HHAI 2024: Hybrid Human AI Systems for the Social Good
F. Lorig et al. (Eds.)

doi:10.3233/FAIA240188

Effective Maintenance of Computational Theory of Mind for Human-AI Collaboration

Emre ERDOGAN [a] and Frank DIGNUM [b] and Rineke VERBRUGGE [c]
[a] *Utrecht University, Utrecht, Netherlands*
[b] *Umeå University, Umeå, Sweden*
[c] *University of Groningen, Groningen, Netherlands*

Abstract. In order to enhance collaboration between humans and artificially intelligent agents, it is crucial to equip the computational agents with capabilities commonly used by humans. One of these capabilities is called Theory of Mind (ToM) reasoning, which is the human ability to reason about the mental contents of others, such as their beliefs, desires, and goals. For an agent to efficiently benefit from having a functioning computational ToM of its human partner in a collaboration, it needs to be practical in computationally tracking their mental attitudes and it needs to create approximate ToM models that can be effectively maintained. In this paper, we propose a computational ToM mechanism based on abstracting beliefs and knowledge into higher-level human concepts, referred to as *abstractions*. These abstractions, similar to those guiding human interactions (e.g., trust), form the basis of our modular agent architecture. We address an important challenge related to maintaining abstractions effectively, namely abstraction consistency. We propose different approaches to study this challenge in the context of a scenario inspired by a medical domain and provide an experimental evaluation over agent simulations.

Keywords. Theory of Mind, Abstraction, Human-AI Collaboration, Human-inspired Computational Model

1. Introduction

Hybrid Intelligence (HI) [1] is about integrating human and machine intelligence for the purpose of expanding human intellect instead of replacing it. Effective HI requires human-agent collaboration at its core, where a human and a computational agent partner, working together, complement each other's abilities to create fruitful partnerships.

One of the capabilities that help humans successfully maintain their social interactions with other humans is called Theory of Mind (ToM) reasoning [2,3]. Put simply, ToM refers to the human capacity to reason about the mental content of others such as their beliefs, desires, values, and goals. Possessing a functioning ToM is critical to understand and predict others' behaviour [4] and can provide further benefits to the possessor when used recursively. Recently, many computational ToM models have been developed to understand its effectiveness in competitive, cooperative, and mixed motive settings [5,6,7,8,9,10,11,12]. The results are mostly positive, suggesting that utilizing ToM leads to enhanced performance in the studied tasks.

Developing a practical computational ToM for human-agent interaction is valuable but challenging. Many existing ToM-using agent models begin by representing individual beliefs about others and constructing a ToM model from these. In contextually rich settings that feature continuous interaction, the agent accumulates numerous beliefs about others over time, some of which are applicable only in specific contexts and others are useful in different situations. This makes developing a comprehensive ToM model infeasible since computationally tracking all individual mental attitudes of all others is a costly approach. To continue being effective in its interactions with human partners over time, the agent should be efficient in keeping, maintaining, and utilizing these beliefs.

One candidate solution to this problem comes from human behaviour, called *abstracting*. As a problem solving technique, abstracting enables humans to form a broad understanding of the problem and its potential solutions, rather than focusing on specific details [13]. In complex social settings, this approach helps us approximate what we should look for in the interaction to reach our goals and make our decisions accordingly. Consider *trust* as an abstraction, which serves as a backbone in collaboration and mainly captures agents' confidence in each other's abilities, reliability, and commitment [14]. A human being, by using the abstracting technique, can efficiently utilize the relevant information about their partner to decide whether to trust the partner or not. Coupled with ToM reasoning, the human being can further understand if the partner trusts them back and correspondingly decide which actions they need to perform in their interactions.

This paper proposes a computational ToM mechanism based on abstracting beliefs and knowledge into higher-level abstractions that serve as practical approximations. We design a formal agent architecture based on epistemic logic [15] that provides a modular structure for storage and maintenance of individual beliefs, knowledge, and abstractions. For this agent to work with humans collaboratively, we need to address a challenge regarding *abstraction consistency*: Since an agent's beliefs and knowledge change over time, it is necessary to devise methods to revise abstractions from time to time in an efficient manner. We propose different mechanisms to study this challenge in the context of a scenario inspired from a medical domain and provide evaluation over simulations.

