Towards Enforcement of Confidentiality in Agent Interactions

Joachim Biskup and **Gabriele Kern-Isberner** and **Matthias Thimm**
Faculty of Computer Science
Technische Universität Dortmund, Germany

Abstract

A lot of work has been done on interactions and negotiations in multi agent systems. In this paper, we introduce new aspects of security into this scenario by dealing with confidentiality preservation. As we assume the agents to be capable of reasoning, we also have to take inferences of conveyed information into account. This problem has been addressed by *controlled query evaluation (CQE)* in the security community. We present a conceptual integration of CQE techniques into a BDI inspired agent model that allows agents to explicitly handle confidentiality preserving concerns when interacting with other agents in a multi agent system. We illustrate our ideas on an example of distributed meeting scheduling.

Introduction

Negotiation between agents involves exchange of information and persuasion in order to reach an agreement. In this paper we are concerned with security aspects and especially with confidentiality preservation aspects of negotiations between agents in multi agent systems which provide rich application scenarios for problems dealing with confidentiality and availability of information. There is much work on formalizing negotiation scenarios using multi agent systems, see for example (Kraus 2001; Karunatillake *et al.* 2005; Rahwan, Sonenberg, & Dignum 2003; Booth 2002). But very little work has been done in security aspects for multi agent systems, most of this work handling secure communication and authentication problems, e. g. (Winslett 2003; Poslad, Charlton, & Calisti 2003; Sierra *et al.* 2003; Boulosa *et al.* 2006). We are addressing the problem of confidentiality preservation under the inference problem (Farkas & Jajodia 2002) in the sense of (Biskup & Bonatti 2004; Biskup & Weibert 2007a; 2007b) for multi agent systems, which is a special case of the general "information hiding" (Halpern & O'Neill 2003; Hughes & Shmatikov 2002). When agents interact with other agents in a multi agent system, pieces of information can be exchanged and by using inference techniques, more information can be obtained. An agent might not be fully aware of the conclusions a second agent can infer from the given information, and thus the second agent may be able to derive information he is not allowed to know.

Controlled query evaluation (Biskup & Bonatti 2004; Biskup & Weibert 2007a; 2007b) is a formal approach to determine the conclusions an agent can infer if provided with a specific information and to check whether the disclosure of said information violates confidentiality. We will present a declarative concept of confidentiality preserving negotiations in multi agent scenarios that is based on quite a general type of negotiation acts. Apart from defining which agent conveys to which other agent which piece of information, we also include interactions concerning justifications that often prove essential for reaching a negotiation goal. We link our ideas to standard agent models by extending the BDI model (Weiss 1999) by components for controlled query evaluation, thus taking first steps towards an operational framework. Our approach is illustrated by the problem of distributed meeting scheduling (Garrido, Brena, & Sycara 1996).

This paper is structured as follows: First, we give a brief overview on controlled query evaluation. Then we identify the problems of confidentiality preservation in agent interactions and introduce our running example. We continue with a formal description of our framework and sketch an agent model that summarizes the formal functionalities in an abstract manner afterwards. We conclude with comparisons to other works and an outlook on future work.

Controlled Query Evaluation

Controlled Query Evaluation (CQE) is an approach for preservation of confidential information in interactions between an information system and its users (Sichermann, de Jonge, & van de Riet 1983; Bonatti, Kraus, & Subrahmanian 1995). Each user of the system might have some restrictions on the information he is allowed to obtain from the system. Ordinary database systems restrict permissions using static access rights, but thus suffer from the inference problem (Farkas & Jajodia 2002). A malicious user can outsmart such a system by exploiting the inference problem, given he has some knowledge about the structure of the data. This is illustrated in the following example, which is taken from (Biskup *et al.* 2007).

Example 1. Suppose a database stores information about employees and their salary but must not disseminate information about the specific salary of a specific employee. If using static access rights the two pieces of information "Alice is a manager" and "a manager's salary is $ 50,000" might appear harmless; but if one combines them they imply the information "Alice's salary is $ 50,000" which should be kept secret.

CQE is a dynamic approach to overcome the inference problem by dynamically checking the user's knowledge to ensure that he can not derive information he is not allowed to know. By distorting answers to queries by either lying or refusing to answer, it has been shown that confidential information can be kept secret in many different scenarios (Biskup & Bonatti 2004; Biskup & Weibert 2007b). What follows is a brief introduction to CQE terminology that will suffice for our needs.

Confidentiality Policies

Let $\mathcal{L}$ be a propositional language. Although $\mathcal{L}$ is propositional we will also use predicates and rule schemata and assume that $\mathcal{L}$ is properly grounded, i. e., all rule schemata are replaced by all their instantiations using the constants appearing in $\mathcal{L}$. When considering subsets of $\mathcal{L}$ we assume these subsets to be consistent when not mentioned otherwise. For a set S we denote with $\mathfrak{P}(S)$ the power set of S.

Let $\phi \subseteq \mathcal{L}$ be a finite set of sentences and $\alpha \in \mathcal{L}$ a sentence. The *evaluation* of α in ϕ is either `true` (iff α can be inferred: $\phi \vdash \alpha$), `false` (iff $\neg\alpha$ can be inferred: $\phi \vdash \neg\alpha$) or `unknown` (else). We consider queries addressing the information system ϕ, that consist only of a sentence $\alpha \in \mathcal{L}$ and awaits an answer of the form `true`, `false` or `unknown` with the obvious meaning.

At all times the system keeps a log of the user's knowledge, denoted `log`, that consists of the assumed a priori knowledge of the user and is updated with every answer the system gives to that particular user. The actual controlled query evaluation consists of two steps: first a *censor* checks whether the actual evaluation of a query, or a possible conclusion of it, together with the user's assumed knowledge violate confidentiality, and then, if necessary, a *modificator* distorts the answer somehow so that the distorted answer can be given to the user without violating confidentiality.

To represent a confidentiality policy we use the notion of *confidentiality targets*. A confidentiality target consists of the sentence to be protected and a set of truth values the user should not infer.

Definition 1 (Confidentiality target, confidentiality policy)**.** A *confidentiality target* is a pair $\langle \psi, V \rangle$ with $\psi \in \mathcal{L}$ and $V \subset \{\texttt{true}, \texttt{false}, \texttt{unknown}\}$ and $\emptyset \neq V \neq \{\texttt{true}, \texttt{false}, \texttt{unknown}\}$. A *confidentiality policy* is a finite set of confidentiality targets.

Example 2. The confidentiality target $\langle a, \{\texttt{true}, \texttt{false}\}\rangle$ defines that the user is not allowed to infer that a is either `true` or `false` (whether this coincides with the information system's actual evaluation of a or not).

