A BELIEF REVISION MECHANISM WITH TRUST REASONING BASED ON EXTENDED RECIPROCAL LOGIC FOR MULTI-AGENT SYSTEMS

Sameera Basit and Yuichi Goto

Department of Information and Computer Science, Saitama University,
338-8570 Saitama, Japan

ABSTRACT

When an agent receives messages from other agents, it does belief revision. A belief revision includes, i) a trust reasoning process, i.e., it obtains new belief related to the messages, and deduces implicitly unknown beliefs from the obtained belief; ii) in the case of contradiction in the belief set, it resolves the contradiction. So, trust reasoning, and belief revision must be included in the decision-making process of an intelligent agent in multi-agent systems. Although a belief revision mechanism with trust reasoning is demanded to construct multi-agent systems, there is no such belief revision mechanism. We, therefore, present a belief revision mechanism with trust reasoning based on extended reciprocal logic for multi-agent systems.

KEYWORDS

Multi-agent Systems, Trust Relationship, Trust Reasoning, Strong Relevant Logics, Belief Revision

1. INTRODUCTION

A trust relationship is one of the important reciprocal relationships in our society and cyberspace. Many reciprocal relationships must concern two parties [1]. Especially, the trust relationship is the basis of communications among agents (human to human, human to system, and system to system), and the basis of the decision-making of the agents.

Trust reasoning must be included in the decision-making process of an agent with reasoning capability, an intelligent agent for short, in multi-agent systems. Trust reasoning is a process to draw propositions from already known propositions using the degree of trust of an agent or a received message. A belief of an agent is a proposition that the agent believes, i.e., observed facts, already given theories and assumptions. Any agent in multi-agent systems can extend its belief set by receiving messages from other agents and observing its external environment or own internal status. Especially, an intelligent agent deduces implicitly included propositions from its belief set. After that, the agent decides the next actions according to its current belief set. An intelligent agent in an open system should be able to change the way it handles messages from other agents depending on the degree of trust of the agents because not all agents in the system can be trusted. Thus, an intelligent agent should be able to do trust reasoning for its decision-making.

David C. Wyld et al. (Eds): CCNET, AIMLA, CICS, IOTBS, NLTM, COIT - 2023
pp. 161-173, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130413

Belief revision must also be included in the decision-making process of an agent in multi-agent systems. Belief revision is a process of solving a contradiction in a target belief set to keep the belief set consistent. A belief set is consistent if and only if the set does not include both a proposition and its negation. In an open multi-agent system in the real world, the belief set of an agent is not always consistent, because a given assumption and an observed fact, or a previously observed fact and the current observed fact are sometimes explicitly or implicitly contradicted. Thus, an agent should be able to do belief revision. Moreover, in general, a trust relationship is not an eternal relationship. Although an agent is trusted at a point in time, the agent will not be trusted at another point in time. Changing trust relationships among agents, an agent updates its belief set by belief revision.

Although a belief revision mechanism with trust reasoning is demanded to construct multi-agent systems, there is no such belief revision mechanism. On one hand, the best-known work on modeling belief revision is the so-called Alchourrón, Gärdenfors, and Makinson's (AGM) theory or AGM model[2,3,4]. The AGM model is not suitable for the belief revision mechanism with trust reasoning because the AGM model adopts classical mathematical logic [5]. Classical mathematical logic is a suitable logic system underlying proving but not reasoning [5]. On the other hand, a well-known belief revision mechanism is the so-called truth maintenance systems, belief revision systems, or reason maintenance systems [6]. Essentially, the concept of truth maintenance systems is independent of a specific logic system. However, there is no truth maintenance system based on a logic system underlying trust reasoning.

This paper presents a belief revision mechanism with trust reasoning based on extended reciprocal logic for multi-agent systems. The belief revision mechanism is a Doyle's-style approach (truth maintenance system approach) to deal with the inconsistency in an agent's belief set. The mechanism consists of two parts. First, trust reasoning based on extended reciprocal logic is applied to the deduction process. Extended reciprocal logic is a candidate fora suitable logic system underlying trust reasoning. The second part deals with the belief revision of each agent in multi-agent systems. The proposed mechanism uses the concept of a derivation path. A derivation path can be viewed as a representation of a belief set that is gradually developed and modified as a result of changes in trust relationships with other agents. If a contradiction occurs in the belief set, a revision process is triggered which allows forward and backtracking within the derivation path to track beliefs that cause inconsistency in the agent's belief set.

The rest of the paper is organized as follows: Section 2 shows extended reciprocal logic as a suitable logic system underlying trust reasoning. Section 3 describes a belief revision mechanism with trust reasoning based on extended reciprocal logic. Section 4 illustrates the application of the belief revision mechanism. Some concluding remarks are given in section 5.

2. EXTENDED RECIPROCAL LOGIC

A logic system underlying trust reasoning should be able to deal with various trust properties. A trust relationship consists of a trustor, a trustee, and the trust property, indicating that the trustor believes that the trustee satisfies the trust property [7]. In the context of trust, not all the information from the other agent can be taken as a true message, i.e., "an agent α trusts another agent β with respect to a certain property" means that "α believes that β satisfies this property." Demolombe [8] defined several trust properties. His definitions are as follows.

