last two domain beliefs' conclusions are *MCI is absent* which attack the hypothesis. Some domain beliefs in this scenario are listed here.

- DBelief1: $(\neg f \wedge g \wedge \neg e \wedge \neg a \rightarrow MCI, 1)$, where the premises are:
  - *No state of dementia.* $(\neg f)$
  - *Cognitive symptoms are present.* (g)
  - *Proven no disability to perform self care.* $(\neg e)$
  - *Status shows no executive dysfunction.* $(\neg a)$
- DBelief2: $(b\_m \wedge \neg a \rightarrow MCI, 1)$, where the premises are:
  - *Status shows mild episodic memory dysfunction.* (b_m)
  - *Status shows no executive dysfunction.* $(\neg a)$
- DBelief3: $(\neg f \wedge g \wedge a\_m \wedge \neg e \rightarrow MCI, 1)$, where the premises are:

  - *No state of dementia.* ($\neg f$)
  - *Cognitive symptoms are present.* (g)
  - *Status shows mild executive dysfunction.* (a_m)
  - *Proven no disability to perform self care.* ($\neg e$)
- DBelief4: ($a\_m \rightarrow \neg MCI, 1$), where the premise is:
  - *Status shows mild executive dysfunction.* (a_m)
- DBelief5: ($a\_s \rightarrow \neg MCI, 1$), where the premise is:
  - *Status shows significant executive dysfunction.* (a_s)

The procedure of the dialogue is as follows. PA initiates the dialogue. It makes the move $\langle PA, open, wi, MCI\rangle$. Meanwhile, it checks its belief base and finds that there is no belief related to the topic. Then, DA checks its belief base and finds five domain beliefs relevant to *MCI*. It stores them to PBQ^2_{DA}. Next, it retrieves and deletes (polls) *DBelief1* from PBQ^2_{DA} and makes the move $\langle DA, open, ai, (\neg f \wedge g \wedge \neg e \wedge \neg a \rightarrow MCI, 1)\rangle$. At the same time, DA adds all the premises of *DBelief1* to the QS. Next, it is PA's turn again. It polls the first premise *no state of dementia* ($\neg f$) from QS and then checks the RS. Since RS is now empty, PA makes the move $\langle PA, open, wi, \neg f\rangle$. Meanwhile, it updates its PBQ according to its belief base. In this example, $(\neg f, 1)$ is proved.

When the wi dialogue with the topic $\neg f$ closes, the result $\langle \neg f, \langle T, 1\rangle\rangle$ is reached and stored in RS. Later, when needed, two agents do not need to check this literal again; instead, they just retrieve it from RS and reuse it. For example, *DBelief3* also has premise $\neg f$. When DA makes the move $\langle DA, open, ai, (\neg f \wedge g \wedge a\_m \wedge \neg e \rightarrow MCI, 1)\rangle$ (move 132 in Fig. 5), the next move for PA to make is $\langle PA, assert, ai, (\rightarrow \neg f, 1)\rangle$, claiming that f is *false* instead of making an *open wi* move.

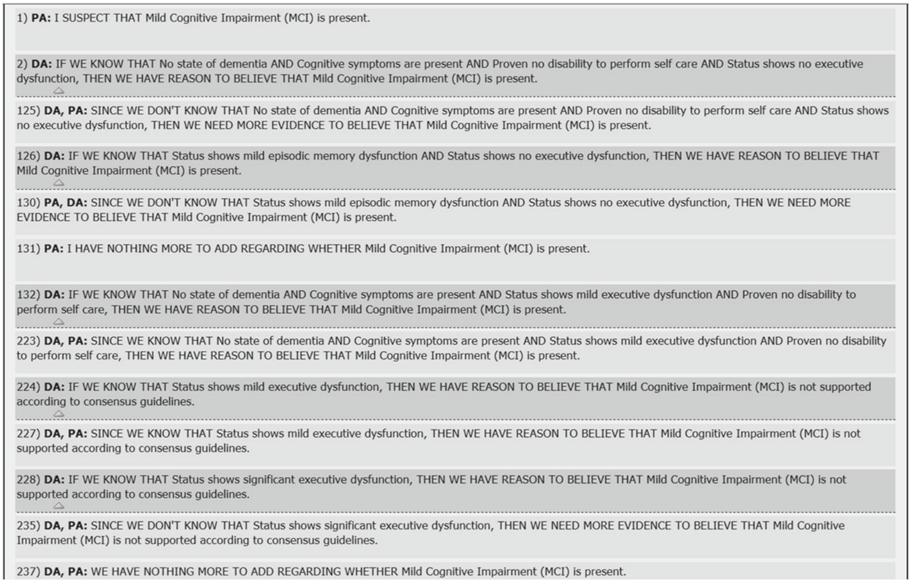
Then, two agents open another wi dialogue with the topic *cognitive symptoms are present* (g). In this example, it can also be proved to be true. The third premise of $DBelief1$ *proven no disability to perform self care* ($\neg e$) can be proved also since PA has a symptom (the last state belief of PA) to show this, and there is no other beliefs to attack it in this example.