The rest of this paper is organized as follows. Section 2 sets up our motivating human-agent collaboration example. Section 3 describes how we formalize abstractions and computational ToM reasoning with epistemic logic for computational agents. Section 4 illustrates the use of the formal agent design, addresses our solutions to the challenge given above, and evaluates our solutions over agent simulations. Section 5 discusses our work in relation to related work and provides pointers for future directions.

2. Motivating Example

We consider a motivating example from the medical domain, featuring a collaboration between a computational agent doctor X and a human doctor Y for diagnosing a patient Z (inspired by the work of Erdogan et al. [16]). In this scenario, the computational agent doctor X is designed to complement the capabilities of the human doctor Y to improve the efficiency of the diagnostic process. They can distribute the tasks based on their respective strengths throughout the diagnostic procedure [17]. For instance, Y can handle tasks that require human traits such as conducting patient interviews and performing physical examinations, while X can focus on tasks that are computationally more viable [18,19] such as diagnostic testing (e.g., medical imaging [20]). In this setup, which can be seen as a collective decision-making process, both doctors can share their results with each other and discuss the diagnosis together.

We are particularly interested in situations where there is a conflict between decisions of X and Y and social skills are being used to resolve such conflicts through different techniques [21,22]. We particularly focus on trust. By aggregating beliefs and knowledge that are contextually relevant, X can determine whether it should trust Y or not. What makes it more interesting is that X can also reason about how Y abstracts her knowledge and beliefs to decide whether to trust X or not (i.e., how Y does her own approximation for trust) with the help of its computational ToM of Y. These abstractions can help X in choosing the best response to go with when a dispute occurs.

In this setting, X needs to explicitly have the contextually relevant beliefs and knowledge that are necessary to create and maintain its trust-related abstractions. These knowledge and beliefs can be generated and revised internally (e.g., by different processes of the agent) or acquired from external sources. X also needs a mechanism to do the required approximations properly. In our scenario, we assume that for X to trust Y, it needs to know that Y is a doctor, *and* believe that Y communicates well with X, *and* believe that Y is an expert in her field. Moreover, as part of ToM reasoning, X should also be able to capture Y's understanding of trust as that can be different than itself. We assume that X's ToM model for Y depicts that for Y to trust X, she needs to believe that X has good diagnostic capabilities *and* good communication skills.

3. Formal Design

3.1. Abstraction Elements

We define *agent* as an entity that has beliefs and knowledge about other agents, maintains its beliefs and knowledge over time, and uses them when interacting with other agents. Our formalization is based on epistemic logic [15] where propositions can be created from propositional atoms, together with negation and conjunction operators and knowledge and belief modalities per agent.

Knowledge and Beliefs: To formally represent *knowledge and beliefs* of a set of agents $\mathscr{X}$, we use the following language $\mathscr{L}_{KB}^{\mathscr{X}}$ given by the *Backus-Naur* form:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_X\varphi \mid L_X\varphi$$

Here, p are propositional atoms and $X \in \mathscr{X}$. Given $p_1 =$ "Y is a doctor", $K_X p_1$ and $L_X p_1$ can be read as "the agent X knows that Y is a doctor" and "the agent X believes that Y is a doctor", respectively. Notice that $L_Y K_X p_1$, which states that "the agent Y believes that the agent X knows that Y is a doctor", is a member of $\mathscr{L}_{KB}^{\mathscr{X}}$. Formulas with nested epistemic operators are useful in expressing agents' higher-order knowledge and beliefs.

Abstractions: An *abstract concept* A is a human-inspired, abstract decision-making heuristic which can guide agents in their interactions. $\mathscr{A}$ denotes the set of abstract concepts in our framework. Essentially, these abstract concepts are meaningful when defined in a relational manner. Thus, we define an *abstraction* as a proposition structured as $A(X,Y)$, where $A \in \mathscr{A}$ and $X,Y \in \mathscr{X}$.

Our formalization allows agents to hold (higher-order) knowledge and beliefs about abstractions in a nested manner. For example, given $p_2 = \mathit{Trust}(X,Y)$, $K_Y p_2$ and $L_X L_Y p_2$

can be read as “the agent Y knows that X trusts Y” and “the agent X believes that the agent Y believes that X trusts Y”, respectively. We refer to such (higher-order) knowledge and beliefs (about abstractions) simply as abstractions.

