Eliminating Opportunism using an Epistemic Mechanism

Jieting Luo*
Centrum Wiskunde & Informatica
Amsterdam, the Netherlands
jieting.luo@cwi.nl

Max Knobbout
Triple
Alkmaar, the Netherlands
m.knobbout@wearetriple.com

John-Jules Meyer
Utrecht University
Utrecht, the Netherlands
J.J.C.Meyer@uu.nl

ABSTRACT

Opportunism is a behavior that takes advantage of knowledge asymmetry and results in promoting agents' own value and demoting other agents' value. It is important to eliminate such a selfish behavior in multi-agent systems, as it has undesirable results for the participating agents. However, as the context we study here is multi-agent systems, system designers actually might not be aware of the value system for each agent thus they have no idea whether an agent will perform opportunistic behavior. Given this fact, this paper designs an epistemic mechanism to eliminate opportunism given a set of possible value systems for the participating agents: an agent's knowledge gets updated so that the other agent is not able to perform opportunistic behavior, and there exists a balance between eliminating opportunism and respecting agents' privacy.

KEYWORDS

Opportunism; Mechanism Design; Logic; Multi-agent Systems

ACM Reference Format:

Jieting Luo, Max Knobbout, and John-Jules Meyer. 2018. Eliminating Opportunism using an Epistemic Mechanism. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

Consider a scenario: A seller sells a cup to a buyer and it is known by the seller beforehand that the cup is actually broken. The buyer buys the cup without knowing it is broken. The seller exploits the knowledge asymmetry about the transaction to achieve his own gain at the expense of the buyer. Such behavior which is intentionally performed by the seller was named opportunistic behavior (or opportunism) by economist Williamson [15]. Opportunistic behavior is a behavior that takes advantage of relevant knowledge asymmetry and results in promoting an agent's own value and demoting another agent's value. On the one hand, it is common in distributed multi-agent systems that agents possess different knowledge, which enables the performance of opportunism; on the other hand, opportunistic behavior has undesirable results for other agents who participate in the system. Thus, we want to design mechanisms to eliminate opportunism. This paper investigates an epistemic mechanism, which allows us to eliminate the performance of opportunism in the system by revealing updates. In papers [6] [7], opportunism is monitored and predicted given a value system for an agent, i.e., an agent performed and will perform opportunistic behavior if he has the value system as we assume. However, as the context we study here is multi-agent systems, system designers might not be aware of the value system for each agent before designing any mechanism to eliminate opportunism in the system. The goal of this paper is thus to design mechanisms to eliminate opportunism given a set of possible value systems of agents, which contains the value systems with opportunistic propensity.

In mechanism design, a mechanism is an institute, procedure, or game for determining outcomes [8] [11]. Differently, we in this paper consider an operation to the system as an indirect mechanism: a revealing update that can eliminate opportunism through updating the knowledge of the agent. More precisely, we remove the precondition of opportunism (knowledge asymmetry) by revealing knowledge to agents such that agents will not be able to perform opportunistic behavior. Since agents' value systems are unknown to the system designer, there might exist privacy norms that prevent agents from having the knowledge for eliminating opportunism. We prove formal properties that allow us to check whether we can eliminate opportunism and respect agents' privacy as well.

2 FRAMEWORK

In this section, we introduce the model we use for multi-agent systems as in [7]. A transition system consists of agents, states of the world, actions, agents' epistemic accessibility relations, transitions which go from one state to another by an action, and a valuation function that returns for each state the properties of the environment.

Definition 2.1. Let $\Phi = \{p, q, \dots\}$ be a finite set of atomic propositional variables. A transition system over Φ is a tuple

$$\mathcal{T} = (Agt, S, Act, \pi, \mathcal{K}, \mathcal{R}, s_0)$$

where

- $Agt = \{1, \dots, n\}$ is a finite set of agents;
- S is a finite set of states;
- Act is a finite set of actions;
- $\pi : S \to 2^{\Phi}$ is a valuation function mapping a state to a set of propositions that are considered to hold in that state;
- $\mathcal{K} : Agt \to 2^{S \times S}$ is a function mapping an agent in Agt to a reflexive, transitive and symmetric binary relation between states; that is, given an agent i, for all $s \in S$ we have $s\mathcal{K}(i)s$; for all $s, t, u \in S$ $s\mathcal{K}(i)t$ and $t\mathcal{K}(i)u$ imply that $s\mathcal{K}(i)u$; and for all $s, t \in S$ $s\mathcal{K}(i)t$ implies $t\mathcal{K}(i)s$; $s\mathcal{K}(i)s'$ is interpreted as state s' is epistemically accessible from state s for agent i; we also use $\mathcal{K}(i, s) = \{s' \mid s\mathcal{K}(i)s'\}$ to denote the set of agent i's epistemically accessible states from state s;

*Corresponding. Work was partially done while the author was at Utrecht University.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

- $\mathcal{R} \subseteq S \times Act \times S$ is a relation between states with actions, which we refer to as the transition relation labeled with an action; we require that for all $s \in S$ there exists an action $a \in Act$ and one state $s' \in S$ such that $(s, a, s') \in \mathcal{R}$, and we ensure this by including a stuttering action sta that does not change the state, that is, $(s, sta, s) \in \mathcal{R}$; we restrict actions to be deterministic, that is, if $(s, a, s') \in \mathcal{R}$ and $(s, a, s'') \in \mathcal{R}$, then $s' = s''$; since actions are deterministic, sometimes we denote state s' as $s\langle a\rangle$ for which it holds that $(s, a, s\langle a\rangle) \in \mathcal{R}$; we use $Ac(s) = \{a \mid \exists s' \in S : (s, a, s') \in \mathcal{R}\}$ to denote the available actions in state s;
- $s_0 \in S$ denotes the initial state.

In the interest of simplicity, we only consider one action that takes place at a transition, thus the model is not concurrent.