- *Sincerity*: An agent α trusts in the sincerity of an agent β if f if β informs α about a proposition p then β believes p.
- *Validity*: An agent α trusts in the validity of an agent β if f if β informs α about a proposition p then p is the case.

– *Completeness*: An agent α trusts in the completeness of an agent βiffif p is the casethen β informs α about p.
– *Cooperativity*: An agent α trusts in the cooperativity of an agent βiffif β believes p then β informs *alpha* about p.
– *Credibility*: An agent α trusts in the credibility of an agent β iffif β believes p then p is the case.
– *Vigilance*: An agent α trusts in the vigilance of an agent β iffif p is the case then β believes p.

Trust reasoning is a process to draw propositions from already known propositions using the degree of trust of an agent or a received message. Thus, a logic system underlying trust reasoning should be able to deal with such trust properties.

A logic system underlying trust reasoning should be suitable for forward reasoning. Classical mathematical logic and its various conservative extensions are not suitable for logic systems underlying reasoning because they have paradoxes of implication [9, 10]. Strong relevant logic has rejected those paradoxes of implication and is considered the universal basis of various applied logic for knowledge representation and reasoning [5]. Thus, strong relevant logic and its conservative extensions are candidates for logic systems underlying reasoning. Reciprocal logic [1] is one of the conservative extensions of strong relevant logic to deal with various reciprocal relationships, including trust relationships. However, the reciprocal logic cannot deal with the trust properties [11, 12].

Therefore, a logic system underlying trust reasoning, named extended reciprocal logic, was proposed [11, 12]. Extended reciprocal logic, ERL for short, is an extension of reciprocal logic by introducing trust properties, i.e., sincerity, validity, completeness, cooperativity, credibility, and vigilance, to the reciprocal logic. The extended reciprocal logic is a hopeful candidate for a logic system underlying trust reasoning.

ERL consists of several predicates, two modal operators, and several axioms added to the reciprocal logic. Since ERL is one of the conservative extensions of strong relevant logic, ERL adopts all logical theorems of strong relevant logic. ERL also adopts all logical theorems of reciprocal logic. Below are the modal operators, predicates for representing messages, axioms, and inference rules of ERL.

Modal Operators are as follows.

- $Bel_i(p)$: agent I believes that a proposition p is true.
- $Inf_{i,j}(p)$: agent I has informed agent j about p.

ERL provides a predicate $TR(pe_1,pe_2,PROP)$ where pe_1 and pe_2 are agents, and $PROP$ is an individual constant that represents trust properties: sincerity, validity, completeness, cooperativity, credibility, and vigilance in extended reciprocal logic. For example, $TR(pe_1,pe_2,sincerity)$ means "pe_1 trusts pe_2 in sincerity", $TR(pe_1,pe_2,credibility)$ means "pe_1 trusts pe_2 in credibility", $TR(pe_1,pe_2,completeness)$ means "pe_1 trusts pe_2 in completeness", and in the same way, we can define a predicate for other trust properties as well. Additionally, $TR(pe_1,pe_2,all)$ means "pe_1 trusts pe_2 in all trust properties".
Axioms are as follows.

*ERcL*1: $\forall i \forall j(TR(i,j,sincerity) \Rightarrow (Inf_{j,i}(A) \Rightarrow Bel_j(A)))$
*ERcL*2: $\forall i \forall j(TR(i,j,validity) \Rightarrow (Inf_{j,i}(A) \Rightarrow A))$
*ERcL*3: $\forall i \forall j(TR(i,j,vigilance) \Rightarrow (A \Rightarrow Bel_j(A)))$

$ERcL4$: $\forall i \forall j(TR(i,j,credibility) \Rightarrow (Bel_j(A) \Rightarrow A))$
$ERcL5$: $\forall i \forall j(TR(i,j,cooperativity) \Rightarrow (Bel_j(A) \Rightarrow Inf_{j,i}(A)))$
$ERcL6$: $\forall i \forall j(TR(i,j,completeness) \Rightarrow (A \Rightarrow Inf_{j,i}(A)))$
BEL: $\forall i(Bel_i(A \Rightarrow B) \Rightarrow (Bel_i(A) \Rightarrow Bel_i(B)))$
ERL has three inference rules: modus ponens $\Rightarrow E$,adjunction $\wedge I$, and necessitation $Bel - Nec$. The two of three rules come from strong relevant logic. The $Bel - Nec$is introduced to the reciprocal logic.

$\Rightarrow E$: "from A and $A \Rightarrow B$ to infer B" (Modus Ponens)
$\wedge I$: "from A and B infer $A \wedge B$" (Adjunction)
$Bel - Nec$: "if A is a logical formula, then so is $Bel_i(A)$" (Necessitation)
Conclusively, ERL is $RcL \cup \{ERcL1,...,ERcL6,BEL\}$ where RcL is all axioms of the reciprocal logic. Trust reasoning based on ERL is deductive reasoning from given logical formulas and all logical theorems of ERL.