Abstractions can come about in different ways. For example, there is a vast literature on how one agent can learn if it should trust another agent using machine learning techniques [23,24]. The use of machine learning techniques necessitates to have a large number of interactions before a decision can be obtained. However, in real life, usually one needs to decide to trust another agent without the opportunity to interact with her too many times. Social cues and organizational constructs help in determining a quick decision. For example, you might trust someone because they are a doctor in a reputable hospital, even though you had no previous interactions with her. At the same time, it is important to be able to identify the reasons that led to trust [25], which is difficult to explain with machine learning techniques. Since our focus is on enabling agents to create abstractions quickly and reason on them effectively, we formalize abstraction through predefined rules, rather than data-driven techniques.

Abstraction Rules: An *abstraction rule* is a derivation rule in the form of $\varphi \rightarrow \phi$ such that ϕ is an abstraction. For instance, $p_1 \rightarrow p_2$ (i.e., “Y is a doctor” implies that “X trusts Y”) and $K_X p_1 \rightarrow L_X K_Y p_2$ (i.e., “X knows that Y is a doctor” implies that “X believes that Y knows that X trusts Y”) are both regarded as abstraction rules. φ could refer to various roles as described above as well as external information (e.g., my friend trusts Y).

Epistemic logic is useful for formally exploiting the implications of various epistemic principles such as $K_X p \rightarrow p$ (i.e., what is known is true) and $K_X p \rightarrow K_X K_X p$ (i.e., what is known is known to be known). We use the following prominent epistemological principles P_J, P_K and P_L in tandem with abstraction rules to derive abstractions:

P_J: $K_X \varphi \rightarrow L_X \varphi$ (i.e., knowledge implies belief)
P_K: $K_X(\varphi \rightarrow \phi) \rightarrow (K_X \varphi \rightarrow K_X \phi)$ (i.e., knowledge is closed under implication)
P_L: $L_X(\varphi \rightarrow \phi) \rightarrow (L_X \varphi \rightarrow L_X \phi)$ (i.e., belief is closed under implication)

For example, say we have the knowledge $K_X(p_1 \rightarrow p_2)$ (i.e., X knows that “Y is a doctor implies that X trusts Y”) and $K_X p_1$ (i.e., X knows that Y is a doctor). By using P_K and modus ponens, we can derive that $K_X p_1 \rightarrow K_X p_2$ and hence, $K_X p_2$.

3.2. Agent Architecture

Our proposed agent architecture consists of three modules. The **knowledge and belief module** keeps the agent’s knowledge and beliefs. At certain times, the agent derives abstractions from these, which are stored in the **abstraction module**. The **deliberation module** uses the abstractions to make decisions on how to interact with other agents.

Knowledge and Belief Module: For an agent X, $X^{KL} = (M, N)$ represents *X’s knowledge and belief module* such that:

- M is the *knowledge set* of X such that every member of the set is either of the form $K_X p$ (first-order knowledge), $K_X K_Y p$, or $K_X L_Y p$ (second-order knowledge) where p is a proposition and $Y \in \mathscr{X}$, and
- N is the *belief set* of X such that every member of the set is either of the form $L_X p$ (first-order belief), $L_X K_Y p$, or $L_X L_Y p$ (second-order belief) where p is a proposition and $Y \in \mathscr{X}$.

Whenever the agent interacts with others, the information that reaches the agent is stored in this module. M is a dynamic set such that new knowledge can be added to it. As generally understood in the literature, we assume that knowledge is always true. Hence, new knowledge would not conflict with existing knowledge, making M conflict-free by definition. On the other hand, N contains beliefs, which may or may not be true. Thus, a new belief can easily conflict with an existing one. For simplicity, we ensure that a newly created belief overrides the older conflicting belief, making the set conflict-free.

Abstraction Module: For an agent X, $X^{Abs} = (O,P,Q)$ represents *X's abstraction module* such that:

- O is X's *abstract concepts* where $O \subseteq \mathscr{A}$,
- P is X's *abstraction rules* such that every member of the set is of the form $K_X(\varphi \rightarrow \phi)$ or $L_X(\varphi \rightarrow \phi)$ where $\varphi \rightarrow \phi$ is an abstraction rule, and
- Q is X's *(current) abstractions*.