Now we define the language we use. The language $\mathcal{L}_{\mathrm{KA}}$, propositional logic extended with knowledge and action modalities, is generated by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid K_i\varphi \mid \langle a\rangle\varphi \qquad (i \in Agt, a \in Act)$$

The semantics of $\mathcal{L}_{\mathrm{KA}}$ are defined with respect to the satisfaction relation $\models$. Given a transition system $\mathcal{T}$ and a state s in $\mathcal{T}$, a formula φ of the language can be evaluated as follows:

- $\mathcal{T}, s \models p$ iff $p \in \pi(s)$;
- $\mathcal{T}, s \models \neg\varphi$ iff $\mathcal{T}, s \not\models \varphi$;
- $\mathcal{T}, s \models \varphi_1 \vee \varphi_2$ iff $\mathcal{T}, s \models \varphi_1$ or $\mathcal{T}, s \models \varphi_2$;
- $\mathcal{T}, s \models K_i\varphi$ iff for all t such that $s\mathcal{K}(i)t$, $\mathcal{T}, t \models \varphi$;
- $\mathcal{T}, s \models \langle a\rangle\varphi$ iff there exists s' such that $(s, a, s') \in \mathcal{R}$ and $\mathcal{T}, s' \models \varphi$;

Other classical logic connectives (e.g.,"$\wedge$", "$\rightarrow$") are assumed to be defined as abbreviations by using $\neg$ and $\vee$ in the conventional manner. As standard, we write $\mathcal{T} \models \varphi$ if $\mathcal{T}, s \models \varphi$ for all $s \in S$, and $\models \varphi$ if $\mathcal{T} \models \varphi$ for all multi-agent systems $\mathcal{T}$. Notice that we can also interpret $\langle a\rangle\varphi$ as the ability to achieve φ by action a. Hence, we write $\neg\langle a\rangle\varphi$ to mean not being able to achieve φ by action a. In addition of the $\mathcal{K}$-relation being S5, we also place restrictions of *no-forgetting* and *no-learning* based on Moore's work [9] [10] to simplify our model. It is specified as follows: given a state s in S, if there exists s' such that $s\langle a\rangle\mathcal{K}(i)s'$ holds, then there is a s'' such that $s\mathcal{K}(i)s''$ and $s' = s''\langle a\rangle$ hold; if there exists s' and s'' such that $s\mathcal{K}(i)s'$ and $s'' = s'\langle a\rangle$ hold, then $s\langle a\rangle\mathcal{K}(i)s''$. Following this restriction, we have

$$\models K_i(\langle a\rangle\varphi) \leftrightarrow \langle a\rangle K_i\varphi.$$

In other words, if an agent has knowledge about the effect of an action, he will not forget about it after performing the action; and the agent will not gain extra knowledge about the effect of an action after performing the action.

One important feature of opportunism is that it promotes agents' own value but demotes others' value. In this section we will specify agents' value system, as it is the standard of agents' consideration about the performance of opportunistic behavior. A value can be seen as an abstract standard according to which agents have their preferences over states. For instance, if we have a value denoting *equality*, we prefer the states where equal sharing or equal rewarding hold. Related work about values can be found in [12] and [13]. Because of the abstract feature of a value, it is usually interpreted in more detail as a state property, which is represented as a $\mathcal{L}_{\mathrm{KA}}$ formula. The most basic value we can construct is simply a proposition p, which represents the value of achieving p. More complex values can be interpreted such as $K\varphi$, meaning that it is valuable to *achieve knowledge*. More examples can be found in [7].

We argue that agents can always compare any two values, as we can combine two equivalent values as one value. Thus, we then define a value system as a total order (representing the degree of importance) over a set of values, which means that agents can always compare any two values. In other words, every element in the set of values is comparable to each other and none of them is logically equivalent to each other. One can see similar approaches in [3] for the definition of preferences and [1] for the definition of goals.

Definition 2.2 (Value System). A value system $V = (\mathrm{Val}, \prec)$ is a tuple consisting of a finite set $\mathrm{Val} = \{v, ..., v'\} \subseteq \mathcal{L}_{\mathrm{KA}}$ of values together with a strict total ordering $\prec$ over Val. When $v \prec v'$, we say that value v' is more important than value v as interpreted by value system V. A value system profile $(V_1, V_2, ..., V_{Agt})$ is a tuple containing a value system V_i for each agent i.

We also use a natural number indexing notation to extract the value of a value system, so if we have the ordering $v \prec v' \prec \ldots$ for a value system V, then $V[0] = v$, $V[1] = v'$, and so on. Note that different agents may or may not have different value systems. We now define a multi-agent system as a transition system together with agents' value systems. Formally, a multi-agent system $\mathcal{M}$ is an $(n + 1)$-tuple:

$$\mathcal{M} = (\mathcal{T}, V_1, ..., V_n),$$

where $\mathcal{T}$ is a transition system, and for each agent i in $\mathcal{T}$, V_i is a value system.

We now define agents' preferences over two states in terms of values, which will be used for modeling agents' decision making and the effect of opportunism. We first define how a value gets promoted and demoted along a state transition:

Definition 2.3 (Value Promotion and Demotion). Given a value v and an action a, we define the following shorthand formulas:

$$\mathrm{promoted}(v, a) := \neg v \wedge \langle a\rangle v$$
$$\mathrm{demoted}(v, a) := v \wedge \langle a\rangle\neg v.$$

We say that a value v is promoted along the state transition (s, a, s') if and only if $s \models \mathrm{promoted}(v, a)$, and we say that v is demoted along this transition if and only if $s \models \mathrm{demoted}(v, a)$.

An agent's value v gets promoted along the state transition (s, a, s') if and only if v doesn't hold in state s and holds in state s'; an agent's value v gets demoted along the state transition (s, a, s') if and only if v holds in state s and doesn't hold in state s'.

We secondly define a function $\mathrm{Mpreferred(i, s, s')}$ that maps a value system and two different states to the most preferred value that changes when going from state s to s' for agent i. In other words, it returns the value that the agent most cares about, i.e. the most important change between these states for the agent, and all the values that are more important than that value remain unchanged from state s to state s'.