3. Belief Revision Mechanism With Trust Reasoning Based On Extended Reciprocal Logic

An agent in a multi-agent system has a set of beliefs as observed facts, previously given theories, and hypotheses. Using a set of beliefs, the agent calculates trust relationships between other agents by using trust reasoning within the domain to determine which agent should be trusted by the agent. When the agent receives messages from other agents, it does belief revision. Each time an agent in a domain receives a message from another agent, it undergoes a series of steps, as depicted in figure 1. The belief revision mechanism is comprised of two stages, as of the first stage it undergoes a trust reasoning process, i.e., it obtains new beliefs related to the messages, and deduces implicitly unknown beliefs from the obtained beliefs. These beliefs become part of the agent's belief set. In the second stage, if the deduced beliefs contradict pre-existing beliefs in the agent's belief set, it resolves the contradiction to maintain consistency.

In our belief revision mechanism, if a contradictory belief is entered into the belief set, a revision procedure is initiated to work backward through the path following the belief contained in the label, seeking to determine which belief may have contributed to the contradiction. In order to eliminate the contradiction, some of the existing beliefs are removed from the set of beliefs, and use the labels once again to remove all deductions that originated from these beliefs from the set of current beliefs. Although this process may result in some complexity issues, it is nevertheless theoretically feasible. Details of each sub-process of the belief revision mechanism are discussed in the following sections.

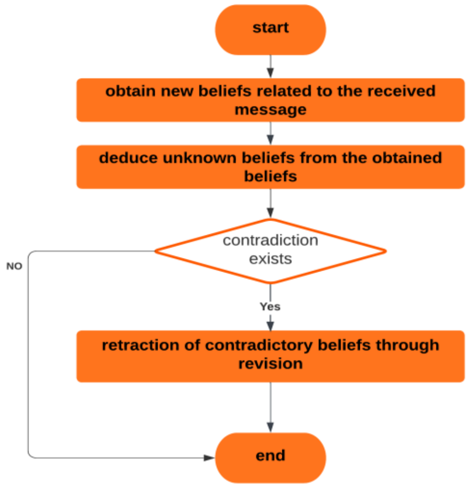

Figure 1.Belief revision process with trust reasoning

3.1. Creation of logical formulas to obtain new beliefs related to the received message

Upon receiving messages from other agents within a domain, new beliefs are obtained by generating logical formulas. To generate a logical formula that indicates that an agent has informed another agent about a message, e.g., "*m is valid*" is a message informed by agent b to agent a, and as a predicate, it will be represented as *is Valid(m)*. Then its related logical formula will be generated as $Inf_{b,a}$ *(is Valid(m))*. From these generated logical formulas new beliefs, e.g., Bel_b *(is Valid(m))* is obtained.

3.2. Deduction of unknown beliefs from the obtained beliefs

Through trust reasoning using axioms, and inference rules from the ERL. This deduced implicit unknown beliefs from the obtained beliefs, and this deduced belief becomes the part of agent's belief set. Each agent maintains a belief set as a derivation path. Deduced beliefs are entered into the derivation path. As a result of the deduction process, an agent gradually adds or modifies its beliefs. As new beliefs are added to the belief set at each time instance, the derivation path evolves over time. Additionally, the derivation path identifies which inference rule was utilized, as well as which beliefs were used as premises or sources using the labeled formula concept.

A deduced belief in a derivation path is labeled with the time stamp, i.e., an integer indicating the instance at which this occurred. The time stamp serves as an index indicating the logical formula position in the belief set. Since these deduced beliefs are derived from premises using inference rules. These labels contain a record of which inference rule was used, as well as which beliefs were used as premises, or sources. This way the agent knows all the logical consequences of each logical formula in his belief set. A label is defined as an ordered 4-tuple (index, from, to, status) [13], where :

1. index is a non-negative integer, the index, representing the position of the deduced belief in the belief set.
2. from-list contains information about premises, and inference rules used to derive thededuced belief.

3. to-list contains an index of all deduced beliefs where the given deduced belief servesas a premise.
4. status, using values *on* and *off*, indicates that only beliefs with status *on* can be used as premises in the deduction process. Whenever a deduced belief is first entered into the belief set, it is assigned status *on*.

3.3. Retraction of Contradictory Belief

Trust reasoning deduces beliefs that sometimes contradict pre-existing beliefs in the agent's belief set. Upon contradiction, a revision procedure is triggered, which disbelieves previously held beliefs, thus retracting the belief set by the contradictory belief. Usually, beliefs can be obtained as a message received from other agent in a domain, or it can be derived from the trust reasoning process. The procedure has three steps:

1. By backtracking through the belief set, starting with the from-list in the label of the contradictory belief, identify the beliefs that were involved in the derivation of the contradictory belief causing inconsistency in the belief set.
2. Change the status of involved beliefs to *off*, as many as necessary to invalidate the derivation of the given contradictory belief. The decision as to which status to turn *off* can be decided by retracting the one that is least believed generally identified by epistemic entrenchment value [3]. In cases where all the involved beliefs are equally believed, a random choice can be made. In some systems, this retraction process may be automated, and in others, it may be human-assisted [15].
3. Forward chains using the to-lists, identify all beliefs whose derivations were based on the retracted belief, and put their status to *off* as well.

This retraction of beliefs will include those beliefs that cause the agent's belief set to be inconsistent. Changing a belief's status from *on* to *off* occurs whenever a contradiction occurs. The objective of the revision procedure is to remove such contradictory beliefs from the agent's belief set.