Two agents then open the fourth wi dialogue with the last premise *Status shown no executive dysfunction* ($\neg a$) as the topic. This premise is proved to be *false* since its opposite *Status shown mild executive dysfunction* is proved to be *true* (a and a_m are true); therefore, the wi dialogue's outcome is $\langle F, null\rangle$. Until now, the ai dialogue with topic $DBelief1$ is closed and its outcome is *null* since $\neg a$ is $false$.

From previous explanation, we can see four wi dialogues have been generated and nested within the ai dialogue based on *DBelief1* since it has four premises. However if the order of its premises is changed, the amount of wi dialogues can be different. For example, if the last premise $\neg a$ is the first one to be proven, there will be only one wi dialogue (with this premise as its topic). The reason is its result is *null*, so the agents need not check other premises.

Since DA has four more domain beliefs related to *MCI*, it will open the other ai dialogues in a sequence. When all five domain beliefs are checked, the wi dialogue with *MCI* as its topic is closed. A total of 237 moves are made by the two agents, which can be seen in the screenshot in Fig. 5. The timepoints in Fig. 5 are not consecutive because the sub-dialogue are hidden, which can be expanded by clicking the small triangles.

In this scenario, there are two conflicting arguments generated due to the two conflicting guidelines, which can be seen in Fig. 6. In this case, the outcome of the dialogue is $\langle U, null\rangle$ since the two arguments are equal in weight.



1) **PA:** I SUSPECT THAT Mild Cognitive Impairment (MCI) is present.

2) **DA:** IF WE KNOW THAT No state of dementia AND Cognitive symptoms are present AND Proven no disability to perform self care AND Status shows no executive dysfunction, THEN WE HAVE REASON TO BELIEVE THAT Mild Cognitive Impairment (MCI) is present.

125) **DA, PA:** SINCE WE DON'T KNOW THAT No state of dementia AND Cognitive symptoms are present AND Proven no disability to perform self care AND Status shows no executive dysfunction, THEN WE NEED MORE EVIDENCE TO BELIEVE THAT Mild Cognitive Impairment (MCI) is present.

126) **DA:** IF WE KNOW THAT Status shows mild episodic memory dysfunction AND Status shows no executive dysfunction, THEN WE HAVE REASON TO BELIEVE THAT Mild Cognitive Impairment (MCI) is present.

130) **PA, DA:** SINCE WE DON'T KNOW THAT Status shows mild episodic memory dysfunction AND Status shows no executive dysfunction, THEN WE NEED MORE EVIDENCE TO BELIEVE THAT Mild Cognitive Impairment (MCI) is present.

131) **PA:** I HAVE NOTHING MORE TO ADD REGARDING WHETHER Mild Cognitive Impairment (MCI) is present.

132) **DA:** IF WE KNOW THAT No state of dementia AND Cognitive symptoms are present AND Status shows mild executive dysfunction AND Proven no disability to perform self care, THEN WE HAVE REASON TO BELIEVE THAT Mild Cognitive Impairment (MCI) is present.