Definition 2.4 (Most Preferred Value). Given a multi-agent system $\mathcal{M}$, an agent i and two states s and s', function Mpreferred : $Agt \times S \times S \longrightarrow$ Val is defined as follows:

$$\text{Mpreferred(i, s, s')}_{\mathcal{M}} := V_i[min\{j \mid \forall k > j : \mathcal{M}, s \models V_i[k] \Leftrightarrow \mathcal{M}, s' \models V_i[k]\}.]$$

We write Mpreferred(i, s, s′) for short if $\mathcal{M}$ is clear from context.

For example, given agent i's value system $u \prec v \prec w$, if formula $u, \neg v$ and $\neg w$ hold in state s and formula $u, v,$ and $\neg w$ hold in state s', function Mpreferred(i, s, s′) will return v. This is because value w remains the same in both states and value v changes from $\neg v$ to v. With this function we can define agents' preferences over two states. We use a binary relation "$\precsim$" over states to represent agents' preferences.

Definition 2.5 (State Preferences). Given a multi-agent system $\mathcal{M}$, an agent i and two states s and s', agent i weakly prefers state s' to state s, denoted as $s \precsim_i^{\mathcal{M}} s'$, iff

$$\mathcal{M}, s \models \text{Mpreferred(i, s, s')} \Rightarrow \mathcal{M}, s' \models \text{Mpreferred(i, s, s')}\,.$$

We write $s \precsim_i s'$ for short if $\mathcal{M}$ is clear from context. As standard, we also define $s \sim_i s'$ to mean $s \precsim_i s'$ and $s' \precsim_i s$, and $s \prec_i s'$ to mean $s \precsim_i s'$ and $s \nsim_i s'$. Moreover, we write $S \precsim_i S'$ for sets of states S and S' whenever $\forall s \in S, \forall s' \in S' : s \precsim s'$.

The intuitive meaning is that agent i weakly prefers state s' to s if and only if the agent's most preferred value does not get demoted (either stays the same or gets promoted). Using the same example for the illustration of function *Mpreferred*, given agent i's value system $u \prec v \prec w$, if formula $u, \neg v,$ and $\neg w$ hold in state s and formula $u, v,$ and $\neg w$ hold in state s', what the agent cares about is value v. Since it doesn't hold in state s but holds in state s', agent i will prefer state s' to state s. Clearly there is a correspondence between state preferences and value promotion or demotion by an action: given a model $\mathcal{M}$ with agent i, state s and available action a in s, and let $v^* = \text{Mpreferred(i, s, s}\langle\text{a}\rangle)$,

$$s \prec_i s\langle a\rangle \Leftrightarrow \mathcal{M}, s \models \text{promoted}(v^*, a)$$
$$s \succ_i s\langle a\rangle \Leftrightarrow \mathcal{M}, s \models \text{demoted}(v^*, a)$$
$$s \sim_i s\langle a\rangle \Leftrightarrow \mathcal{M}, s \models \neg(\text{demoted}(v^*, a) \vee \text{promoted}(v^*, a)).$$

One can refer to [7] for the proof. Moreover, the $\precsim_i$ relation is reflexive, transitive and total, which have been proved in [7]. It is possible that agents have different preferences over states, since they might not share the same value system.

Since we have already defined values and value systems as agents' basis for decision making, we can start to apply decision theory to reason about agents' decision-making. Given a state in the system, there are several actions available to an agent, and he has to choose one in order to go to the next state. Before choosing an action to perform, an agent must think about which actions are available to him. We have already seen that, for a given state s, the set of available actions is $Ac(s)$. However, since an agent only has partial knowledge about the state, we argue that the actions that an agent knows to be available is only part of the actions that are physically available to him in a state. For example, an agent can call a person if he knows the phone number of the person; without this knowledge, he is not able to do it, even though he is holding a phone. Recall that the set of states that agent i considers as being the actual state in state s is the set $\mathcal{K}(i, s)$. Given an agent's partial knowledge about a state as a precondition, he knows what actions he can perform in that state, which is the intersection of the sets of actions physically available in the states in this knowledge set.

Definition 2.6 (Subjectively Available Actions). Given an agent i and a state s, agent i's subjectively available actions are the set:

$$Ac(i, s) = \bigcap_{s' \in \mathcal{K}(i,s)} Ac(s').$$

Because a stuttering action sta is always included in $Ac(s)$ for any state s, we have that $sta \in Ac(i, s)$ for any agent i. When only sta is in $Ac(i, s)$, we say that the agent cannot do anything because of his limited knowledge. Obviously an agent's subjectively available actions are always part of his physically available actions ($Ac(i, s) \subseteq Ac(s)$). Based on agents' rationality assumptions, he will choose an action with his partial knowledge of the current state and the next state. Given a state s and an action a, an agent considers the next possible states as the set $\mathcal{K}(i, s\langle a\rangle)$. For another action a', the set of possible states is $\mathcal{K}(i, s\langle a'\rangle)$. The question now becomes: How do we compare these two possible set of states? Clearly, when we have $\mathcal{K}(i, s\langle a\rangle) \prec_i \mathcal{K}(i, s\langle a'\rangle)$, meaning that all alternatives of performing action a' are more desirable than all alternatives of choosing action a, it is always better to choose action a'. However, in some cases it might be that some alternatives of action a are better than some alternatives of action a' and vice-versa. In this case, an agent cannot decisively conclude which of the actions is optimal, which implies that the preferences over actions (namely sets of states) is not total. This leads us to the following definition:

Definition 2.7 (Rational Alternatives). Given a state s, an agent i and two actions $a, a' \in Ac(i, s)$, we say that action a is *dominated* by action a' for agent i in state s iff $\mathcal{K}(i, s\langle a\rangle) \prec_i \mathcal{K}(i, s\langle a'\rangle)$. The set of *rational alternatives* for agent i in state s is given by the function $a_i^* : S \longrightarrow 2^{Act}$, which is defined as follows:

$$a_i^*(s) = \{a \in Ac(i, s) \mid \neg\exists a' \in Ac(i, s) : a \neq a' \text{ and } a' \text{ dominates } a \text{ for agent } i \text{ in state } s\}.$$

The set $a_i^*(s)$ are all the actions for agent i in state s which are available to him and are not dominated by another available action. In other words, it contains all the actions which are rational alternatives for agent i. Since it is always the case that $Ac(i, s)$ is non-empty because of the stuttering action sta, and since it is always the case that there is one action which is non-dominated by another action, we conclude that $a_i^*(s)$ is non-empty. We can see that the actions that are available to an agent not only depend on the physical state, but also depend on his knowledge about the state and the next state. The more he knows, the better he can judge what his rational alternative is. In other words, an agent tries to make a best choice based on his value system and incomplete knowledge. We will illustrate the above definitions and our approach through the following example.