The following sections will discuss the application of the belief revision mechanism in two case studies, a scenario about public key infrastructure, and a scenario about a spy novel.

4. APPLICATION OF THE BELIEF REVISION MECHANISM IN PUBLIC KEY INFRASTRUCTURE

As an example, we demonstrated the application of the belief revision mechanism in public key infrastructure PKI. When a change in trust relationships occurs between agents, it affects the trust reasoning process, and as a result, it deduces different results from trust reasoning. Following is the public key infrastructure scenario depicting trust relationships, and the exchange of messages between agents.

4.1. Public key infrastructure PKI scenario

In the PKI scenario, agents e_1, e_2, and e_3 exchange messages as certificates among themselves. Agent e_1 is informed about certificate $c_1$by the parent of the agent. We consider that every agent trusts its parent agent in its validity. Furthermore, agents e_2 and e_3 inform agent e_1 about certificates c_2 and c_3 respectively. Agent $e1$ doesn't believe the certificates c_2 and $c_3$but wishes to use them. Therefore, based on the trust relationships between agents, messages such as certificates can be reasoned out as beliefs through trust reasoning. Moreover, taking into consideration that agent $e_4$informs that c_1 is not valid, here if the deduced belief through the trust reasoning process contradicts the existing beliefs of agent e_1 belief set revision process will be invoked.

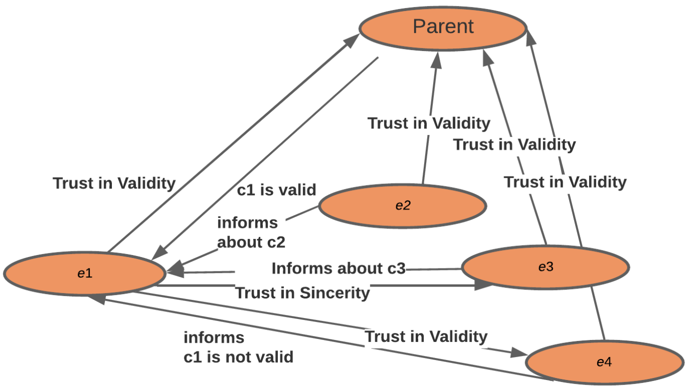

Figure 2. Public key infrastructure scenario

4.2. Formalization

To formalize the above scenario, we defined the following constants, functions, and predicates.

Individual variables:

- *e*: an agent
- *c*, *c*′: certifications

Individual constants:

- e_1,e_2, e_3, e_4: agents
- c_1, c_2, c_3, c_4: certifications
- *today*: date of today

Functions:

- *I*(*c*): Issuer of certification *c*.
- *S*(*c*): Subject of certification *c*.
- *PK*(*c*): Public key of *c*.
- *SK*(*c*): Share key of *c*.
- *DS*(*c*): Start date of *c*.
- *DE*(*c*): End date of *c*.
- *Sig*(*c*): Signature of *c*.
- *parent*(*e*): The parent of agent *e*.

Predicates:

- *inCRL*(*c*): *c* is in the certification revocation list.
- *isValid*(*x*): *x* is valid.
- *isSigned*(*x*,*k*): *x* is message signed by key *k*.
- $x = y$: *x* is equal to *y*.

- $x \le y$: x is equal to or less than y.
- $x < y$: x is less than y.

Empirical theories of PKI

We can assume the following empirical theories.

PKI1: $\forall e(TR(e,parent(e),validity))$

(Any agent trusts its parent agent in validity.)

PKI2: $\forall c(\exists c'((isValid(c'))) \wedge (I(c) = S(c')) \wedge (isSigned(c,PK(c')))) \Rightarrow isValid(Sig(c)))$

PKI3: $\forall c((isValid(Sig(c)) \wedge (DS(c) \le today) \wedge (today < DE(c)) \wedge \neg inCRL(c)) \Rightarrow isValid(c))$

(PKI2 and PKI3 allow to verify the signature, and certificate itself on the basis of another certificate whose validity has been proven.)

Logical theories

We can assume the following logical formulas.

P1-1: $I(c_2) = S(c_1)$, P1-2: $I(c_3) = S(c_1)$

(These observed facts are used as premises in our reasoning process and it is true in this scenario only.)

P2-1: $isSigned(c_2,PK(c_1))$

P2-2: $isSigned(c_3,PK(c_1))$

(A certificate c_2 or c_3 is signed by the subject of certificate c_1 with the private key corresponding to the public key of c_1.)

P3-1: $Inf_{parent(e1),e1}(isValid(c_1))$,

P3-2: $Inf_{e3,\, e1}(isValid(c_3))$

(The parent agent of e_1 has informed e_1 about "certificate c_1 is valid".)

P3-3: $Inf_{e4,e1}(\neg isValid(c_1))$

P4: $TR(e_1,e_3,sincerity)$ (*assumption*)

P4-1: TR $(e_1,e_4, validity)$ (*assumption*)

P5-1: $DS(c_2) \le today$, P5-2: $DS(c_3) \le today$ (*assumption*)

P6-1: $today < DS(c_2)$, P6-2: $today < DS(c_3)$ (*assumption*)

P7-1: $\neg inCRL(c_2)$, P7-2: $\neg inCRL(c_3)$ (*assumption*)

4.3. Trust reasoning process

Case 1: Agent e_1 received certificate c_1 as a message from its parent.