We assume that O is a static set that contains all possible abstract concepts the agent can have over time. Note that this set might be large but that does not mean that the agent will have abstraction rules in P or existing instances of abstractions in Q related to them. Each agent has a set of abstraction rules in P, which can be different for each agent. While we do not necessarily focus on the rules themselves, it is possible that this set is dynamic such that new rules are added as the agent sees fit or some rules are removed if not seen fit. Q holds the abstractions which the agent derives by using P_J, P_K, P_L, and modus ponens on its beliefs and knowledge. Similar to N, it is conflict-free: newer values of abstractions override the older values.

Deliberation Module: For an agent X, $X^{Del} = (R,S,T)$ represents *X's deliberation module* such that:

- R is *the set of actions* that X can do when interacting with other agents,
- S is *action deliberation rules* of X such that every member of the set is of the form $\phi \xrightarrow{\text{action}} r$ where ϕ is an abstraction or a conjunction of abstractions and $r \in R$, and
- T is X's *(current) action* where $T \in R$.

We assume R is a static set, such that possible actions cannot change during execution. S defines how the agent will deliberate with others based on its abstractions. For this paper, we assume these are given for each agent but essentially these rules can evolve over time based on interactions with others. For simplicity, we assume deliberation rules clearly identify which action will be taken in a given situation. T keeps the current action.

We represent X as $X = \langle X^{KL}, X^{Abs}, X^{Del} \rangle = \langle (M,N),(O,P,Q),(R,S,T) \rangle$. Although separated conceptually, the modules are connected to each other functionally. For example, for the agent to maintain its abstractions in P, it needs to check the content of X^{KL}. Also, for the agent to decide on its next action, it uses the action deliberation rules in S in combination with P. The next section illustrates this flow within our motivating example.

4. Abstraction Consistency

To illustrate use of abstractions in human-agent collaboration, we formalize the conflict resolution scenario given in Section 2 in which X needs to decide on the action to take next depending on the set of abstractions that X has about Y. For the sake of simplicity, we limit the number of possible abstractions that X can use to two, namely $L_X(\mathit{Trust}(X,Y))$ and $L_X(\mathit{Trust}(Y,X))$ (i.e., X's trust towards Y and Y's trust towards X, respectively) and the number of possible actions to four, namely $\textbf{Converse}(Y)$, $\textbf{Agree}(Y)$, $\textbf{Persuade}(Y)$, and $\textbf{Consult}(Y)$. Table 1 describes the content of all modules of $X = \langle(M,N),(O,P,Q),(R,S,T)\rangle$ and shows how X can use them.

Table 1. Design in use: X decides on its next action according to its abstractions.

M	$K_X(\mathit{Doctor}(Y))$
N	$L_X(\mathit{Expert}(Y))$ $L_X(\mathit{GoodCommunication}(Y))$ $L_X L_Y(\mathit{GoodCommunication}(X))$ $L_X L_Y(\mathit{GoodCapabilities}(X))$
O	$\{\mathit{Trust}\}$
P	$L_X(\mathit{Doctor}(Y) \wedge \mathit{Expert}(Y) \wedge \mathit{GoodCommunication}(Y) \rightarrow \mathit{Trust}(X,Y))$ $L_X(L_Y(\mathit{GoodCommunication}(X)) \wedge L_Y(\mathit{GoodCapabilities}(X)) \rightarrow \mathit{Trust}(Y,X))$
Q	$L_X(\mathit{Trust}(X,Y))$ $L_X(\mathit{Trust}(Y,X))$
R	$\{\mathit{Converse, Agree, Persuade, Consult}\}$
S	$L_X(\mathit{Trust}(X,Y)) \wedge L_X(\mathit{Trust}(Y,X)) \xrightarrow{\text{Action}} \textbf{Converse}(Y)$ $L_X(\mathit{Trust}(X,Y)) \wedge \neg L_X(\mathit{Trust}(Y,X)) \xrightarrow{\text{Action}} \textbf{Agree}(Y)$ $\neg L_X(\mathit{Trust}(X,Y)) \wedge L_X(\mathit{Trust}(Y,X)) \xrightarrow{\text{Action}} \textbf{Persuade}(Y)$ $\neg L_X(\mathit{Trust}(X,Y)) \wedge \neg L_X(\mathit{Trust}(Y,X)) \xrightarrow{\text{Action}} \textbf{Consult}(Y)$
T	**Converse(Y)**