Example 2.8. Assume that we have a transition system $\mathcal{M}$ for agent i. State s and s' are agent i's epistemic alternatives, that is, $\mathcal{K}(i, s) = \{s, s'\}$. Now consider the actions that are physically available and subjectively available to agent i. $Ac_i(s) = \{a_1, a_2, a_3, sta\}$,

$Ac_i(s') = \{a_1, a_2, sta\}$. Because $Ac(i,s) = Ac_i(s) \cap Ac_i(s')$, agent i knows that only sta, a_1 and a_2 are available to him in state s .

Next we talk about agent i's rational alternatives in state s. Given agent i's value system $V_i = (u \prec v \prec w)$, and the following valuation: u, $\neg v$ and $\neg w$ hold in $\mathcal{K}(i,s)$, $\neg u$, $\neg v$ and w hold in $\mathcal{K}(i, s\langle a_1\rangle)$, and u, v and $\neg w$ hold in $\mathcal{K}(i, s\langle a_2\rangle)$, we then have the following state preferences: $\mathcal{K}(i,s) \prec \mathcal{K}(i, s\langle a_1\rangle)$, $\mathcal{K}(i,s) \prec \mathcal{K}(i, s\langle a_2\rangle)$ and $\mathcal{K}(i, s\langle a_2\rangle) \prec \mathcal{K}(i, s\langle a_1\rangle)$, meaning that action a_2 and the stuttering action sta are dominated by action a_1. Thus, we have $a_i^*(s) = \{a_1\}$.

3 DEFINING OPPORTUNISTIC PROPENSITY

An agent will perform opportunistic behavior when he has the ability and the desire of doing it, which is called opportunistic propensity in [7]. By intuition, we can eliminate opportunism in the system by removing the ability or the desire. In this section, we will provide the definition of opportunistic propensity, serving as a prerequisite of the mechanism design for eliminating opportunism. Opportunism is a selfish behavior that takes advantage of relevant knowledge asymmetry and results in promoting one agent's own value and demoting another agent's value. It means that it is performed with the precondition of relevant knowledge asymmetry and the effect of value opposition. Firstly, knowledge asymmetry is defined as follows.

Definition 3.1 (Knowledge Asymmetry). Given two agents i and j, and a formula ϕ, knowledge asymmetry about ϕ between agent i and j is the abbreviation:

$$\text{Knowasym}(i,j,\phi) := K_i\phi \wedge \neg K_j\phi \wedge K_i(\neg K_j\phi).$$

It holds in a state where agent i knows ϕ while agent j does not know ϕ and this is also known by agent i. It can be the other way around for agent i and agent j. But we limit the definition to one case and omit the opposite case for simplicity. Now we can define opportunism as follows:

Definition 3.2 (Opportunism Propensity). Given two agents i and j, the assertion Opportunism(i,j,a) that action a performed by agent i is opportunistic behavior is defined as:

$$\text{Opportunism}(i,j,a) := \text{Knowasym}(i,j,\text{promoted}(v^*,a) \wedge \text{demoted}(w^*,a))$$

where $v^* = \text{Mpreferred(i, s, s⟨a⟩)}$ and $w^* = \text{Mpreferred(j, s, s⟨a⟩)}$. We use OPP$(i,j,s)$ to denote the set of opportunistic behavior that can be performed by agent i to agent j in state s. That is,

$$\text{OPP}(i,j,s) = \{a \in Ac(i,s) \mid \mathcal{M}, s \models \text{Opportunism}(i,j,a)\}.$$

This definition shows that if the precondition, Knowasym, is satisfied in a given state then the performance of action a will be opportunistic behavior. As the definition is given with the value systems of agent i and agent j, a value system profile (V_i, V_j) corresponds to one type of opportunistic behavior. The asymmetric knowledge that agent i has is about the change of the truth value of v^* and w^* along the transition by action a, where v^* and w^* are the propositions that agent i and agent j most prefer along the transition respectively. It follows that agent j is partially or completely not aware of it. Definition 3.2 follows our definition of opportunism for reasoning about opportunistic propensity of an agent in a state. As is stressed in [5], opportunistic behavior is performed by intent rather than by accident. In this paper, instead of explicitly modeling intention, we interpret it from agents' rationality that they always intentionally promote their own values. We can derive a proposition from the definition, which is the effect of opportunism.

Proposition 3.3 (Value Promotion and Demotion). *Given a multi-agent system $\mathcal{M}$ and an opportunistic behavior a performed by agent i to agent j in state s, action a will promote agent i's value but demote agent j's value, which can be formalized as*

$$\mathcal{M}, s \models \text{Opportunism}(i,j,a) \quad \textit{implies} \quad s \prec_i s\langle a\rangle \textit{ and } s \succ_j s\langle a\rangle.$$

Proof. From $\mathcal{M}, s \models \text{Opportunism}(i,j,a)$ we have: $\mathcal{M}, s \models K_i(\text{promoted}(v^*,a) \wedge \text{demoted}(w^*,a))$. And thus, since all knowledge is true, we have that $\mathcal{M}, s \models \text{promoted}(v^*,a) \wedge \text{demoted}(w^*,a)$. Since $v^* = \text{Mpreferred(i, s, s⟨a⟩)}$ and $w^* = \text{Mpreferred(j, s, s⟨a⟩)}$, using Definition 2.5, we can conclude $s \prec_i s\langle a\rangle$ and $s \succ_j s\langle a\rangle$. □