The specification of CQE does not make sense, when the user already knows a confidential piece of information in his a priori knowledge $\texttt{log}_0$. So it is reasonable to assume that the user's a priori knowledge does not already violate confidentiality in the first place. This condition and all other possible restrictions on the use of the system are formalized as a precondition $precond$. In general, for a finite set of sentences $\phi \subseteq \mathcal{L}$, the user's assumed a priori knowledge $\texttt{log}_0$ and a confidentiality policy $policy$, the tuple $(\phi, \texttt{log}_0, policy)$ has to *satisfy* a given precondition $precond$ in order to use the system.

Preserving confidentiality

The approach of CQE is described via a CQE-function that basically maps a sequence of queries to a sequence of answers. As a side effect, the function updates the maintained user log appropriately with the newly acquired knowledge of the user. The formal definition of *confidentiality preserving* is expressed in terms of an indistinguishability property, roughly saying that for all reasonable situations, including all possible sequences of queries, the user cannot distinguish the actual information system instance from an alternative instance in which the evaluation of a target-sentence ψ is not in the corresponding target-set V.

A CQE-function can preserve confidentiality by either lying, i. e., providing a false answer to a query, or refusing to answer at all, i. e., returning a special answer value `refuse`. Nonetheless, such a CQE-function should also provide a maximal availability towards the user, i. e., it should distort as few answers as possible in order to preserve confidentiality.

Example 3. We continue Example 1. Suppose the user has the a priori knowledge $\texttt{log}_0 = \{m(X) \wedge mSalary(Y) \Rightarrow salary(X, Y)\}$ which says that "If X is a manager and a manager's salary is Y, then X's salary is Y". Furthermore the information system ϕ is given by $\phi = \{m(alice), mSalary(50000)\}$. The information about Alice's salary must be preserved in ϕ as given by the confidentiality policy $\texttt{conf} = \{\langle salary(alice, 50000), \{\texttt{true}\}\rangle\}$. Suppose the user asks ϕ for the evaluation of $m(alice)$. The information system can answer this query truthfully with `true` as confidentiality is still preserved when this information is disclosed. The CQE method then adds $m(alice)$ to the user's knowledge yielding $\texttt{log}_1 = \texttt{log}_0 \cup \{m(alice)\}$. When the user now asks ϕ about the evaluation of $mSalary(50000)$, the information system must not answer `true` because then the user can infer the confidential piece of information $salary(alice, 50000)$. Therefore ϕ must lie by answering either `false` or `unknown` or refuse to answer.

Operational frameworks for CQE have been investigated in many different scenarios, see e. g. (Biskup & Weibert 2007b).

Confidentiality in Multi Agent Systems

Although CQE is developed mostly for interactions between an information system and its users, preservation of confidentiality is needed in many different situations. Further-

more in the standard CQE scenarios confidentiality is only the concern of the information system. When considering multi agent systems, and especially multi agent systems performing negotiation, preservation of confidentiality is a bilateral concern of all agents. Although the agents are willing to participate in a negotiation, they also might have secrets they do not want to disclose to other agents participating in that negotiation (Winslett 2003).

To illustrate our ideas of a confidentiality preserving multi agent system, we use the problem of meeting scheduling as an example (Garrido, Brena, & Sycara 1996). Although this problem has already been studied under privacy issues in, for example, (Wallace & Freuder 2002), we pursue a more sophisticated approach. While in those papers the agents attempt to preserve privacy and thus confidentiality by only disseminating as little information as possible to achieve a successful negotiation, we aim at implementing well-known methods for CQE directly into the agent.

The problem of meeting scheduling as it fits to our purposes is described as follows.[1] We consider a set of n different agents, each of them has its own calendar of the week, consisting of six days from Monday to Saturday, with eight time slots of one hour each day. Every agent may already have some time slots filled with appointments and some agents may share the same appointments. The "negotiation goal" of the n agents is to determine a specific time slot for a new appointment, such that every agent can attend to that appointment. To reach this goal the agents can exchange information by querying other agents about their calendars and other beliefs, give proposals for the new appointment, and agree as well as reject given proposals. Agents may change their beliefs over time and abandon other appointments in order to reach an agreement. As negotiation involves persuasion and argumentation (Parsons, Sierra, & Jennings 1998; Kraus, Nirkhe, & Sycara 1993; Karunatillake *et al.* 2005) we also enable the agents to ask for justifications for other agents' beliefs.

The confidentiality issues of an agent in this scenario can be of different kinds. Suppose the agents are in an employer/employee relationship, then the employee surely wants to hide information about spare time activities or the attendance to a strike commission. Furthermore the employer may also want to hide information about spare time activities but also about the existence of a job interview for a possible replacement for the said employee. In general we consider the following four different privacy issues as relevant for the problem of meeting scheduling: 1.) the date/time of a specific appointment, 2.) whether a specific agents attends a specific appointment or not, 3.) the existence of an appointment at a specific date/time and 4.) whether an agent is busy at a specific date/time or not.

The Declarative Concept

In this section we develop a declarative view of a multi agent system with negotiating and confidentiality preserving agents. Our approach is inspired by the work of Kraus in (Kraus 2001) but due to space restrictions we only give a short overview of our ideas. Therefore we do not provide a formal definition of the semantics, but give some examples to illustrate the integration of confidentiality preservation into an agent model.

Let $\mathfrak{A}$ be a set of agent identifiers $\mathfrak{A} = \{a_1, \ldots, a_n\}$. The agents negotiate on a given *negotiation goal* NG which is a subset of $\mathcal{L}$. NG specifies the search space for possible solutions.

Example 4. Suppose $\mathcal{L}$ contains grounded predicates of the form $\texttt{daytime}(AP, D, TS, TE)$ where AP denotes an appointment, D is the day and TS resp. TE is the start resp. end time. Then the meaning of the instance $\texttt{daytime}(staff\_meeting, monday, 12, 13)$ is "The staff meeting takes place on Monday between 12 and 13". Suppose a group of agents wants to negotiate about the day and time for a project meeting with a duration of one hour. Then the corresponding negotiation goal $\mathsf{NG} \subseteq \mathcal{L}$ is

$$\begin{aligned}\mathsf{NG} \;=\; \{&\texttt{daytime}(proj\_meeting, monday, 8, 9),\\ &\texttt{daytime}(proj\_meeting, monday, 9, 10), \ldots\}\end{aligned}$$

We abbreviate the above negotiation goal with $\texttt{daytime}(project\_meeting, X, Y, Y{+}1)$.

Given a negotiation goal NG the task of the agents is to determine a sentence $\mathsf{ng} \in \mathsf{NG}$ which can be mutually accepted by all agents.