223) **DA, PA:** SINCE WE KNOW THAT No state of dementia AND Cognitive symptoms are present AND Status shows mild executive dysfunction AND Proven no disability to perform self care, THEN WE HAVE REASON TO BELIEVE THAT Mild Cognitive Impairment (MCI) is present.

224) **DA:** IF WE KNOW THAT Status shows mild executive dysfunction, THEN WE HAVE REASON TO BELIEVE THAT Mild Cognitive Impairment (MCI) is not supported according to consensus guidelines.

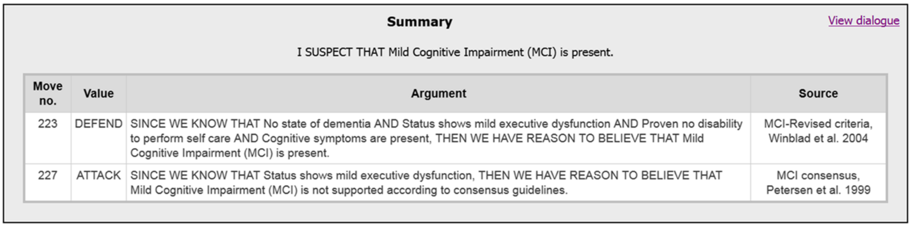
227) **DA, PA:** SINCE WE KNOW THAT Status shows mild executive dysfunction, THEN WE HAVE REASON TO BELIEVE THAT Mild Cognitive Impairment (MCI) is not supported according to consensus guidelines.

228) **DA:** IF WE KNOW THAT Status shows significant executive dysfunction, THEN WE HAVE REASON TO BELIEVE THAT Mild Cognitive Impairment (MCI) is not supported according to consensus guidelines.

235) **DA, PA:** SINCE WE DON'T KNOW THAT Status shows significant executive dysfunction, THEN WE NEED MORE EVIDENCE TO BELIEVE THAT Mild Cognitive Impairment (MCI) is not supported according to consensus guidelines.

237) **DA, PA:** WE HAVE NOTHING MORE TO ADD REGARDING WHETHER Mild Cognitive Impairment (MCI) is present.

Fig. 5 Example of moves in the dialogue about the presence of MCI



Summary

View dialogue

I SUSPECT THAT Mild Cognitive Impairment (MCI) is present.

Move no.	Value	Argument	Source
223	DEFEND	SINCE WE KNOW THAT No state of dementia AND Status shows mild executive dysfunction AND Proven no disability to perform self care AND Cognitive symptoms are present, THEN WE HAVE REASON TO BELIEVE THAT Mild Cognitive Impairment (MCI) is present.	MCI-Revised criteria, Winblad et al. 2004
227	ATTACK	SINCE WE KNOW THAT Status shows mild executive dysfunction, THEN WE HAVE REASON TO BELIEVE THAT Mild Cognitive Impairment (MCI) is not supported according to consensus guidelines.	MCI consensus, Petersen et al. 1999

Fig. 6 Conflicting arguments generated by the two agents

6 Related work

MASs are widely studied in a variety of fields such as task allocation [16], online trading [37], disaster response [9], and modeling social structures [39]. There are some works related to MASs in the healthcare domain, such as in medical data management [41], patient scheduling [25], and remote care (e.g., for senior citizens) [5,10,17,40]. There are also some works on decision-support applications in MASs to help physicians diagnose patients [11,38]. However, there are few works on how agents communicate and make decisions in collaboration in a transparent manner for the purpose of educating medical professionals.

The authors in [11] present a decision-support system for diagnosing brain tumors and predicting the progress. In their system, there are several medical centres where each centre has several agents (e.g., a classifier agent, database agent, preprocessing agent) contributing to different roles. The classifier agent aims to provide tumor classifications based on its case data to support the decision-making process when receiving a request to diagnose a new patient from another centre. However, this classification process is only based on its local database, which means a classifier agent does not interact with other classifier agents to make the decision. As the authors mentioned in [11], the accuracy of the classifiers depends heavily on the volume of cases. Therefore, the result is less accurate when it is only based on the case data in its local database than when it uses the data in all the classifier agents. In our system, the agents make a decision based on the combination of their belief bases, and they provide transparent reasons for the decision which are comprehensible to the clinician.