Example 3.4. Figure 1 shows the example of selling a broken cup: The action selling a cup is denoted as *sell* and we use two value systems V_s and V_b for the seller and the buyer respectively. State s_1 is the seller's epistemic alternative, while state s_1 and s_2 are the buyer's epistemic alternatives. We also use a dashed circle to represent the buyer's knowledge $\mathcal{K}(b, s_1)$ (not the seller's). In this example, $\mathcal{K}(s, s_1) \subset \mathcal{K}(b, s_1)$. Moreover,

$$hm = \text{Mpreferred}(\text{s}, \text{s}_1, \text{s}_1\langle\text{sell}\rangle),$$

$$\neg hb = \text{Mpreferred}(\text{b}, \text{s}_1, \text{s}_1\langle\text{sell}\rangle),$$

meaning that the seller only cares if he gets money from the transition, while the buyer only cares about if he doesn't have a broken cup from the transition. Note that having a broken cup (hb) is not the same as the cup is broken. We also have

$$\mathcal{M}, s_1 \models K_s(\text{promoted}(hm, sell) \wedge \text{demoted}(\neg hb, sell)),$$

meaning that the seller knows the transition will promote his own value while demote the buyer's value in state s_1. For the buyer, action *sell* is available in both state s_1 and s_2. However, hb doesn't hold in both $s_1\langle sell\rangle$ and $s_2\langle sell\rangle$, so he doesn't know whether he has a broken cup or not after action sell is performed. Therefore, there is knowledge asymmetry between the seller and the buyer about the value changes from s_1 to $s_1\langle sell\rangle$. Action *sell* is potentially opportunistic behavior in state s_1.

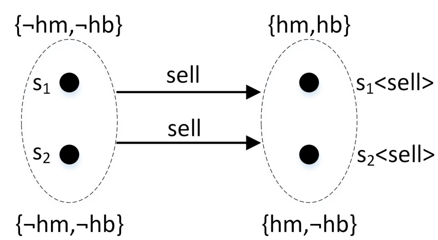

Figure 1: Selling a broken cup

4 ELIMINATING OPPORTUNISM USING AN EPISTEMIC APPROACH

One possible way to eliminate opportunism in the system is to remove the possibility of being opportunistic for agents. Since the precondition of opportunistic behavior is knowledge asymmetry in all states, we can simply prevent the satisfaction of knowledge asymmetry so that it is impossible for agents to perform opportunistic behavior. If we are interested in how the system will behave after updating agents' knowledge, we enter the field of dynamic epistemic logic. Dynamic Epistemic Logic is the study of modal logics of model change by epistemic and doxastic consequences of actions such as public announcements and epistemic actions [2][14]. Opportunism can be eliminated through revealing certain information to the agent involved, such that knowledge asymmetry is removed. This requires the system or someone else in the system to be aware of the information that needs to be revealed. Since the system is not aware of the value system of each agent but has a finite set of possible value systems for each agent, we argue that it is still practical for the system to reveal the important facts to the agent involved. For example, given two possible value systems of the buyer, namely one that cares about the usage of the cup and the other one that cares about the appearance of the cup, the system can make a 3D scan of the cup and then send it to the buyer, so that the buyer gets valuable information about the transaction to decide whether to buy the cup. The event or the procedure is called a revealing update that is performed by the system and results in updating agents' knowledge, and we want to study how to eliminate opportunism by revealing updates in this section. In this paper, we denote a revealing update as $\text{reveal}(\varphi)$ that reveals whether or not formula φ is true. Given a multi-agent system, our logical language $\mathcal{L}_{\text{KA[]}}$ is an extension of $\mathcal{L}_{\text{KA}}$ as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid K_i\varphi \mid \langle a \rangle \varphi \mid [\text{reveal}(\varphi)_i]\psi \quad (i \in Agt, a \in Act)$$

As is standard, formulas with revealing updates are evaluated as follows: given a multi-agent system $\mathcal{M}$ and a state s in $\mathcal{M}$,

- $\mathcal{M}, s \models [\text{reveal}(\varphi)_i]\psi$ iff $\mathcal{M}| \text{reveal}(\varphi)_i, s \models \psi$

where

$$\mathcal{M}| \text{reveal}(\varphi)_i = (Agt, S, Act, \pi, \mathcal{K}', \mathcal{R}, s_0, V_1, ..., V_n)$$

and $\mathcal{K}'$ is defined as follows:

$$s\mathcal{K}'(i)s' \text{ iff } (s\mathcal{K}(i)s' \text{ and } (\mathcal{M}, s \models \varphi \text{ iff } \mathcal{M}, s' \models \varphi)).$$

The above semantics shows that, after the system performs the revealing update $\text{reveal}(\varphi)$ to agent i, agent i's knowledge about φ gets updated, in the way that the access regarding to the indistinguishability of the truth value of φ is removed while the rest of the model remains unchanged. In other words, if φ is true in state s, the epistemic access of agent i that connects state s with the states where φ is false will be removed; if φ is false in state s, the epistemic access of agent i that connects state s with the states where φ is true will be removed. Notice that, after performing a revealing update, it is always possible to make the system consistent with our *no-learning* and *no-forgetting* restriction by repeatedly removing corresponding epistemic access. As this part of making consistent is not what we want to study in this paper, we skip its formal definition. We can also see update $\text{reveal}(\varphi)$ as a process of monitoring performed by the system for the given agent, distinguishing states which satisfy φ from those which do not satisfy φ. Since this procedure returns a value from the set $\{\varphi, \neg\varphi\}$, in the rest of the paper we always discuss two cases where φ holds and doesn't hold in the actual state for any definition and proof. We have the following validity, given a multi-agent system $\mathcal{M}$, a revealing update $\text{reveal}(\varphi)_i$,

$$\mathcal{M} \models \varphi \to [\text{reveal}(\varphi)_i]K_i\varphi,$$

which means that if φ holds then agent i knows φ after φ is revealed. Further, if the system reveals something to an agent that he has already knew, the model will remain the same. We formalize it as

$$\text{if } \mathcal{M} \models K_i\varphi, \text{ then } \mathcal{M}| \text{reveal}(\varphi)_i = \mathcal{M}.$$

This is because the revealing update will not cause any epistemic access removal from the model.