1. $Inf_{parent(e1),e1}(isValid(c_1)) \Rightarrow isValid(c_1)$ [from PKI1, ERcL2 with $\Rightarrow E$]
2. $isValid(c_1)$ [from P3-1, 2]
3. $Bel_{e1}(isValid(c_1))$ [from 2 with $Bel - Nec$]

Case 2: Agent e_1 received certificate c_2 as message from agent e_2

4. $(I(c_2) = S(c_1)) \wedge isSigned(c_2,PK(c_1))$ [from P1-1 and P2-1 with $\wedge I$]
5. $Bel_{e1}((I(c_2) = S(c_1)) \wedge isSigned(c_2,PK(c_1)))$ [from 4 with $Bel - Nec$]
6. $(isValid(c_1) \wedge (I(c_2) = S(c_1)) \wedge (isSigned(c_2,PK(c_1))) \Rightarrow isValid(Sig(c_2))$ [Replaced c with c_2 and c' with c_1 in PKI2]
7. $Bel_{e1}(isValid(c_1) \wedge (I(c_2) = S(c_1)) \wedge (isSigned(c_2,PK(c_1)))) \Rightarrow isValid(Sig(c_2)))$ [from 6 with $Bel - Nec$]
8. $Bel_{e1}(isValid(c_1) \wedge (I(c_2) = S(c_1)) \wedge (isSigned(c_2,PK(c_1)))) \Rightarrow Bel_{e1}(isValid(Sig(c_2)))$ [from BEL and 7 with $\Rightarrow E$]

9. $Bel_{e1}(isValid(Sig(c_2)))$ [from 5 and 8 with $\Rightarrow E$]
10. $Bel_{e1}(DS(c_2) \leq today)$, $Bel_{e1}(today < DS(c_2))$, $Bel_{e1}(\neg inCRL(c_2))$ [from each of P5-1, P6-1, and P7-1 with $Bel - Nec$]
11. $Bel_{e1}(isValid(Sig(c_2)) \wedge (DS(c_2) \leq today) \wedge (today < DE(c_2)) \wedge \neg inCRL(c_2))$ [from 10 with $\wedge I$]
12. $isValid(Sig(c_2)) \wedge (DS(c_2) \leq today) \wedge (today < DE(c_2)) \wedge \neg inCRL(c_2)) \Rightarrow isValid(c_2)$ [Replaced c with c_2 in PKI3]
13. $Bel_{e1}(isValid(Sig(c_2)) \wedge (DS(c_2) \leq today) \wedge (today < DE(c_2)) \wedge \neg inCRL(c_2)) \Rightarrow isValid(c_2)$ [from 12 with $Bel - Nec$]
14. $Bel_{e1}(isValid(Sig(c_2)) \wedge (DS(c_2) \leq today) \wedge (today < DE(c_2)) \wedge \neg inCRL(c_2)) \Rightarrow Bel_{e1}(isValid(c_2))$ [from BEL and 13 with $\Rightarrow E$]
15. $Bel_{e1}(isValid(c_2))$ [from 11 and 14 with $\Rightarrow E$]

In cases 1 and 2, beliefs $Bel_{e1}(isValid(c_1))$ and $Bel_{e1}(isValid(c_2))$ are deduced from the trust reasoning process, and these deduced beliefs will be entered into the agent's belief set with their labels, i.e. labels of beliefs $Bel_{e1}(isValid(c_1))$ and $Bel_{e1}(isValid(c_2))$ will be (3, (2, $Bel - Nec$), (7, 8), On), and (15, (11, 14, =>E), {}, On) respectively.

Case 3: Agent e_1 received certificate c_3 as a message from agent e_3.

16. $isValid(c_1) \wedge (I(c_3) = S(c_1)) \wedge isSigned(c_3, PK(c_1))$ [from 2, P1-2, and P2-2 with $\wedge I$]
17. $\exists c'((isValid(c')) \wedge (I(c_3) = S(c')) \wedge (isSigned(c_3, PK(c'))) \Rightarrow isValid(Sig(c_3))$ [Substitute c_3 for c in PKI2]
18. $isValid(Sig(c_3))$ [from 16 and 17 with $\Rightarrow E$]
19. $isValid(Sig(c_3)) \wedge (DS(c_3) \leq today) \wedge (today < DE(c_3)) \wedge \neg inCRL(c_3)$ [from 18 and P5-2, P6-2, and P7-1 with $\wedge I$]
20. $isValid(Sig(c_3)) \wedge (DS(c_3) \leq today) \wedge (today < DE(c_3)) \wedge \neg inCRL(c_3) \Rightarrow isValid(c_3)$ [Substitute c_3 for c in PKI3]
21. $isValid(c_3)$ [Deduced from 19 and 20 with $\Rightarrow E$]
22. $Inf_{e3,e1}(A) \Rightarrow Bel_{e3}(A)$ [from P3-2 and ERcL1 with $\Rightarrow E$]
23. $Bel_{e1}(isValid(c_3)$ [from P4 and 22 with $\Rightarrow E$]

In case 3, $Bel_{e1}(isValid(c_3))$ is deduced from the trust reasoning process, and deduced belief will be entered into the agent's belief set with its respective label (23, (P4, 21, =>E), {}, On).