The authors in [38] present the architecture of a healthcare intelligent assistant using various existing organizational knowledge to solve medical cases. They use grid technology and do not need to consider the underlying details of each node, such as resource management, security, and others. In [18], a generic computational model is proposed to implement the development of interoperable intelligent software agents for medical applications. None of these approaches provide details about how agents interact to collaboratively make a decision.

Although the approaches presented in [11,38] are related to decision-support systems using case-base reasoning in MASs, the inference procedure is still within one agent. The reasoner agent receives a new case from the sender agent, infers with its database to match the received case and then forwards the result to the sender or a third agent. This approach is more like standard programs that receive the input and return the output. From a software development perspective, communication between agents in these systems are simpler than in our system. In our system, neither of the agents can execute the reasoning process alone since a single agent does not have enough knowledge to do so. The data needed for reasoning depends on all the agents. Before the dialogue, no agent has explicit requirement to get all the necessary data from another agent. Only when executing does an agent asks/asserts new knowledge related to the current topic, its local knowledge, and the retrieved knowledge from the other agent. Therefore, we cannot model this system with the standard approach using inputs and outputs.

MASs are still mostly studied in academia and rarely widely deployed in the practical medical domain [14]. Our framework solves the problem of the necessary knowledge being located in a distributed environment, which opens the possibility to provide a generic, consensus-based domain expert agent (DA) that can support physicians in reasoning about a particular patient case, regardless of the physical location, while keeping the patient anonymous to the DA to protect sensitive information. The solution is implemented for experimental purposes in an existing medical application. We anticipate it could be widely used in the near future since the hypothesis-based dialogue complements the diagnostic reasoning support that is currently implemented in the system. However, user studies need to be conducted in clinical practice among users with different levels of knowledge and skills.

Our architecture allows users to store patient data in their local databases/repositories as well as share data and make decisions together with the latest DA knowledge. DA's knowledge (mainly represented as rules in our application) is dynamically increased. Therefore, it is not a good idea to store this knowledge in the client's local repository to make decisions locally.

We should note that although DA has rules and no facts in our application, our inquiry dialogues presented in Sects. 2 and 3 can be used when both agents have both facts and rules, for example, when the two agents represent different complementary disciplines in a team.

From a theoretical perspective, to the best of our knowledge, there are only a few works on generating inquiry dialogues apart from [2]. Authors in [33] define *locutions* (similar as moves in our system) and *attitudes*. Locutions describe which legal moves can be made at a specific point and how to update the commitment store after the move. Attitudes control the assertion and acceptance of propositions. It also gives a precise notion of the outcome of the dialogue. Similar to the approach presented by Parsons et al. [33], McBurney and Parsons [27] define utterance rules to generate legal moves. They also define a rule for preclude-infinite regression by malevolent participants so that the same move may only be executed once by only one of the participants. However, neither of these two approaches provide a strategy for generating inquiry dialogues.

Fan and Toni propose a generic dialogue model that is not tailored to any particular dialogue type [7]. Both their approach and ours have some similarities, such as providing legal-move and outcome functions and agents can jointly construct arguments and play both roles (defend vs. attack). However their system is based on assumption-based argumentation (ABA) framework. To use it in real applications, a specific kind of dialogue and a specific knowledge representation language should be considered before turn it to ground. However, to our knowledge, it has not been implemented in a real application.

Argumentation is very useful in dealing with conflicts, as evidenced by recent research in healthcare. Chalaguine et al. [3] use argumentation to change patients' behavior. They propose four main dimensions to categorize the arguments in their domain model: ontological, functional, context, and topic types. Noor et al. [32] extract arguments from patient experience expressed on the social web to assess a particular drug. They formalise ten classification rules to sort the arguments, evaluate the arguments, and compare with user ratings of the drug. Unlike these works which obtained arguments directly from practical domains, we let the agents make dialogues to generate arguments since the beliefs are distributed in different places.