We continue our development by defining the possible *negotiation acts*, i. e., the possible actions an agent can undertake in this system. As in the previous section we only consider simple yes/no/unknown-queries. But to enable the agents to use refusal as possible distortion method for modification of confidentiality violating information, let $\Theta = \{\texttt{true}, \texttt{false}, \texttt{unknown}, \texttt{refuse}\}$ be the set of possible answers to a query. To define the possible actions of an agent in our system, we define the set Ψ of negotiation (speech) acts as follows.

Definition 2 (Negotiation Acts)**.** The set of *negotiation acts* Ψ is the minimal set containing the following:

- For every $A \in \mathfrak{A}$, $\{A_1, \ldots, A_k\} \subseteq \mathfrak{A}$, $\alpha \in \mathcal{L}$ and $x \in \{\texttt{true}, \texttt{false}, \texttt{unknown}\}$ it is $\langle A : \mathsf{inform}\ \{A_1, \ldots, A_k\}\ \alpha\ x\rangle \in \Psi$ with the meaning: A tells $A_1, \ldots, A_k$ that his evaluation of α is x.
- For every $A_1, A_2 \in \mathfrak{A}$, $\alpha \in \mathcal{L}$ it is $\langle A_1 : \mathsf{query}\ A_2\ \alpha\rangle \in \Psi$ with the meaning: A_1 asks A_2 for his evaluation of α.
- For every $A_1, A_2 \in \mathfrak{A}$, $\alpha \in \mathcal{L}$, $x \in \Theta$ it is $\langle A_1 : \mathsf{answer}\ A_2\ \alpha\ x\rangle \in \Psi$ with the meaning: the answer of A_1 to A_2 regarding α is x.
- For every $A \in \mathfrak{A}$ it is $\langle A : \mathsf{abandon}\rangle \in \Psi$ with the meaning: A abandons the current negotiation.
- For every $A \in \mathfrak{A}$, $\alpha \in \mathcal{L}$ it is $\langle A : \mathsf{propose}\ \alpha\rangle \in \Psi$ with the meaning: A proposes α as a solution for the negotiation.
- For every $A_1, A_2 \in \mathfrak{A}$, $\alpha \in \mathcal{L}$ and $x \in \{\texttt{true}, \texttt{false}\}$ it is $\langle A_1 : \mathsf{justify}\ A_2\ \alpha\ x\rangle \in \Psi$ with the meaning: A_1 asks A_2 to justify that α has the evaluation x.

[1] We simplify the problem of meeting scheduling in comparison to (Wallace & Freuder 2002) by omitting the locations of the meetings.

- For every $A_1, A_2 \in \mathfrak{A}$, $\alpha \in \mathcal{L}, \Phi \subseteq \mathcal{L}$ and $x \in \{\texttt{true}, \texttt{false}\}$ it is $\langle A_1 : \text{justification } A_2\ \alpha\ x\ \Phi\rangle \in \Psi$ with the meaning: A_1 justifies that α has the evaluation x towards A_2 with Φ.
- It is $\circ \in \Psi$ which denotes the "empty" action.

We restrain our presentation of these negotiation acts on the syntactic representation above and omit definitions of semantics.

Let Σ denote the set of all *negotiation sequences*, i. e., all ordered tuples $(\tau_1, \ldots, \tau_l)$ with $\tau_1, \ldots, \tau_l \in \Psi$, $l \in \mathbb{N}$ and $\diamond$ denoting the empty negotiation sequence. We use the element operator $\in$ with the usual meaning also on negotiation sequences, i. e., it is $\tau \in (\tau_1, \ldots, \tau_l)$ iff $\tau \in \{\tau_1, \ldots, \tau_l\}$.

Negotiating agents

We assume a suitable BDI architecture (Weiss 1999) as a basis for an agent. In this section we focus on modeling beliefs and confidentiality issues before proposing a complete suitable BDI-inspired agent model in the next section.

An agent's beliefs will comprise beliefs about himself and the current state of the world as well as his view of the beliefs of other agents (Kraus, Nirkhe, & Sycara 1993). Furthermore we explicitly represent an agent's confidentiality policy as a separate piece of the agent's belief. As information may be confidential differently with respect to the different agents, we extend the definition of confidentiality target appropriately.

Definition 3 (Personalized confidentiality target, personalized confidentiality policy)**.** A *personalized confidentiality target* is a triple $\langle A', \psi, V\rangle$ with $A' \in \mathfrak{A}$, $\psi \in \mathcal{L}$ and $V \subset \{\texttt{true}, \texttt{false}, \texttt{unknown}\}$ and $\emptyset \neq V \neq \{\texttt{true}, \texttt{false}, \texttt{unknown}\}$. A *personalized confidentiality policy* is a finite set of personalized confidentiality targets.

The first component of a personalized confidentiality target is the subject for this target, i. e., the agent from whom the piece of information, that ψ has a truth-value in V, should be kept secret.

Example 5. Let e_1 and b_1 be agents. Suppose e_1 wants to hide from b_1 the information that he attends the appointment scm (*strike committee meeting*). This can be represented as the personalized confidentiality target

$$\langle b_1, \texttt{attends}(e_1, scm), \{\texttt{true}\}\rangle$$

being part of e_1's confidentiality policy.

Example 6. Let a_1 and b_1 be agents. Suppose agent a_1 wants agent b_1 to know nothing definite about the date and time of an appointment m_1. This can be represented as (for all possible X, Y, Z)

$$\langle b_1, \texttt{daytime}(m_1, X, Y, Z), \{\texttt{true}, \texttt{false}\}\rangle$$

being part of a_1's confidentiality policy. , which is an abbreviation for

$$\langle b_1, \texttt{daytime}(m_1, monday, 8, 9), \{\texttt{true}, \texttt{false}\}\rangle,$$
$$\langle b_1, \texttt{daytime}(m_1, monday, 9, 10), \{\texttt{true}, \texttt{false}\}\rangle$$
$$\ldots$$

Observe that the above personalized confidentiality target also prohibits the disclosure of information that might not be true in the current belief of an agent. Given adequate background constraints as "One appointment can not take place at two different times" one of the above (sub-)targets necessarily has to be false in the agent's belief. But the above target states that b_1 must know nothing definite about the day and time of m_1 and so he should also not believe a wrong date for it.

When agents gather new information about other agents by observing a negotiation act, they have to incorporate this new information into their individual belief and their beliefs about other agents, using belief operations as revision or update (Krümpelmann *et al.* 2008). Thus the beliefs of an agent have to be represented with respect to a given sequence σ of hitherto executed negotiation acts.

Furthermore, we improve our underlying framework of propositional logic in two ways.