In this paper, we want to investigate how to eliminate the performance of opportunism, typically through removing knowledge asymmetry in the system. In order to do that, we firstly introduce the notion *Eliminating Opportunism by a Revealing Update*: we say that a revealing update can eliminate opportunism if and only if the revealing update disables its performance, namely precondition *knowledge asymmetry* is removed by the revealing update. Formally,

Definition 4.1 (Eliminating Opportunism by a revealing update). Given a multi-agent system $\mathcal{M}$, an opportunistic behavior a performed by agent i to agent j in state s, and a revealing update $\text{reveal}(\xi)_j$, we say the revealing update can eliminate opportunistic behavior a iff

$$\mathcal{M}, s \models [\text{reveal}(\xi)_j] \neg \text{Knowasym}(i, j, \text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a)),$$

where $v^* = \text{Mpreferred}(\text{i, s, s}\langle\text{a}\rangle)$ and $w^* = \text{Mpreferred}(\text{j, s, s}\langle\text{a}\rangle)$.

This definition shows how a revealing update eliminates opportunistic behavior: revealing update $\text{reveal}(\xi)_j$ disables the performance of opportunistic behavior a by making knowledge asymmetry false in the new system. Notice that based on the semantics of our framework, action a, which was opportunistic, is still not removed. However, since there is no knowledge asymmetry between agent i and agent j, agent j can prevent agent i from performing opportunistic behavior a, or can still accept it. In the latter case, action a is no longer opportunistic as knowledge asymmetry is false. For instance, *sell* and *buy* are synchronized to be one action. After the system reveals to the buyer that the cup is broken, the buyer will not buy the cup so that the deal cannot be done, or the buyer will still buy the broken cup as it is his only choice, but the latter case is not opportunistic behavior since there is no knowledge asymmetry about the deal. Moreover, as the system is not aware of the value system of each agent, the system reveals to agent j all the information that he might most care about in the transition, given a set of possible value systems of agent j. We can immediately have the following proposition, which shows the relationship between revealing updates and asymmetric knowledge:

PROPOSITION 4.2. *Given a multi-agent system $\mathcal{M}$, an opportunistic behavior a performed by agent i to agent j in state s and a revealing update* $\text{reveal}(\xi)_j$*, the revealing update can eliminate opportunistic behavior a if*

- *in the case* $\mathcal{M}, s \models \xi$*,* $\mathcal{M} \models K_j(\xi \rightarrow (\text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a)))$*,*
- *in the case* $\mathcal{M}, s \models \neg\xi$*,* $\mathcal{M} \models K_j(\neg\xi \rightarrow (\text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a)))$*,*

where $v^* = \text{Mpreferred(i, s, s}\langle\text{a}\rangle)$ *and* $w^* = \text{Mpreferred(j, s, s}\langle\text{a}\rangle)$.

Proof. When ξ holds in state s, $\mathcal{M}, s \models K_j\xi$ holds after reveal$(\xi)_j$ is performed. As $\mathcal{M} \models K_j(\xi \rightarrow (\text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a)))$ implies $\mathcal{M} \models K_j\xi \rightarrow K_j(\text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a))$, we have $\mathcal{M}, s \models K_j(\text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a))$. Thus, there is no knowledge asymmetry between agent i and agent j about formula $\text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a)$. Therefore, according to Definition 4.1, reveal$(\xi)_j$ eliminate opportunistic behavior a. We can prove it in a similar way when $\neg\xi$ holds in state s. □

That is what we can directly derive from the definition of opportunism: to eliminate opportunism by removing the precondition of knowledge asymmetry between different agents. Notice that agent j is not aware of the whole formula $\text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a)$ but might know part of the formula, for example $\text{demoted}(w^*, a)$. In that case, the system needs to reveal ξ to agent j and agent j knows $\xi \rightarrow \text{promoted}(v^*, a)$ or $\neg\xi \rightarrow \text{promoted}(v^*, a)$.

Ideally we can let every agent have exactly the same knowledge such that there is no knowledge asymmetry thus nobody can perform opportunistic behavior. However, it is difficult to implement such an extreme case in reality, because sometimes we would like to design a system that can respect agents' privacy, which is realized through setting privacy norms. However, since the system designer is not aware of agents' value systems thus doesn't know what to reveal to agents for eliminating opportunistic behavior, there might exist privacy norms that prevent the system from revealing to agents the information that can eliminate opportunism. Namely, the revealing update performed by the system might reveal the information that the system wants to keep secret through setting a privacy norm. One simple example is that the system wants to reveal φ to an agent for eliminating opportunistic behavior but as is stated in a privacy norm that the agent should not be aware of the information about φ. Hence, there exists a balance between respecting of agents' privacy and eliminating of opportunism. In other words, the system can perform revealing updates to agents for eliminating opportunistic behavior, but also lower the privacy level in the system. Privacy Norms are formalized as follows:

Definition 4.3 (Privacy Norms). Let i and j be two agents, and γ be a formula in $\mathcal{L}_{\text{KA}}$, a privacy norm is in the form of Knowasym(i, j, γ), stating that agent i should have the privacy about the fact γ from agent j. Given a multi-agent system $\mathcal{M}$ with a state s, we say that privacy norm Knowasym(i, j, γ) in state s is respected if $\mathcal{M}, s \models$ Knowasym(i, j, γ), and we use $\Pi(s)_{\mathcal{M}}$ to denote the set of privacy norms that are implemented in state s. We will write $\Pi(s)$ for short if it is clear from context.

In this paper, we assume that there are some privacy norms that are supposed to be respected in the system. For instance, privacy norm Knowasym$(s, b, oprice)$ is interpreted as the seller should keep the original price in secret from the buyer. Privacy norms are state-sensitive in the sense that a privacy norm can be active in a state while inactive in another state.