Case 4: Agent e_1 received a message about the certificate c_1 from agent e_4

24. $Inf_{e4,e1}(\neg isValid(c_1)) \Rightarrow \neg isValid(c_1)$ [from P4-1, ERcL2 with $\Rightarrow E$]
25. $\neg isValid(c_1)$ [from P3-3, 25]
26. $Bel_{e1}(\neg isValid(c_1))$ [from 25 with $Bel - Nec$]

In case 4, $Bel_{e1}(\neg isValid(c_1))$ is deduced, and deduced belief will be entered into the agent's belief set with its respective label (26, (25, =>E), {}, On).

4.4. Revision process under the belief revision mechanism

Belief set of agent e_1 represented as BS_{e1}={}. Initially, the belief set will be empty as $BS_{e1} = \phi$. Based on the current scope of study beliefs can be obtained in two ways, i) A belief can be received as a message from other agents in the domain; ii) A belief can be derived as a deduced belief from the trust reasoning process, i.e., change in trust relationship deduces different reasoning results. So, until four beliefs are part of the agent belief set. Currently,agente_1 belief set has Bel_{e1}={$Bel_{e1}(isValid(c_1))$,$Bel_{e1}(isValid(c_2))$,$Bel_{e1}(isValid(c_3)$, $Bel_{e1}(\neg isValid(c_1))$}.

Beliefs are retained in the agent's belief set with their labels which helps to maintain the derivation path.Entries of other beliefs are handled in a similar manner. Now the belief set of agent e1 consists of two contradictory beliefs along with their labels. i.e., $Bel_{e1}(isValid(c_1))$ and $Bel_{e1}(\neg isValid(c_1))$. So, the revision process in section 3.0.3 will be triggered to retract the contradictory belief. If belief $Bel_{e1}(isValid(c_1))$ is selected as discussed in point 2 of section 3.0.3, then the revision procedure forward chains through to-lists, changing the status of deduced belief at 7, and 8 from *on* to *off*. To this point, beliefs $Bel_{e1}(isValid(c_1))$,$Bel_{e1}(isValid(c_2))$ will have their statuses *off*, leaving BS_{e1}={$Bel_{e1}(isValid(c_3)$, $Bel_{e1}(\neg isValid(c_1))$} in belief set of agent$e_1$. Using this method, agents would retain their beliefs, but their status would be set to*off*. As a result, it will be possible to trace the beliefs, but at the same time prevent the agent from re-acquiring them, therefore making belief revisions a practical, and useful process.

5. APPLICATION OF THE BELIEF REVISION MECHANISM IN SPY NOVEL

5.1. Spy novel scenario

We consider another scenario from [8] in which multiple agents exchange messages with each other as an information source.

We consider three agents a_1, b_1, and c_1 who are interested in exchanging information about the two facts "there is a spy in the train T", denoted by p_1, and "the train T has arrived at the railway station", denoted by q. In this situation agent a_1 trusts b_1 in regard to his validity for p_1, and in regard to his sincerity for q_1, and a_1 trusts c_1 in regard to his completeness for q_1. a_1 trust may be supported, for instance, by the fact that b belongs to some intelligence service, and c_1 is an employee of the railway station who stands on the platform where the train is supposed to arrive. In this situation, b_1 has informed a_1 of information p_1, and he has also informed q_1, and c_1 has not informed a_1 of information q_1.The formalization of the above scenario is as follows:

5.2. Formalization

Individual variables:

- *agents: a,b,c*
- *facts: p, q*

Individual constants:

- *agents: a1,b1,c1*
- *facts: p1, q1*

Predicates:

- *isFact(x): x is a fact.*

Empirical and logical theories

We can assume the following theories.
IS1: $TR(a_1,b_1,validity)$(Agent a_1 trusts b_1 in his validity)

IS2: $TR(a_1,b_1,sincerity)$(Agent a_1 trusts b_1 in his sincerity)
IS3: $TR(a_1,c_1,completeness)$(Agent a_1 trusts c_1 completeness)
IS3-1: $TR(a_1,c_1,sincerity)$ *(Agent* a_1 trusts c_1 sincerity)
IS4: $Inf_{b1,a1}(isFact(p_1))$(b_1 has informed to a_1 about $isFact(p_1)$)
IS5: $\neg Inf_{c1,a1}(isFact(q_1))$($c_1$ has not informed to a_1 about $isFact(q_1)$)
IS6: $Inf_{c1,a1}(\neg isFact(q_1))$($c_1$ has informed to a_1 about $\neg isFact(q_1)$)
IS7: $\neg Inf_{b1,a1}(isFact(q_1))$($b_1$ has not informed to a_1 about $isFact(q_1)$)

5.3. Trust reasoning process

From the above formalization, empirical and logical theories obtained as logical formulas will be used in the trust reasoning process

.

Case 1: Agent a_1 received information about p_1 as a message from agent b_1.