First, as agents may have different beliefs about the world and especially about the beliefs of other agents, the propositional language $\mathcal{L}$ is not expressive enough to capture these needs. We therefore extend the propositional language $\mathcal{L}$ by introducing a family of modal operators $\mathcal{B}_X$ that read "Agent X believes…" as in epistemic logic with a standard Kripke-style semantics, yielding an extended language $\mathcal{L}^{\mathcal{B}}$ with $\mathcal{L} \subseteq \mathcal{L}^{\mathcal{B}}$. We assume the standard properties for $\mathcal{B}_X$ and refer to (Fagin *et al.* 2003) for a full axiomatization. With the use of these modal operators, we can represent agents' beliefs about other agents and the agents can reason about other agents' beliefs. Therefore, let $\models^{\mathcal{B}}$ denote an appropriate inference relation for $\mathcal{L}^{\mathcal{B}}$.

Definition 4 (Beliefs)**.** The *beliefs* $\texttt{bel}_A^\sigma$ of an agent A after the negotiation sequence σ is a tuple

$$\texttt{bel}_A^\sigma = (\texttt{is}_A^\sigma, \texttt{conf}_A^\sigma, \{\texttt{view}_{A,A_1}^\sigma, \ldots, \texttt{view}_{A,A_m}^\sigma\})$$

with individual belief $\texttt{is}_A^\sigma \subseteq \mathcal{L}^{\mathcal{B}}$, a personalized confidentiality policy $\texttt{conf}_A^\sigma$, and views, i. e., the beliefs of agent A about the beliefs of agents A_i, $\texttt{view}_{A,A_1}^\sigma, \ldots, \texttt{view}_{A,A_m}^\sigma \subseteq \mathcal{L}^{\mathcal{B}}$. The a priori belief of the agent is denoted $\texttt{bel}_A^\diamond$.

Second, given the current beliefs $\texttt{bel}_A^\sigma$ and observing a negotiation act τ, or even participating in it, an agent A needs to process the new information in order to derive the new beliefs $\texttt{bel}_A^{\sigma+\tau}$, where $+$ denotes concatenation. Such derivations might just apply propositional or modal logic inferences, or suitable combinations of these with more sophisticated and generally non-monotonic techniques like belief operations, which are not the topic of this discussion, see e. g. (Krümpelmann *et al.* 2008; Kern-Isberner 2001) for more information.

As a simple example of deriving a new belief, let agent A perform the negotiation act $\langle A : \text{inform } \{A'\}\ \alpha\ \{\texttt{true}\}\rangle$ after the negotiation sequence σ resulting in a negotiation sequence σ'. If A' now derives by means of some appropriate belief operations, that A truly believes in α to be true, then this negotiation act results in $\mathcal{B}_A\alpha \in \texttt{view}_{A',A}^{\sigma'}$.

The above definition also offers the option to change the personalized confidentiality policy $\texttt{conf}_A^\sigma$ during a negotiation process. In fact, to guarantee a successful negotiation it

might be necessary for the agent to abandon some of his personalized confidentiality targets in order to reach an agreement as the abandonment of beliefs in general is a crucial issue in non-trivial negotiations (Zhang *et al.* 2004). Nevertheless the change of confidentiality policies is an open research problem for ordinary CQE as well, so assume that $\mathtt{conf}^{\sigma}_{A} = \mathtt{conf}^{\diamond}_{A}$ for all σ and A.

To initiate a negotiation between several agents, the agents have to accept a previously determined negotiation goal as a precondition. Although the process of determining an acceptable negotiation goal can itself be seen as a negotiation we do not formalize this process but require the agents to be willing to participate in a negotiation for a given negotiation goal.

Definition 5 (Acceptance function)**.** An *acceptance function* accept_A for an agent A is a function $\mathsf{accept}_A : \mathfrak{P}(\mathcal{L}) \rightarrow \{\mathtt{true}, \mathtt{false}\}$.

An agent A accepts a subset NG of $\mathcal{L}$ as a negotiation goal, if $\mathsf{accept}_A(\mathsf{NG}) = \mathtt{true}$. We call a set of agents $\mathfrak{A}$ *unwilling* if there is no negotiation goal NG such that for all $A \in \mathfrak{A}$ it holds $\mathsf{accept}_A(NG) = \mathtt{true}$. We only consider sets of agents $\mathfrak{A}$ that are willing to participate in a negotiation.

Once the negotiation goal is determined, the agents start exchanging information using the possible negotiation acts in Ψ. We introduce a simple function that determines the best next action in the current situation (possibly the empty action which is denoted by $\circ$).

Definition 6 (Action function)**.** An *action function* action_A for agent A is a function $\mathsf{action}_A : \Sigma \rightarrow \Psi$.

After incorporating new information into their beliefs, the agents' attitudes towards proposals of other agents may change. A negotiation ends when all agents agree to a given proposal ng in the solution space of the given negotiation goal NG.

Definition 7 (Agreement function)**.** An *agreement function* agree_A for an agent A is a function $\mathsf{agree}_A : \Sigma \times \mathcal{L} \rightarrow \{\mathtt{true}, \mathtt{false}\}$.

An agent A *agrees* to a proposal $\mathsf{ng} \in \mathsf{NG} \subseteq \mathcal{L}$ after exchanging some information during a sequence σ, if it holds $\mathsf{agree}_A(\sigma, \mathsf{ng}) = \mathtt{true}$. If $\mathsf{agree}_A(\sigma, \mathsf{ng}) = \mathtt{false}$, A *rejects* the proposal.

With the use of the above functions we can describe the final product of a negotiation in our approach. We call a negotiation sequence σ *semi-complete*, if it is intuitively well-formed. That means for example, that every query and every call for justification is answered, that there are no answers without a query, and so on.

Definition 8 (Negotiation Product)**.** A sentence ng is a *negotiation product* with respect to $\mathfrak{A}$, iff there exists a negotiation goal NG, such that

1. $\mathsf{ng} \in \mathsf{NG}$,
2. for all $A \in \mathfrak{A}$ it holds $\mathsf{accept}_A(NG) = \mathtt{true}$ and
3. there exists a semi-complete negotiation sequence σ for $\mathfrak{A}$ and NG, such that
   (a) $\langle A : \mathsf{propose\ ng} \rangle \in \sigma$ for some $A \in \mathfrak{A}$ and
   (b) for all $A \in \mathfrak{A}$ it holds $\mathsf{agree}_A(\sigma, \mathsf{ng}) = \mathtt{true}$.

We say that a negotiation sequence σ *failed*, if $\langle A : \mathsf{abandon} \rangle \in \sigma$ for a $A \in \mathfrak{A}$.

Preserving confidentiality in agent interactions

The confidentiality features are given on a declarative and an operational layer.

On a declarative layer, a formal definition for "confidentiality preserving" basically requires the following: If an agent is not allowed to learn some piece of information, then the agent's particular view on the behaviour of the overall system, applying for all possible initializations and all possible action sequences of the system, should never leave the agent with a belief that the said piece of information holds.