In principle, given a set of possible value system profiles and a privacy norm, the system has to consider every possible value system profile in order to identify an action to be opportunistic, and then think about whether there exists a revealing update that can eliminate opportunistic behavior and respect the privacy norm as well. In this paper, we skip the first part for simplification, assuming that opportunistic behavior is given, in order to focus on the study about the trade-off between eliminating opportunistic behavior and respecting the privacy norm. Namely, suppose we already identified an action to be opportunistic behavior with a possible value system profile, a question arises: *Given opportunistic behavior and a privacy norm, does there exist a revealing update that can eliminate opportunistic behavior and respect the privacy norm as well?* Intuitively, an agent gets to know something after something was revealed to the agent, but the revealing update might disrespect another agent's privacy, which is stated by our privacy norms in the system. The following proposition shows that in which case a revealing update respects a privacy norm:

Proposition 4.4. *Given a multi-agent system* $\mathcal{M}$ *in a state* s*, a privacy norm* Knowasym$(i, j, \gamma) \in \Pi(s)$ *with respect to formula* γ*, and a revealing update* reveal$(\xi)_j$*, the revealing update respects privacy norm* Knowasym(i, j, γ) *if:*

- *in the case* $\mathcal{M}, s \models \xi$*,* $\mathcal{M}, s \models \neg K_j(\xi \rightarrow \gamma)$*,*
- *in the case* $\mathcal{M}, s \models \neg\xi$*,* $\mathcal{M}, s \models \neg K_j(\neg\xi \rightarrow \gamma)$*,*

Proof. In order to respect privacy norm Knowasym(i, j, γ), according to Definition 3.1, we need to ensure $\mathcal{M}, s \models [\text{reveal}(\xi)_j]\neg K_j\gamma$ so that $\mathcal{M}, s \models [\text{reveal}(\xi)_j]\,\text{Knowasym}(i, j, \gamma)$. In the case where ξ holds, $\mathcal{M}, s \models [\text{reveal}(\xi)_j]K_j\xi$ after the revealing update is performed to agent j. Furthermore, $\mathcal{M}, s \models \neg K_j(\xi \rightarrow \gamma)$ implies that there exists $s' \in \mathcal{K}(j, s) : \mathcal{M}, s' \models \neg(\xi \rightarrow \gamma)$, which is equivalent to $\mathcal{M}, s' \models \xi \wedge \neg\gamma$. Since agent j's epistemic access which connects $\neg\xi$-state to state s gets removed after the revealing update is performed, state s' where $\xi \wedge \neg\gamma$ holds is still in agent j's knowledge set. In other words, there exists $s' \in \mathcal{K}(j, s) : \mathcal{M}|\,\text{reveal}(\xi)_j, s' \models \xi \wedge \neg\gamma$. Therefore, we can conclude that $\mathcal{M}, s \models [\text{reveal}(\xi)_j]\neg K_j\gamma$ and it leads to $\mathcal{M}, s \models [\text{reveal}(\xi)_j]\,\text{Knowasym}(i, j, \gamma)$. We can prove it in a similar way when $\neg\xi$ holds in state s. □

The proposition shows that privacy norm Knowasym(i, j, γ) is respected if agent j is not aware of the inference. Conversely, if the above statement doesn't hold, the revealing update will reveal the information that the system wants to keep in private between agents. From Proposition 4.2 and Proposition 4.4, we can see our research problem is equivalent to the problem whether there exists a formula ξ such that the formulas from both propositions hold. Therefore,

Proposition 4.5. *Given a multi-agent system* $\mathcal{M}$ *in state* s*, an opportunistic behavior* a *performed by agent* i *to agent* j*, a privacy norm* Knowasym$(i, j, \gamma) \in \Pi(s)$ *and a revealing update* reveal$(\xi)_j$*,* reveal$(\xi)_j$ *can eliminate opportunistic behavior* a *and respect privacy norm* Knowasym(i, j, γ) *if:*

- *in the case* $\mathcal{M}, s \models \xi$*,* $\mathcal{M}, s \models K_j(\xi \rightarrow (\text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a))) \wedge \neg K_j(\xi \rightarrow \gamma)$*,*
- *in the case* $\mathcal{M}, s \models \neg\xi$*,* $\mathcal{M}, s \models K_j(\neg\xi \rightarrow (\text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a))) \wedge \neg K_j(\neg\xi \rightarrow \gamma)$*,*

where $v^* = \text{Mpreferred(i, s, s}\langle\text{a}\rangle)$ *and* $w^* = \text{Mpreferred(j, s, s}\langle\text{a}\rangle)$.

Proof. The statement is the combination of the statements from Proposition 4.2 and Proposition 4.4. When agent j is aware of $\xi \rightarrow (\text{promoted}(v^*, a) \wedge \text{demoted}(w^*, a))$, $\text{reveal}(\xi)_j$ can eliminate opportunistic behavior a; when agent j is not aware of $\xi \rightarrow \gamma$, $\text{reveal}(\xi)_j$ respects privacy norms $\text{Knowasym}(i, j, \gamma)$. Again, we can prove it in a similar way when $\neg\xi$ holds in state s. □

Essentially, the above proposition shows the relation among a revealing update, agents' value systems and a privacy norm: if what an agent cares about, which his value system reflects, is not respected by the system through setting corresponding privacy norms, such a revealing update to the agent doesn't exist. In other words, it is dependent on the compatibility between agents' value systems and the privacy norms in the system. For example, for the case where ξ holds, in order to eliminate opportunistic behavior a, the system has to reveal (verify) ξ to agent j, who knows that ξ implies value opposition along the transition. However, if he is also aware of the formula $\xi \rightarrow \gamma$, such a revealing update will reveal to agent j the information about γ, which is against the privacy norm. Hence, there is no revealing update that can eliminate opportunistic behavior a and respect the privacy norm with respect to γ as well. Further, sometimes formula $\xi \rightarrow \gamma$ is valid in $\mathcal{M}$ thus it becomes universal knowledge in the system. In that case, revealing update $\text{reveal}(\xi)$ will always reveal the information about γ we want to keep in private. Thus, we have to remove privacy norm $\text{Knowasym}(i, j, \gamma)$ so that it is allowed to perform revealing update $\text{reveal}(\xi)$ to eliminate opportunistic behavior a, which can be seen as an alternative normative approach.