The goal of X is to interact with Y according to the abstractions that it has about Y. To do that, it first derives the abstractions in Q by using M, N, P, and epistemological principles P_J, P_K, and P_L. Then, by using Q and S, it decides on the action to take next. Table 1 shows that both $L_X(\mathit{Trust}(X,Y))$ and $L_X(\mathit{Trust}(Y,X))$ hold at the beginning; so, X chooses to resolve the conflict by conversing with Y to arrive at a joint resolution.

X may need to resolve many other conflicts during its partnership with Y. Thus, X needs to choose the correct actions through its abstractions that are consistent with its current knowledge and beliefs. To do that, X needs to update Q in accordance with the changes in M and N. Furthermore, the update mechanism should be as efficient as possible for X to be effective in its action-decision process: X should update Q as much as needed, but also, only when necessary. Since M and N can also change with knowledge and beliefs that do not affect the derivation of abstractions, X should be careful not to do any unnecessary update-checks for its abstractions. Thus, X requires a concrete procedure for deciding when to revise its abstractions and the accompanying actions.

4.1. Update Methods

In order to realize abstraction consistency, an agent needs to check and update its abstractions.There are a number of factors that are important to consider, such as frequency (of updates), change (in the knowledge and beliefs), and engagement (with others). Considering these factors, we formulate the following update strategies (in **bold**):

- **Frequent** updates its abstractions after every change in N, without considering engagement.
- **Infrequent** updates its abstractions after every 10 rounds, without considering if there are changes or engagement.
- **Revision** updates its abstractions after every belief revision in N, but does not consider belief addition or engagement.
- **Deliberation** updates its abstractions only before deliberation without considering changes or frequency.
- **Change** updates its abstractions only before deliberation and only if there is a change in N, without considering frequency.
- **Selective-Change** updates its abstractions only before deliberation and only if there is an abstraction-related change in N.

Hypothesis 1 *An agent* $X = \langle X^{KL}, X^{Abs}, X^{Del} \rangle$ *with* ***Selective-Change*** *Strategy is* ***the most effective*** *(among all six strategies) in obtaining* ***abstraction consistency****.*

4.2. Evaluation

We evaluate the performance of these strategies over simulations. We have designed a computational agent to simulate the behaviour of $X = \langle (M,N),(O,P,Q),(R,S,T) \rangle$ for the human-agent collaboration scenario. X is capable of adding beliefs to and revising beliefs in N, updating its abstractions in Q, and doing deliberations. We do not simulate human doctor Y's behaviour explicitly in the simulations.

A simulation lasts 10000 rounds. There are three types of events that can occur with the same probability (1/3) during the simulation: i) an ordinary (i.e., non-abstraction-related) belief is created and X adds it to N (or revises it in N); ii) an abstraction-related (e.g., $L_X(\mathit{Expert}(Y))$) belief is created and X adds it to N (or revises it in N); or iii) a deliberation moment comes (i.e., conflict occurs) and X decides on the action to take.

The experiment provides two basic metrics to measure agent performance. One is the number of abstraction updates an agent does throughout the simulation. The second one is the number of consistent abstractions that it has at the time of deliberations, where an abstraction is considered consistent at the time of deliberation if current knowledge and belief base of the agent would also infer the same abstraction. For example, suppose X chooses **Converse**(Y) because of having $L_X(\mathit{Trust}(X,Y))$ and $L_X(\mathit{Trust}(Y,X))$ in Q but in fact $L_X(\mathit{Expert}(Y))$ is recently removed from N. This implies $L_X(\mathit{Trust}(X,Y))$ should not be in Q as well, implying a mistake; hence, makes it an inconsistent abstraction. We measure *abstraction-effectiveness* by combining these two metrics, which is equal to the number of consistent abstractions used in deliberations per abstraction update.