7 Conclusions and future work

In domains such as the medical domain, knowledge bases sometimes cannot be combined due to various constrictions, and the knowledge applied in reasoning and decision making is often uncertain and inconsistent. To deal with these complex medical situations, inquiry dialogues in MASs are pursued, which simulate reality well. However, this field is relatively unexplored, and the very limited number of existing pioneer research studies, (e.g., [2]) are theoretically oriented and not trivial to implement.

The presented research contributes to the topic at least from three aspects. First, the theoretical framework of an earlier approach is modified and made more practical and feasible for real applications. For example, a wi dialogue is introduced not only in the top level but also in each sublevel of reasoning. Thus, the structure is more confined, allowing the reasoning to be performed in well-defined steps. Each problem is isolated and solved one-by-one, avoiding unnecessary complexity and uncertainty. This clearer and more efficient system is also very suitable for educational purposes. Second, possibilistic logic is used for capturing uncertain information, and an argumentation framework is used for dealing with inconsistent knowledge in reasoning about a diagnosis. An approach for generating two kinds of inquiry dialogues in a MAS is presented. Possibilistic logic is identified as a key factor for achieving the goal of managing uncertain information. Other methods for dealing with fuzzy situations (e.g., the probability-based approach) present challenges due to the lack of statistical data, which is frequently the case in the medical domain, when treating individual cases. Last and most importantly, the new methods and theories are facilitating the realization of inquiry dialogue systems. By implementing the results into DMSS-W, the decision-support system developed for the dementia domain, it is demonstrated how the uncertain and inconsistent data is properly dealt with in the practice of dementia diagnosis. One scenario is presented and discussed in detail. These kinds of real applications are particularly rare in earlier work in the field; however, it is necessary to demonstrate the strong potential of applying CDSS in real medical practice and show how uncertainty can be managed using possibilistic logic and argumentation. The study develops the solution a step further, when compared to earlier work [44], in the application of possibilistic logic and argumentation, providing a systematic and comprehensive solution that can be implemented.

In future work, a graphical display will be used instead of the current text display (e.g., Fig. 5), so that the system appears more intuitive and user friendly. This way, the user can more easily get an overview of the dialogues about the related diagnostic reasoning and decision-making. This is important for medical educational purposes. Also, the consideration of nested formulas in the conclusion of an argument [30] is an interesting topic to explore in our future work. For instance, by considering disjunctive operators in the head of a rule, one can express comorbidity, which is the presence of one or more diseases.

Finally, the MAS will be evaluated in clinical practice. The log data will be analyzed to detect reasoning patterns of the users. In this way, the educational function of the system can be verified by observing how novice users develop their reasoning and decision-making skills while using the system.

References

1. Alsinet, T., Chesnevar, C. I., Godo, L., Sandri, S., & Simari, G. (2008). Formalizing argumentative reasoning in a possibilistic logic programming setting with fuzzy unification. *International Journal of Approximate Reasoning*, *48*(3), 711–729.
2. Black, E., & Hunter, A. (2009). An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, *19*(2), 173–209.
3. Chalaguine, L., Hadoux, E., Hamilton, F., Hayward, A., Hunter, A., Polberg, S., et al. (2018). *Domain modelling in computational persuasion for behaviour change in healthcare*. arXiv:1802.10054.
4. Chesñevar, C. ., Simari, G. R., Alsinet, T., & Godo, L. (2004). A logic programming framework for possibilistic argumentation with vague knowledge. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence* (pp. 76–84). AUAI Press.
5. Cortés, U., Annicchiarico, R., Vázquez-Salceda, J., Urdiales, C., Cañamero, L., López, M., et al. (2003). Assistive technologies for the disabled and for the new generation of senior citizens: The e-tools architecture. *AI Communications*, *16*(3), 193–207.
6. Dubois, D., Lang, J., & Prade, H. (1994). Possibilistic logic. In D. M. Gabbay, C. J. Hogger, & J. A. Robinson (Eds.), *Handbook of logic in artificial intelligence and logic programming* (Vol. 3, pp. 439–513). Oxford: Oxford University Press.
7. Fan, X., & Toni, F. (2011). Assumption-based argumentation dialogues. In *IJCAI* (pp. 198–203).
8. Ferber, J. (1999). *Multi-agent systems: An introduction to distributed artificial intelligence*. Reading: Addison-Wesley.
9. Genc, Z., Heidari, F., Oey, M. A., van Splunter, S., & Brazier, F. M. (2013). Agent-based information infrastructure for disaster management. In *Intelligent systems for crisis management* (pp. 349–355). Springer.
10. Gómez-Sebastià, I., Moreno, J., Álvarez-Napagao, S., Garcia-Gasulla, D., Barrué, C., & Cortés, U. (2016). Situated agents and humans in social interaction for elderly healthcare: From coaalas to avicena. *Journal of Medical Systems*, *40*(2), 38.
11. González-Vélez, H., Mier, M., Julià-Sapé, M., Arvanitis, T. N., García-Gómez, J. M., Robles, M., et al. (2009). Healthagents: Distributed multi-agent brain tumor diagnosis and prognosis. *Applied Intelligence*, *30*(3), 191–202.
12. Grimshaw, J. M., & Russell, I. T. (1993). Effect of clinical guidelines on medical practice: A systematic review of rigorous evaluations. *The Lancet*, *342*(8883), 1317–1322.