On an operational layer a control component censors each planned individual action for potential harmful consequences and possibly modifies the plan appropriately. We give some ideas on the operationalizing of the following declarative concept in the next section.

Clearly, the basic challenges are to design a policy control, i. e., censors and modificators necessarily operating action-wise, such that these mechanisms provably achieve the confidentiality goal declaratively expressed referring to all possible initializations and action sequences. These challenges are well-known to be highly demanding, see, e. g., the rich work on noninterference (Goquen & Mesequer 1982; Mantel 2001) or on cryptographic protocols (Goldreich 2004). Accordingly, in this paper we can only sketch our general approach to provide a tentative solution for negotiation in multi agent systems.

Given a suitable definition of "indistinguishability of situations", we propose the following (rough) generic outline for the declarative layer (as above let $precond$ expresses that confidentiality is not violated in the first place):

Definition 9 (Mutual confidentiality preservation)**.** The agents $A_1, \ldots, A_n$ *mutually preserve confidentiality* iff it holds

for all negotiation goals NG,
for all "actual situations", i. e.,
  for all initial a priori beliefs $\mathtt{bel}^{\diamond}_{A_1}, \ldots, \mathtt{bel}^{\diamond}_{A_n}$,
  for all negotiation sequences σ,
for all personalized confidentiality targets
  $\langle A_j, \psi, V \rangle \in \mathtt{conf}^{\sigma}_{A_i}$ for some i, j

where $(\mathsf{NG}, \{\mathtt{bel}^{\diamond}_{A_1}, \ldots, \mathtt{bel}^{\diamond}_{A_n}\})$ satisfies $precond$

there exists an "alternative situation", i. e.,
  there exist alternative a priori beliefs $\Delta^{\diamond}_{A_1}, \ldots, \Delta^{\diamond}_{A_n}$
    such that $(\mathsf{NG}, \{\Delta^{\diamond}_{A_1}, \ldots, \Delta^{\diamond}_{A_n}\})$
    satisfies $precond$, and
  there exists an alternative negotiation sequence σ',

such that the following two conditions are met:

1. The actual situation as given above and the alternative situation as postulated are indistinguishable from the point of view of agent A_j.
2. From the point of view of agent A_j in the alternative situation, the evaluation of ψ is not in V.

Towards an Operational Framework

We now give some ideas on how to operationalize the declarative layer from Definition 9 on confidentiality preservation in agent interactions, i. e., we propose a method that an agent can use to satisfy the above given security requirements for some exemplary cases.

Censoring and modification

Whenever an agent is about to execute an action (which is equivalent to the disclosure of information), he has to check whether confidentiality will be preserved after having executed said action. He does so by simulating the derivation methods that would (presumably) be applied by the other agents, when observing said action. Here we assume, that an agent has complete knowledge about the deriving methods of other agents and about their background knowledge. So the agent is capable of checking whether an action will violate confidentiality at any time. This check is accomplished by the censor of the agent which prevents the agent to disclose confidential information.

Definition 10 (Censor)**.** The *censor function* $\mathsf{violates}_A$ for an agent A is a function $\mathsf{violates}_A : \Sigma \times \Psi \rightarrow \{\mathtt{true}, \mathtt{false}\}$.

The censor evaluates the action under consideration regarding A's confidentiality policy $\mathtt{conf}_A^\sigma$ at sequence σ. The definition of $\mathsf{violates}_A$ depends on the type of action. Due to lack of space we do not give a full definition of $\mathsf{violates}_A$ for all types of actions but only some examples for necessary conditions for $\mathsf{violates}_A$ to be true in order meet the declarative definition of confidentiality preservation above (Definition 9). For an action $\tau = \langle A : \mathsf{inform}\ \{A'\}\ \alpha\ \mathtt{true}\rangle$ and a sequence σ it is $\mathsf{violates}_A(\sigma, \tau) = \mathtt{true}$ if

$$\exists\, \langle A', \psi, \{\mathtt{true}\}\rangle \in \mathtt{conf}_A^\sigma : \mathtt{view}_{A,A'}^\sigma \cup \{\alpha\} \models^{\mathcal{B}} \psi$$
$$\vee\ \exists\, \langle A', \psi, \{\mathtt{false}\}\rangle \in \mathtt{conf}_A^\sigma : \mathtt{view}_{A,A'}^\sigma \cup \{\alpha\} \models^{\mathcal{B}} \neg\psi$$
$$\vee\ \ldots$$

The definition of $\mathsf{violates}_A$ is the same as above for the action type answer. Interestingly, even a query can violate confidentiality.

Example 7. The question "Are you busy on Wednesday at 12?" provides several pieces of information about the questioner. First the respondent can infer, that the questioner does not know what the respondent does on Wednesday at 12[2] and second, that the questioner himself is assumably not busy on Wednesday at 12.

Thus for an action $\tau = \langle A : \mathsf{query}\ A'\ \alpha\rangle$ and a sequence σ it is $\mathsf{violates}_A(\sigma, \tau) = \mathtt{true}$ if

$$\exists\, \langle A', \psi, \{\mathtt{true}\}\rangle \in \mathtt{conf}_A^\sigma : \mathtt{view}_{A,A'}^\sigma \cup \{\neg\mathcal{B}_A\alpha, \neg\mathcal{B}_A\neg\alpha\} \models^{\mathcal{B}} \psi$$
$$\vee\ \exists\, \langle A', \psi, \{\mathtt{false}\}\rangle \in \mathtt{conf}_A^\sigma : \mathtt{view}_{A,A'}^\sigma \cup \{\neg\mathcal{B}_A\alpha, \neg\mathcal{B}_A\neg\alpha\} \models^{\mathcal{B}} \neg\psi$$
$$\vee\ \ldots$$

[2]We assume that agents do only perform these queries if they do not know the answer.

Remember that $\mathcal{B}_A$ is the modal operator that reads "Agent A believes...".

As mentioned before, agents can either use lying, refusal or a combination of both to distort information, such that confidentiality is preserved. When defining the censor function $\mathsf{violates}_A$, one has to consider the actual distortion method to be used in order to actually preserve confidentiality. If the actual distortion method is lying, then the censor must not only prohibit, that any individual confidential piece of information is preserved, but the disjunction of all confidential pieces of information (Bonatti, Kraus, & Subrahmanian 1995).

Example 8. Suppose that $\mathtt{conf}_A^\sigma$ only consists of confidentiality targets regarding agent A' with the evaluation "`true`" being the only confidential evaluation, i. e. $\mathtt{conf}_A^\sigma = \{\langle A', \psi_1, \{\mathtt{true}\}\rangle, \ldots, \langle A', \psi_l, \{\mathtt{true}\}\rangle\}$. If A' already knows, that $\psi_1 \vee \ldots \vee \psi_l$ must be true, then the sequence of queries for ψ_1 to ψ_l results in an inconsistent view of the beliefs of agent A', because the query for every ψ_i must be answered with `false`.