We have run the simulation 10 times with 10 different seeds for randomization purposes and averaged the results. Table 2 shows the number of times abstractions are updated as well as the number of times correct abstractions are used in deliberations (for each agent type). As expected, **Frequent** agent performs the most updates as it does so with every change in its belief set and thus has zero errors (e.g., its abstractions are always up-to-date). **Infrequent** agent, on the other hand, performs the fewest updates but because of this it had the most errors in its abstractions. **Revision** agent is less efficient than **Infrequent** agent, yet it also has a lower usage of inconsistent abstractions. **Deliberation** agent and **Change** agent do not create inconsistent abstractions but they are also not as economical as **Infrequent** agent in their abstraction updates. **Selective-Change**,

Table 2. Abstraction-effectiveness of 6 strategies over 10 simulations. Average number of deliberations: 3305

Agent Type	*#Abstraction Updates*	*#Consistent Abstractions*	*Abstraction-Effectiveness*
Frequent	3401.1	3305	0.97
Infrequent	1000	2584.7	2.58
Revision	3296.1	3302.1	1.00
Deliberation	3305	3305	1.00
Change	1668.2	3305	1.98
Selective-Change	1110.1	3305	2.98

on the other hand, excels in the task. It does not make mistakes when using abstractions in deliberation moments and is nearly as effective as **Infrequent** agent in updating its abstractions. Among all, **Selective-Change** agent illustrates the best use of abstractions in deliberations, being the most abstraction-effective; this corroborates Hypothesis 1.

5. Discussion

There exist recent agent models that use various forms of computational ToM reasoning for human-agent collaboration. Piazza and Behzadan [26] design a ToM-based technique to differentiate agents based on cooperativeness where communication plays a crucial role. Wu, Sequeria, and Pynadath [27] focus on understanding how humans interact in collaborative teaming settings where they know little about others' goals and intentions. Bara et al. [28] introduce the concept of collaborative plan acquisition, where humans and AI agents work together to learn and communicate to acquire a complete plan for joint tasks. Their results highlight the importance of modeling a partner's mental states explicitly. Montes et al. [12] introduce an agent model that combines ToM reasoning with abductive reasoning capabilities and demonstrate their computational ToM model's performance in the context of the Hanabi game [29]. Erdogan et al. [16] provide an outline of a computational ToM framework based on abstracting individual beliefs into higher-level concepts such as social roles, norms, and human values.

Our work provides a novel approach in ToM-based agent modeling with explicit use and maintenance of abstractions, yet it is also constrained by several limitations. Firstly, an abstraction-using agent should model that different people can build and maintain their abstractions in different manners (e.g., their trust [30]), which our examples do not feature. Thus, to be more flexible in collaborations, such an agent should be capable of properly updating its abstractions when they are not working well (e.g., changing its abstraction rules concerning others' abstractions such as their trust in itself). Moreover, we only include the concept of trust in our examples and evaluation so that we sufficiently focus on the mechanics of the abstraction use and maintenance. Including other abstractions, such as respect or affinity, can enable us to create more detailed models.

Acknowledgements

This research was funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research (https://www.hybrid-intelligence-centre.nl).

References

[1] Akata Z, Balliet D, De Rijke M, Dignum F, Dignum V, Eiben G, et al. A research agenda for hybrid intelligence: augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. Computer. 2020;53(08):18-28.

[2] Premack D, Woodruff G. Does the chimpanzee have a theory of mind? Behavioral and Brain Sciences. 1978;1(4):515-26.

[3] Carruthers P, Smith PK. Theories of theories of mind. Cambridge University Press; 1996.

[4] Baker JE, Myles BS. Social skills training : for children and adolescents with Asperger syndrome and social-communication problems. Autism Asperger; 2003. Available from: https://cir.nii.ac.jp/crid/1130282271670507008.

[5] de Weerd H, Verbrugge R, Verheij B. How much does it help to know what she knows you know? An agent-based simulation study. Artificial Intelligence. 2013;199-200:67-92.

[6] de Weerd H, Verbrugge R, Verheij B. Higher-order theory of mind in the tacit communication game. Biologically Inspired Cognitive Architectures. 2015;11:10-21.