13. Holmes, C., Cairns, N., Lantos, P., & Mann, A. (1999). Validity of current clinical criteria for Alzheimer's disease, vascular dementia and dementia with Lewy bodies. *The British Journal of Psychiatry*, *174*(1), 45–50.
14. Isern, D., Sánchez, D., & Moreno, A. (2010). Agents applied in health care: A review. *International Journal of Medical Informatics*, *79*(3), 145–166.
15. Karunatillake, N. C., Jennings, N. R., Rahwan, I., & Ramchurn, S. D. (2007). Managing social influences through argumentation-based negotiation. In *Argumentation in multi-agent systems* (pp. 107–127). Springer.
16. Kim, Y., & Matson, E. T. (2016). A realistic decision making for task allocation in heterogeneous multi-agent systems. *Procedia Computer Science*, *94*, 386–391.
17. Koutkias, V. G., Chouvarda, I., & Maglaveras, N. (2005). A multiagent system enhancing home-care health services for chronic disease management. *IEEE Transactions on Information Technology in Biomedicine*, *9*(4), 528–537.
18. Lanzola, G., Gatti, L., Falasconi, S., & Stefanelli, M. (1999). A framework for building cooperative software agents in medical applications. *Artificial Intelligence in Medicine*, *16*(3), 223–249.
19. Lindgren, H. (2007). *Knowledge management for decision support in the investigation of dementia*. Ph.D. thesis, Umeå University, Sweden.
20. Lindgren, H. (2011). Limitations in physicians knowledge when assessing dementia diseases—An evaluation study of a decision-support system. *Studies in Health Technology and Informatics*, *169*, 120–124.
21. Lindgren, H. (2011). Towards context-based inquiry dialogues for personalized interaction. In Y. Demazeau, M. Pechoucek, J. M. Corchado, & J. B. Pérez (Eds.), *PAAMS, advances in intelligent and soft computing* (Vol. 88, pp. 151–161). Berlin: Springer.
22. Lindgren, H. (2011). Towards personalized decision support in the dementia domain based on clinical practice guidelines. *User Modeling and User-Adapted Interaction*, *21*(4–5), 377–406.
23. Lindgren, H., & Winnberg, P. (2011). A model for interaction design of personalised knowledge systems in the health domain. In M. Szomszor & P. Kostkova (Eds.), *E-Health 2010, LNICST* (Vol. 69, pp. 235–242). Berlin: Springer.
24. Lindgren, H., & Yan, C. (2015). ACKTUS: A platform for developing personalized support systems in the health domain. In *Proceedings of the 5th international conference on digital health 2015, DH '15* (pp. 135–142). New York: ACM.
25. Marinagi, C. C., Spyropoulos, C. D., Papatheodorou, C., & Kokkotos, S. (2000). Continual planning and scheduling for managing patient tests in hospital laboratories. *Artificial Intelligence in Medicine*, *20*(2), 139–154.
26. McBurney, P., Eijk, R. M. V., Parsons, S., & Amgoud, L. (2003). A dialogue-game protocol for agent purchase negotiations. *Autonomous Agents and Multi-Agent Systems*, *7*(3), 235–273.
27. McBurney, P., & Parsons, S. (2001). Representing epistemic uncertainty by means of dialectical argumentation. *Annals of Mathematics and Artificial Intelligence*, *32*(1–4), 125–169.
28. McKEITH, I. G. (2002). Dementia with Lewy bodies. *The British Journal of Psychiatry*, *180*(2), 144–147.
29. Musen, M. A., Middleton, B., & Greenes, R. A. (2014). Clinical decision-support systems. In E. Shortliffe & J. Cimino (Eds.), *Biomedical informatics* (pp. 643–674). London: Springer.
30. Nieves, J. C., & Lindgren, H. (2015). Possibilistic nested logic programs and strong equivalence. *International Journal of Approximate Reasoning*, *59*, 1–19.
31. Nieves, J. C., Osorio, M., & Cortés, U. (2013). Semantics for possibilistic disjunctive programs. *Theory and Practice of Logic Programming*, *13*(1), 33–70.
32. Noor, K., Hunter, A., & Mayer, A. (2017). Analysis of medical arguments from patient experiences expressed on the social web. In *International conference on industrial, engineering and other applications of applied intelligent systems* (pp. 285–294). Springer.
33. Parsons, S., Wooldridge, M., & Amgoud, L. (2003). On the outcomes of formal inter-agent dialogues. In *Proceedings of the second international joint conference on autonomous agents and multiagent systems (AAMAS 2003)* (pp. 616–623). ACM.
34. Pasquier, P., & Chaib-Draa, B. (2005). Agent communication pragmatics: The cognitive coherence approach. *Cognitive Systems Research*, *6*(4), 364–395.
35. Patel, V. L., Kaufman, D. R., & Arocha, J. F. (2002). Emerging paradigms of cognition in medical decision-making. *Journal of Biomedical Informatics*, *35*(1), 52–75.
36. Prakken, H. (2006). Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, *21*(02), 163–188.
37. Rogers, A., David, E., Jennings, N. R., & Schiff, J. (2007). The effects of proxy bidding and minimum bid increments within ebay auctions. *ACM Transactions on the Web (TWEB)*, *1*(2), 9.