Thus, for the confidentiality policy given in Example 8, an action $\tau = \langle A : \mathsf{inform}\ \{A'\}\ \alpha\ \mathtt{true}\rangle$ and a sequence σ it is not sufficient to define $\mathsf{violates}_A(\sigma, \tau) = \mathtt{true}$ if

$$\exists\, i \in \{1, \ldots, l\} : \mathtt{view}_{A,A'}^\sigma \cup \{\alpha\} \models^{\mathcal{B}} \psi_i$$

but necessary to define $\mathsf{violates}_A(\sigma, \tau) = \mathtt{true}$ if

$$\mathtt{view}_{A,A'}^\sigma \cup \{\alpha\} \models^{\mathcal{B}} \psi_1 \vee \ldots \vee \psi_l$$

Furthermore, if the actual distortion method is refusal, then the censor must also take the possibility for meta inference into account.

Example 9. Let "The evaluation of α is `true`" be a confidential piece of information. Suppose Agent A' believes, that α must be either `true` or `false`, and that agent A' is fully aware of how agent A distorts answers to preserve confidentiality. Assume α is actually `false` for agent A and agent A' asks A about the truth-value of α. Then A truthfully returns the answer "The evaluation of α is `false`" as it does not violate confidentiality. But suppose now, that α is actually `true` for agent A and A' asks the same question. Now A must refuse to answer, in order to preserve confidentiality. But now A' can infer, that α must be `true` for A, because if α would have been `false` for A, then A had not refused to answer. To overcome this problem A must refuse to answer the query about α in any case, so that A' can not distinguish these two cases.

Suppose now that $\mathsf{violates}_A$ is properly defined and handles the above mentioned security problems accordingly. Then $\mathsf{violates}_A$ restrains the action function action_A of an agent A by ensuring the following constraint:

$$\textbf{IF}\ \mathsf{action}_A(\sigma) = \tau \quad \textbf{THEN} \quad \mathsf{violates}_A(\sigma, \tau) = \mathtt{false}$$

As agreeing and rejecting a proposal ng is equivalent to informing all agents about ng or $\neg$ng, the censor $\mathsf{violates}_A$ restrains the agreement function agree_A of an agent A by ensuring the following constraint:

$$\textbf{IF}\ \mathsf{agree}_A(\sigma, \alpha) = x\ \textbf{THEN}$$
$$\mathsf{violates}_A(\sigma, \langle A : \mathsf{inform}\ \mathfrak{A}\ \alpha\ x\rangle) = \mathtt{false}$$

If an action endangers confidentiality, the agent has to choose another action to execute. In the case of actions of the type inform, abandon, query, justify this can be realized by executing no action at all, as no other agent expects a particularly action from the first agent. But if one or more agents expect an action, either an answer to a query or a justification for a belief, the agent must produce an alternative answer that preserves confidentiality. The same is true for the agreement function of an agent, but as there are only two values possible, preservation can only be achieved by setting $\mathsf{agree}_A(\sigma, \alpha) = \texttt{false}$ if $\mathsf{agree}_A(\sigma, \alpha) = \texttt{true}$ violates confidentiality and vice versa[3]. In the case of answers to queries, the agent has the additional options to answer with `unknown` or to refuse to answer at all. Here standard CQE methods can be used to determine the best alternative answer (Biskup & Bonatti 2004; Biskup & Weibert 2007b).

Confidentiality preserving issues regarding disclosure of justifications have not been investigated in CQE so far. When the true justification for a belief violates confidentiality, many solutions to modify the answer are possible. The agent can make up a new justification, present another one that does not violate confidentiality or refuse to justify at all. But as we only want to formalize a general framework for enforcement of confidentiality between agents in this paper, we do not discuss the matter here and leave it open for future research. We conclude this section with an example that illustrates the above definitions.

Example 10. We continue our example of meeting scheduling. Suppose agent e_1 features the personalized confidentiality policy $\texttt{conf}^{\sigma}_{e_1}$ after a negotiation sequence σ with

$$\texttt{conf}^{\sigma}_{e_1} = \{\langle b_1, \texttt{attends}(e_1, scm), \{\texttt{true}\}\rangle\}$$

and does truly attend the strike committee meeting: $\texttt{attends}(e_1, scm) \in \texttt{is}^{\sigma}_{A}$. Furthermore e_1 has a pretty good clue that b_1 knows that there is a strike committee meeting being held on Wednesday at 12 to 13 and that if someone is busy at that time, he assumably attends said meeting. So e_1's view of b_1's beliefs includes

$$\begin{aligned}\texttt{view}^{\sigma}_{e_1,b_1} \supseteq \{&\texttt{daytime}(scm, wednesday, 12, 13),\\ &\texttt{daytime}(scm, X, Y, Z) \wedge \texttt{busy}(E, X, Y, Z)\\ &\Rightarrow \texttt{attends}(E, scm))\}\end{aligned}$$

Let $\langle b_1 : \texttt{query}\ e_1\ \texttt{busy}(e_1, wednesday, 12, 13)\rangle$ be the last action of the sequence σ, i. e. b_1 asks e_1 whether he is busy on Wednesday at 12 to 13. Then e_1 must not answer this query truthfully, because we have $\mathsf{violates}_A(\sigma, \langle e_1 : \mathsf{answer}\ b_1\ \texttt{busy}(e_1, wednesday, 12, 13)\ \texttt{true}\rangle) = \texttt{true}$ due to

$$\begin{aligned}\texttt{view}^{\sigma}_{e_1,b_1} \cup \{\texttt{busy}(e_1, wednesday, 12, 13)\} \models^{\mathcal{B}}\\ \texttt{attends}(e_1, scm).\end{aligned}$$

Therefore confidentiality is at risk and e_1 must alter his answer in order to preserve confidentiality. It is reasonable to assume that an agent knows whether he is busy at a given time or not. Especially e_1 must assume that b_1 thinks so. So it is for every X, Y, Z:

$$\mathcal{B}_{e_1} busy(e_1, X, Y, Z) \vee \mathcal{B}_{e_1} \neg busy(e_1, X, Y, Z) \in \texttt{view}^{\sigma}_{e_1,b_1}$$

Due to this constraint e_1 can not undertake $\chi = \langle e_1 : \mathsf{answer}\ b_1\ \texttt{busy}(e_1, wednesday, 12, 13)\ \texttt{unknown}\rangle$ as next action, because this would result in the sentence

$$\neg\mathcal{B}_{e_1} busy(e_1, X, Y, Z) \wedge \neg\mathcal{B}_{e_1} \neg busy(e_1, X, Y, Z)$$

to be incorporated into $\texttt{view}^{\sigma}_{e_1,b_1}$ and therefore leads to an inconsistency. So we have $\mathsf{violates}_A(\sigma, \chi) = \texttt{true}$, because every confidential piece of information can be inferred from $\texttt{view}^{\sigma}_{e_1,b_1}$ and the above piece of information. It follows that e_1 can only answer `false` or refuse to answer at all. Given that e_1 and b_1 are in an employee/employer relationship, refusal does not seem appropriate, so the answer `false` is the best choice for agent e_1.