[7] Devin S, Alami R. An implemented theory of mind to improve human-robot shared plans execution. In: 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE; 2016. p. 319-26.

[8] Baker CL, Jara-Ettinger J, Saxe R, Tenenbaum JB. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. Nature Human Behaviour. 2017;1(4):1-10.

[9] Winfield AFT. Experiments in artificial theory of mind: From safety to story-telling. Frontiers in Robotics and AI. 2018;5:75.

[10] Buehler MC, Weisswange TH. Theory of mind based communication for human agent cooperation. In: 2020 IEEE International Conference on Human-Machine Systems (ICHMS). IEEE; 2020. p. 1-6.

[11] de Weerd H, Verbrugge R, Verheij B. Higher-order theory of mind is especially useful in unpredictable negotiations. Autonomous Agents and Multi-Agent Systems. 2022;36(2):30.

[12] Montes N, Luck M, Osman N, Rodrigues O, Sierra C. Combining theory of mind and abductive reasoning in agent-oriented programming. Autonomous Agents and Multi-Agent Systems. 2023;37(2):36.

[13] Polya G. How to Solve It: A New Aspect of Mathematical Method. vol. 85. Princeton university press; 2004.

[14] Mattessich PW, Monsey BR. Collaboration: what makes it work. A review of research literature on factors influencing successful collaboration. ERIC; 1992.

[15] Meyer JJC, van der Hoek W. Epistemic logic for AI and computer science. 41. Cambridge University Press; 2004.

[16] Erdogan E, Dignum F, Verbrugge R, Yolum P. Abstracting minds: Computational theory of mind for human-agent collaboration. In: HHAI2022: Augmenting Human Intellect. IOS Press; 2022. p. 199-211.

[17] Balogh EP, Miller BT, Ball JR. The Diagnostic Process. In: Balogh EP, Miller BT, Ball JR, editors. Improving Diagnosis in Health Care. National Academies Press; 2015. p. 31-80.

[18] Chen Y, Argentinis JE, Weber G. IBM Watson: How cognitive computing can be applied to big data challenges in life sciences research. Clinical Therapeutics. 2016;38(4):688-701.

[19] Briganti G, Le Moine O. Artificial intelligence in medicine: Today and tomorrow. Frontiers in Medicine. 2020;7:27.

[20] Liu X, Faes L, Kale AU, Wagner SK, Fu DJ, Bruynseels A, et al. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: A systematic review and meta-analysis. The Lancet Digital Health. 2019;1(6):e271-97.

[21] Mohr J, Spekman R. Characteristics of partnership success: partnership attributes, communication behavior, and conflict resolution techniques. Strategic Management Journal. 1994;15(2):135-52.

[22] Thomas KW. Conflict and conflict management: Reflections and update. Journal of Organizational Behavior. 1992:265-74.

[23] Teacy WL, Patel J, Jennings NR, Luck M. Travos: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and Multi-Agent Systems. 2006;12:183-98.

[24] Granatyr J, Botelho V, Lessing OR, Scalabrin EE, Barthès JP, Enembreck F. Trust and reputation models for multiagent systems. ACM Computing Surveys (CSUR). 2015;48(2):1-42.

[25] Castelfranchi C, Falcone R. Trust theory: A socio-cognitive and computational model. John Wiley & Sons; 2010.

[26] Piazza N, Behzadan V. A Theory of Mind Approach as Test-Time Mitigation Against Emergent Adversarial Communication. In: Proceedings of the 2023 International Conference on Autonomous Agents

and Multiagent Systems; 2023. p. 2842-4.

[27] Wu H, Sequeira P, Pynadath DV. Multiagent inverse reinforcement learning via theory of mind reasoning. In: Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems; 2023. p. 708-16.

[28] Bara CP, Ma Z, Yu Y, Shah J, Chai J. Towards collaborative plan acquisition through theory of mind modeling in situated dialogue. arXiv preprint arXiv:230511271. 2023.

[29] Bard N, Foerster JN, Chandar S, Burch N, Lanctot M, Song HF, et al. The Hanabi challenge: A new frontier for AI research. Artificial Intelligence. 2020;280:103216.

[30] Cho JH, Chan K, Adali S. A survey on trust modeling. ACM Computing Surveys (CSUR). 2015;48(2):1-40.