38. Singh, S., Ismail, B. I., Haron, F., & Yong, C. H. (2005). Architecture of agent-based healthcare intelligent assistant on grid environment. In *Parallel and distributed computing: Applications and technologies* (pp. 58–61). Springer.
39. Sun, R., & Naveh, I. (2004). Simulating organizational decision-making using a cognitively realistic agent model. *Journal of Artificial Societies and Social Simulation*, *7*(3), 1–5.
40. Tablado, A., Illarramendi, A., Bagüés, M. I., Bermúdez, J., & Goni, A. (2004). Aingeru: An innovating system for tele assistance of elderly people. In *TELECARE* (pp. 27–36).
41. Tentori, M., Favela, J., & Rodriguez, M. D. (2006). Privacy-aware autonomous agents for pervasive healthcare. *IEEE Intelligent Systems*, *21*(6), 55–62.
42. Walton, D. N. (1998). *The new dialectic: Conversational contexts of argument*. Toronto: University of Toronto Press.
43. Woolf, S. H., Grol, R., Hutchinson, A., Eccles, M., & Grimshaw, J. (1999). Clinical guidelines: Potential benefits, limitations, and harms of clinical guidelines. *BMJ: British Medical Journal*, *318*(7182), 527.
44. Yan, C., Nieves, J. C., & Lindgren, H. (2014). A multi-agent system for nested inquiry dialogues. In Y. Demazeau, F. Zambonelli, J. M. Corchado, & J. Bajo (Eds.), *Advances in practical applications of heterogeneous multi-agent systems. The PAAMS collection. PAAMS 2014. Lecture notes in computer science* (Vol. 8473, pp. 303–314). Cham: Springer.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.