1. $Inf_{b1,a1}(isFact(p_1)) \Rightarrow isFact(p_1)$ [from $IS1$ and $ERcL2$ with ⇒E]
2. $isFact(p_1)$[from IS4 and 1 with ⇒E]
3. $Bel_{a1}(isFact(p_1))$[from 2withBel –Nec]

After deduction, we have $Bel_{a1}(isFact(p_1))$. The deduced belief will be added to the belief set of agents a_1 with its respective label (3, (2, Bel – Nec), {11}, On).

Case 2: Agent a_1 received information about q_1 as a message from agent c_1.

4. $Bel_{a1}(\neg Inf_{c1,a1}(isFact(q_1)))$[from IS5 with$BEL$ –Nec]
5. $A \Rightarrow Inf_{c1,a1}(A)$ [from IS3 and ERcL6 with ⇒ E]
6. $isFact(q_1) \Rightarrow Inf_{c1,a1}(isFact(q_1))$ [from 5]
7. $\neg Inf_{c1,a1}(isFact(q_1) \Rightarrow \neg isFact(q_1)$ [*contraposition* of 6]
8. $Bel_{a1}(\neg Inf_{c1,a1}(isFact(q_1) \Rightarrow \neg isFact(q_1))$ [from 7 with BEL – Nec]
9. $Bel_{a1}(\neg Inf_{c1,a1}(isFact(q_1)) \Rightarrow Bel_{a1}(\neg isFact(q_1))$ [from 8 with BEL]
10. $Bel_{a1}(\neg isFact(q_1))$ [from 4and 9with ⇒ E]
11. $Bel_{a1}(isFact(p_1) \wedge \neg isFact(q_1))$ [from 3and10 with ∧I]

After deduction we have $Bel_{a1}(isFact(p_1) \wedge \neg isFact(q_1))$. The deduced belief will be added to the belief set of agents a_1 with its respective label (11, (3, 10, ∧I), {}, On).

Case 3: Agent a_1 received information about p_1 as a message from agent $c_1$with a change in a trust relationship.

12. $Inf_{b1,a1}(A) \Rightarrow Bel_{b1}(A)$ [from IS2 and ERcL1 with ⇒E]
13. $Inf_{b1,a1}(isFact(p_1)) \Rightarrow Bel_{b1}(isFact(p_1))$ [from 12]
14. $Bel_{b1}(isFact(p_1))$[from IS4 and 13 with ⇒E]
15. $Bel_{a1}(Inf_{c1,a1}(\neg isFact(q_1))$ [from IS6withBel –Nec]
16. $Inf_{c1,a1}(A) \Rightarrow Bel_{c1}(A)$ [from IS3-1 and ERcL1 with ⇒E]
17. $Inf_{c1,a1}(\neg isFact(p_1)) \Rightarrow Bel_{c1}(\neg isFact(p_1))$ [from 16]
18. $Bel_{c1}(\neg isFact(p_1))$ [from IS6 and 17 with ⇒E]
19. $Bel_{a1}(Bel_{c1}(\neg isFact(p_1))$ [from 18with BEL – Nec]
20. $Bel_{a1}(Bel_{b1}(isFact(p_1)))$ [from 14 with BEL – Nec]
21. $Bel_{a1}(Bel_{b1}(isFact(p_1) \wedge Bel_{c1}(\neg isFact(p_1))$ [from 19 and 20 with ∧I]

After the trust reasoning process, $Bel_{a1}(Bel_{b1}(isFact(p_1) \wedge Bel_{c1}(\neg isFact(p_1))$ has been deduced. The deduced result will be added to the belief set of agent a_1 with its respective label (21, (19, 20, $\wedge I$), {}, On). Change in a trust relationship from completeness to sincerity between agent a_1 trusts c_1 deduces different reasoning results $Bel_{a1}(isFact(p_1) \wedge \neg isFact(q_1))$, and $Bel_{a1}(Bel_{b1}(isFact(p_1) \wedge Bel_{c1}(\neg isFact(p_1))$ respectively. Therefore, it is evident from the deduced results that a change in trust relationships leads to different deduced results.

5.4. Revision process under the belief revision mechanism

Initially, the belief set of agents a_1 is empty $BS_{a1} = \phi$. After the reasoning process, the belief set of agent a_1 will include deduced beliefs, i.e., BS_{a1}={$Bel_{a1}(isFact(p_1) \wedge \neg isFact(q_1))$,$Bel_{a1}(Bel_{b1}(isFact(p_1) \wedge Bel_{c1}(\neg isFact(p_1))$}.As discussed before, a belief can be obtained as a message from another agent in the domain, or it can be derived through the trust reasoning process. So, in the current scenario, if we consider receiving a belief as a message from other agents, and it contradicts the existing beliefs of the agent's a_1 belief set BS_{e1} then the revision process discussed in section 3.0.3 will be triggered to retract the contradictory belief. If belief $Bel_{a1}(isFact(p_1) \wedge \neg isFact(q_1))$ is selected, then the revision procedure forward chains through to, and from lists, changing the status of belief from *on* to *off*. To this point, the contradictory belief causing inconsistency will have their statuses both subsequent beliefs will have their statuses *off*, leaving $Bel_{a1}(Bel_{b1}(isFact(p_1) \wedge Bel_{c1}(\neg isFact(p_1))$ in the belief set of agent a_1. Using this method, agents would retain their beliefs, but their status would be set to *off*. As a result, it will be possible to trace the beliefs, but at the same time prevent the agent from re-acquiring them. Thus, the resulting belief set is consistent.