Confidentiality preservation for BDI agents

The above developed framework summarizes the essential aspects of a negotiation in a formal but nonetheless mostly declarative manner. A fully featured model of multi agent negotiation needs among other things also to comprise decision making processes (Kraus 2001) and belief operations (Alchourrón, Gärdenfors, & Makinson 1985; Kern-Isberner 2001; Booth 2002; Krümpelmann *et al.* 2008). In this section we only give a brief overview about the model of a negotiating and confidentiality preserving agent in an abstract manner. Our agent model incorporates standard BDI architecture in order to represent a rational and autonomous agent (Rao & Georgeff 1995; Weiss 1999). BDI stands for Beliefs, Desires and Intentions and a BDI architecture separates the logical model of an agent into these three areas.

Figure 1 shows an abstract view of our negotiating and confidentiality preserving agent which incorporates both BDI as well as CQE techniques. However, in our model the beliefs component is explicitly divided into the beliefs of the agent about the world and himself (`is`), the beliefs about other agents (`view`) and a confidentiality policy (`conf`) as developed in our formal framework in the previous section. Furthermore the agent itself is divided into an active part (upper half) and a reactive part (lower half) which cooperate in a parallel mode; information flow is depicted with dashed lines, while action flow with solid lines. As in the BDI model developed in (Weiss 1999) the agents continuously evaluate the current state of the world, generate possible options for the next actions, and filter the best options using their beliefs, some underlying preferences (not depicted in the figure), their desires (`des`) and their intentions (`int`). Thereupon the best options are furthermore evaluated in the sense of confidentiality preservation by an agent's `policy control`. If an intention can be selected to be performed, the necessary actions are executed as depicted in Figure 1. Newly acquired information must be incorporated into the beliefs of the agent using revision and update techniques (Krümpelmann *et al.* 2008). As in the active part, also in the reactive part the preservation of confidentiality

[3] We disregard the case that both values violate confidentiality.

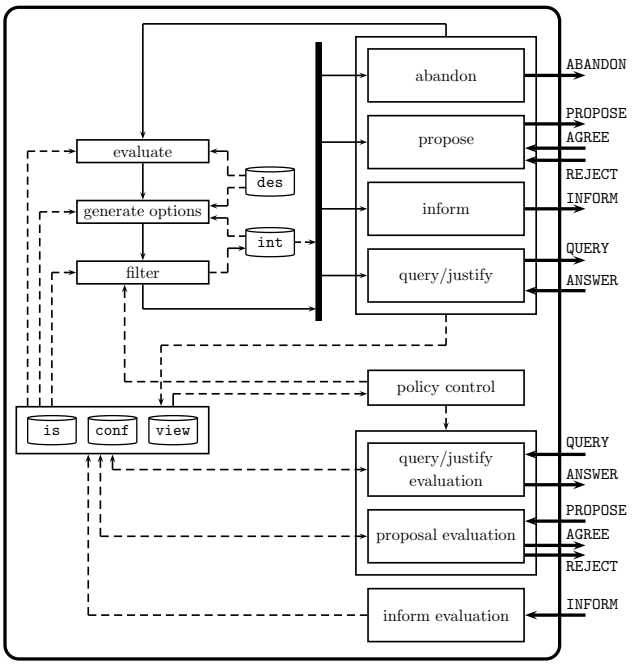

Figure 1: A negotiating agent

is handled by the component `policy control`, which prohibits the dissemination of confidential information. Whenever the agent needs to reply to a general query, a query for justification or a proposal, the `policy control` checks whether confidentiality is preserved and eventually alters the intended reply.

Related Work

Frameworks for distributed negotiation have been investigated very broadly so far, see (Rueda, Garcia, & Simari 2002; Kraus 2001; Karunatillake *et al.* 2005; Rahwan, Sonenberg, & Dignum 2003) for some examples. In this paper we do not intend to neglect this huge body of work but to add the new feature of confidentiality preservation. In fact it should be investigated if confidentiality preservation can be modularized and integrated into existing frameworks and implementations for negotiation.

The example of meeting scheduling as a negotiation problem has been elaborated before (Garrido, Brena, & Sycara 1996; Wallace & Freuder 2002). In (Wallace & Freuder 2002) also security issues are raised. However, in contrast to our approach, the preservation of confidentiality there is handled in a very simple manner as the agents only aim at disseminating as little information about themselves as necessary but do not consider confidentiality targets. The dissemination of as less information as possible can also be achieved in our approach with a suitable representation of the BDI core of the agent. When restraining the agent to act only passively in the environment so that he only reacts on queries, he does not disclose any other information. However, in a negotiation scenario this is not a desirable feature for all agents as then no negotiation will succeed as no action takes place; so a compromise between these two requirements can be made by adjusting the preferences of the agents accordingly. However, with the use of methods for CQE the agents do have a better formal representation of how to preserve their privacy. We therefore consider in our framework a lot more confidentiality problems as in (Wallace & Freuder 2002).

When talking about negotiation, another important aspect is argumentation. Argumentation theory has become a very active field of research and many proposals exist for introducing argumentative capabilities into negotiation systems (Amgoud, Dimopolous, & Moraitis 2007; Bench-Capon 2003; Rueda, Garcia, & Simari 2002; Karunatillake *et al.* 2005; Thimm & Kern-Isberner 2008). In our framework we provide a declarative support for handling argumentation in a multi agent system with the action types justify and justification. However, future research includes the evaluation of argumentation formalism regarding our perspective of privacy and confidentiality issues.

Conclusion and future work

In this paper we presented a formal approach to adapt methods for CQE for the use in multi agent systems. CQE has some history in scientific research but has not been adapted for the use in multi agent systems so far. We are currently developing a full framework for handling negotiation, belief and confidentiality issues that bases on the approach proposed in this paper. Although the work presented here is just preliminary, the developed framework provides a solid basis for future work. As mentioned above this includes the exploration of argumentation formalism as well as an adaption of techniques for CQE for more sophisticated approaches of knowledge representation. More precisely, the fitness of CQE techniques for non-monotonic representation formalism has not yet been investigated. As default logics are common representation formalisms, an adaption of CQE techniques for these is mandatory. Furthermore, as persuasion (Bench-Capon 2003) is a fundamental aspect of negotiation, agents must have the ability to abandon specific confidentiality targets in order to reach agreements (Biskup *et al.* 2007). Also the adaption of other agents' confidentiality targets must be taken into account in order to ensure effective confidentiality handling.

Another main concern in CQE is the warranty of availability. This means that, although an agent must preserve confidentiality, he is also committed to provide as much useful information as possible. An agent can be equipped with an availability policy as well as a confidentiality policy. In the employer/employee example the employee might be committed to provide the employer with any information concerning a specific project, even if this violates confidentiality. The discrepancy between these two requirements has to be handled by the agent appropriately.

Acknowledgments We thank the reviewers for their helpful comments to improve the original version of this paper.

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and

revision functions. *Journal of Symb. Logic* 50(2):510–530.

Amgoud, L.; Dimopolous, Y.; and Moraitis, P. 2007. A unified and general framework for argumentation-based negotiation. In *Proc. of AAMAS'07*.

Bench-Capon, T. 2003. Persuasion in practical argument using value based argumentation frameworks. *Journal of Logic and Computation* 13(3):429–448.

Biskup, J., and Bonatti, P. 2004. Controlled query evaluation for enforcing confidentiality in complete information systems. *Int. Journal of Information Security* 3(1):14–27.

Biskup, J., and Weibert, T. 2007a. Confidentiality policies for controlled query evaluation. In *Proceedings of the 21th IFIP WG11.3 Working Conference on Data and Applications Security, LNCS 4602*, 1–13. Springer.

Biskup, J., and Weibert, T. 2007b. Keeping secrets in incomplete databases. *Int. Journal of Information Security* online first.

Biskup, J.; Burgard, D. M.; Weibert, T.; and Wiese, L. 2007. Inference control in logic databases as a constraint satisfaction problem. In *Proc. of the Third International Conference on Information Systems Security*, 128–142.

Bonatti, P. A.; Kraus, S.; and Subrahmanian, V. S. 1995. Foundations of secure deductive databases. *IEEE Transactions on Knowledge and Data Engineering* 7:406–422.

Booth, R. 2002. Social contraction and belief negotiation. In *Proc. of the 8th Conference on Principles of Knowledge Representation and Reasoning*, 375–384.

Boulosa, M.; Caib, Q.; Padgetc, J. A.; and Rushton, G. 2006. Using software agents to preserve individual health data confidentiality in micro-scale geographical analyses. *Journal of Biomedical Informatics* 39(2):160–170.

Fagin, R.; Halpern, J.; Moses, Y.; and Vardi, M. 2003. *Reasoning about Knowledge*. MIT Press.

Farkas, C., and Jajodia, S. 2002. The inference problem: a survey. *ACM SIGKDD Explorations Newsletter* 4:6–11.

Garrido, L.; Brena, R.; and Sycara, K. 1996. Cognitive modeling and group adaptation in intelligent multi-agent meeting scheduling. In *First Iberoamerican Workshop on DAI and MAS*, 55–72.

Goldreich, O. 2004. *Foundations of Cryptography II – Basic Applications*. Cambridge University Press.

Goquen, J. A., and Mesequer, J. 1982. Security policies and security models. In *Proc. IEEE Symposium on Security and Privacy*, 11–22.

Halpern, J., and O'Neill, K. 2003. Anonymity and information hiding in multiagent systems. In *Proc. of the 16th IEEE Comp. Sec. Foundations Workshop 2003*, 75 – 88.

Hughes, D., and Shmatikov, V. 2002. Information hiding, anonymity and privacy: A modular approach. *Journal of Computer Security* 12(1):3–36.

Karunatillake, N. C.; Jennings, N. R.; Rahwan, I.; and Norman, T. J. 2005. Argument-based negotiation in a social context. In *Proc. of AAMAS'05*, 1331–1332.

Kern-Isberner, G. 2001. *Conditionals in nonmonotonic reasoning and belief revision*. Number 2087 in Lecture Notes in Computer Science. Springer.

Kraus, S.; Nirkhe, M.; and Sycara, K. P. 1993. Reaching agreements through argumentation: a logical model and implementation. In *Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*, 233–247.

Kraus, S. 2001. Automated negotiation and decision making in multiagent environments. In *Selected Tutorial Papers from the 9th ECCAI Advanced Course ACAI 2001 and Agent Link's 3rd European Agent Systems Summer School on Multi-Agent Systems and Applications*, 150–172.

Krümpelmann, P.; Thimm, M.; Ritterskamp, M.; and Kern-Isberner, G. 2008. Belief operations for motivated BDI agents. In *Proceedings of AAMAS'08*.

Mantel, H. 2001. Preserving information flow properties under refinement. In *Proc. IEEE Symposium on Security and Privacy*, 78–91.

Parsons, S.; Sierra, C.; and Jennings, N. 1998. Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8(3):261–292.

Poslad, S.; Charlton, P.; and Calisti, M. 2003. Specifying standard security mechanisms in multi-agent systems. In *Trust, Reputation, and Security: Theories and Practice*, volume 2631 of *LNCS*. Springer. 227–237.

Rahwan, I.; Sonenberg, L.; and Dignum, F. 2003. Towards interest-based negotiation. In *Proc. of the 2nd Int. Conf on Autonomous Agents and Multi-Agent Systems*, 773–780.

Rao, A. S., and Georgeff, M. P. 1995. BDI-agents: from theory to practice. In *Proceedings of the First Intl. Conference on Multiagent Systems*, 312–319.

Rueda, S. V.; Garcia, A.; and Simari, G. R. 2002. Argument-based negotiation among bdi agents. *Journal of Computer Science and Technology* 2(7):1–8.

Sichermann, G. L.; de Jonge, W.; and van de Riet, R. P. 1983. Answering queries without revealing secrets. *ACM Transactions on Database Systems* 8:41–59.

Sierra, J. M.; Hernández, J. C.; Ponce, E.; and Ribagorda, A. 2003. Protection of multiagent systems. In *Computational Science and its Applications*, 984–990.

Thimm, M., and Kern-Isberner, G. 2008. A distributed argumentation framework using defeasible logic programming. In *Proc. of COMMA'08*, 381–392.

Wallace, R., and Freuder, E. 2002. Constraint-based multiagent meeting scheduling: effects of agent heterogeneity on performance and privacy loss. In *Proc. Workshop on DCR*, 176–182.

Weiss, G., ed. 1999. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press.

Winslett, M. 2003. An introduction to trust negotiation. In *iTrust 2003. Volume 2692 of Lecture Notes in Computer Science*. Springer-Verlag. 275–283.

Zhang, D.; Foo, N.; Meyer, T.; and Kwok, R. 2004. Negotiation as mutual belief revision. In *Proc. of the 19th Nat. Conf. on Artificial Intelligence (AAAI-04)*, 317–322.