6. CONCLUDING REMARKS

In this paper, we presented a belief revision mechanism with trust reasoning based on extended reciprocal logic (ERL) for multi-agent systems. A single mechanism that includes trust reasoning, and belief revision for the decision-making process of an agent in multi-agent systems. Trust reasoning based on ERL is used for the deduction process because extended reciprocal logic is a suitable logic system underlying trust reasoning. As a result, an agent maintains its belief set. If a contradiction occurs in the agent's belief set, a revision process based on Doyle's procedural approach is triggered. Doyle's procedural approach uses the concept of derivation path which allows forward, and backtracking to track beliefs that cause inconsistency in the agent's belief set. Furthermore, we demonstrated the application of the belief revision mechanism in the field of public key infrastructure PKI. A unique feature of the belief revision mechanism is that it is based on extended reciprocal logic, which makes it a general mechanism. As part of future work, we will demonstrate the application of the belief revision mechanism in other areas as well.

REFERENCES

[1] J. Cheng, "Reciprocal Logic: Logics for Specifying, Verifying, and Reasoning About Reciprocal Relationships," Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, vol. 3682, pp. 437–445, 2005. doi:10.1007/11552451_58

[2] F. Lévy, "A Survey of Belief Revision and Updating in Classical Logic," International Journal ofIntelligent Systems, vol. 9, no. 1, pp. 29–59, 1994. doi:10.1002/int.4550090104

[3] P. Gärdenfors and H. Rott, "Belief Revision," in Handbook of Logic in Artificial Intelligence and LogicProgramming, vol. 4, D. M. Gabbay, C. J. Hogger, and J. A. Robinson, Eds. Oxford University Press, pp. 35–132, 1995.

[4] L. Sombé, "A Glance at Revision and Updating in Knowledge Bases," International Journal of Intelligent Systems, vol. 9, no. 1, pp. 1–27, 1994, doi:10.1002/int.4550090103

[5] J. Cheng, "A Strong Relevant Logic Model of Epistemic Processes in Scientific Discovery," in Information Modelling and Knowledge Bases XI, E. Kawaguchi, H. Kangassalo, H. Jaakkola, and I. A. Hamid, Eds. Frontiers in Artificial Intelligence and Applications, vol. 61, IOS Press, pp. 136–159, 2000

[6] J. Doyle, "A Truth Maintenance System," Artificial Intelligence, vol. 12, no. 3, pp. 231–272, 1979. doi:10.1016/0004-3702(79)90008-0

[7] W. Zhao, V. Varadharajan, and G. Bryan, "Modelling Trust Relationships in Distributed Environments," Trust and Privacy in Digital Business, Lecture Notes in Computer Science, vol. 3184, pp. 40–49, 2004. doi:10.1007/978-3-540-30079-3_5

[8] R. Demolombe, "Reasoning About Trust: A Formal Logical Framework," Trust Management, LectureNotes in Computer Science, vol. 2995, pp. 291–303, 2004. doi:10.1007/978-3-540-24747-0_22

[9] R. Anderson and N. D. Belnap Jr., "Entailment: the Logic of Relevance and Necessity," vol. 1,Princeton University Press, 1975.

[10] R. Anderson, N. D. Belnap Jr., and J. M. Dunn "Entailment: the Logic of Relevance and Necessity,"vol. 2, Princeton University Press, 1992.

[11] S. Basit and Y. Goto, "An Extension of Reciprocal Logics for Trust Reasoning," Intelligent Information and Database Systems, Lecture Notes in Computer Science, vol. 12034, pp. 65–75, 2020, doi:10.1007/ 978-3-030-42058-1_6

[12] S. Basit and Y. Goto, "An Extension of Reciprocal Logic for Trust Reasoning: A Case Study in PKI,"Intelligent Information and Database Systems, Lecture Notes in Computer Science, vol. 13757, pp. 496–506, 2022. doi:10.1007/978-3-031-21743-2_39

[13] S. Ustymenko and D. G. Schwartz, "Dynamic Agent-oriented Reasoning about Belief and Trust,"Multiagent and Grid Systems, vol. 4, no. 3, pp. 335–346, 2008, doi:10.3233/mgs-2008-4307

[14] D. G. Schwartz, "Nonmonotonic Reasoning as a Temporal Activity," Proceedings of the 15th International Workshop on Non-Monotonic Reasoning, 2014. doi:10.48550/arXiv.1404.7173

AUTHORS

Sameera Basit received her bachelor's degree in Information Technology from Punjab University College of Information Technology, and her master's degree in Computer Science from the Lahore College for Women, University. She is currently a doctoral course student at Saitama University, Japan. Her research interests include intelligent agents, trust in Multi-Agent Systems, trust reasoning, agent decision-making, logic, and software engineering.

Yuichi Goto received the degree of Bachelor of Engineering in computer science, the degree of Master of Engineering in computer science, and the degree of Doctor of Engineering in computer science from Saitama University in 2001, 2003, and 2005, respectively. His current research interests include relevant reasoning and its applications, automated theorem finding, epistemic programming, anticipatory reasoning reacting systems, and Web services.