Example 4.6. We again consider the senario shown in Example 3.4. There is knowledge asymmetry between the seller and the buyer,

$$\text{Knowasym}(s, b, \text{promoted}(hm, sell) \wedge \text{demoted}(\neg hb, sell)),$$

which is equivalent to

$$\text{Knowasym}(s, b, \neg hm \wedge \langle sell\rangle hm \wedge \neg hb \wedge \langle sell\rangle hb).$$

In this scenario the seller knows the transition will promote his own value while demote the value of the buyer, but the buyer is not aware of the demotion part, as $\langle sell\rangle hb$ doesn't hold in both state s_1 and state s_2. Now the buyer performs revealing update $\text{reveal}(broken)_b$ to check whether the cup is broken or not, and he also knows that his value will get demoted while the buyer's value will get promoted if the cup is broken, that is,

$$\mathcal{M}, s \models K_b(broken \rightarrow (\text{promoted}(hm, sell) \wedge \text{demoted}(\neg hb, sell))),$$

which implies

$$\mathcal{M}, s \models K_b broken \rightarrow K_b(\text{promoted}(hm, sell) \wedge \text{demoted}(\neg hb, sell)).$$

Since the cup is actually broken ($\mathcal{M}, s \models broken$), the buyer gets to know the cup is broken after the system performs revealing update $\text{reveal}(broken)_b$ to him ($\mathcal{M}, s \models K_b broken$) and thus he knows his value will get demoted while the buyer's value will get promoted,

$$\mathcal{M}, s \models K_b(\text{promoted}(hm, sell) \wedge \text{demoted}(\neg hb, sell)).$$

Therefore, there is no knowledge asymmetry about the transition between the seller and the buyer (shown in Fig. 2), which prevents the seller from opportunistically selling the broken cup to the buyer, according to Definition 3.2. Next we suppose a privacy norm $\text{Knowasym}(s, b, oprice)$ in the system, which means that the seller should keep the original price in private. Since inference $broken \rightarrow oprice$ is not valid in $\mathcal{M}$ intuitively, the buyer is not aware of it,

$$\mathcal{M}, s \models \neg K_b(broken \rightarrow oprice).$$

Therefore, revealing update $\text{reveal}(broken)_b$ won't reveal the original price to the buyer and privacy norm $\text{Knowasym}(s, b, oprice)$ is still respected in the updated system.

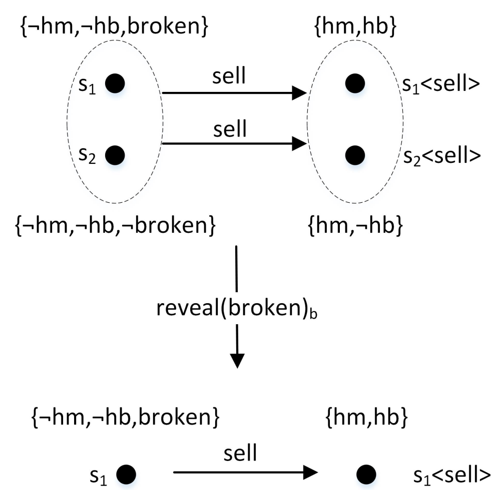

Figure 2: Update by revealing update $reveal(broken)_b$

5 RELATION TO MECHANISM DESIGN

Mechanism design is a field to design a game with desirable properties (outcomes) for various agents to play [8] [11]. Given agents' preferences $\precsim$ and an assumed solution concept g that defines agents' way of finding optimal outcomes, we can make a prediction of the outcomes that will be achieved, which is represented as $g(\precsim)$. Given agents' preferences $\precsim$ and a social choice rule f that specifies the criteria of the desirable outcomes, we say that $f(\precsim)$ are the set of social optimal outcome, which are the outcomes we want to have. Since agents' preference might be unknown to us, our goal is to design mechanisms such that for all the possible preference $\precsim$ the predicted outcomes $g(\precsim)$ coincide with (or is a subset of) the desirable outcomes $f(\precsim)$ (more elaboration can be found in [4]). In this paper, we take a slightly different view of mechanism design from the traditional one above: we consider a mechanism as an operation or an update to the system, which is a revealing update. When applying the theory of mechanism design to eliminate opportunism, we see agents' rational alternatives as predicted outcomes, opportunistic behaviors as undesirable outcomes, and our goal is to design revealing updates to the system such that for all the possible value system profiles the intersection of an agent's *rational alternatives* (using our decision theory) and *opportunistic behaviors* in the new system is empty. In this section, we will discuss how a revealing update implements non-opportunism respectively.

Given an opportunistic behavior, we know what kind of information the system needs to reveal to an agent for eliminating it. However, if we take into account an agent's decision-making, it can be the case where it is not optimal for the agent to perform such an opportunistic behavior thus it is not necessary to eliminate it. In this sense, we connect revealing updates with rational alternatives. Hence, the goal of this paper is to find out an update, namely a revealing update, such that it is not optimal for the agent to behave opportunistically after it is implemented. Given a value system for agent i, we know the set of agent i's rational alternatives $a_i^*(s)$. Given a value system profile for agent i and j, we can identify the set of opportunistic behaviors $OPP(i, j, s)$ that agent i and j are involved in. We use $a_i^*(s)|\operatorname{reveal}(\xi)_j$ and $OPP(i, j, s)|\operatorname{reveal}(\xi)_j$ to denote the set of rational alternatives and the set of opportunistic behaviors after $\operatorname{reveal}(\xi)_j$ is performed in state s respectively. Because opportunistic behavior is undesired from the perspective of the system and agents form their rational alternatives (possibly opportunistic) based on their value systems, it is important to know whether a revealing update implements non-opportunism. Formally, we define *non-opportunistic implementation* as follows:

Definition 5.1 (Non-opportunistic Implementation). Given a multi-agent system $\mathcal{M}$ with two agents i and j in state s, and a revealing update $\operatorname{reveal}(\xi)_j$, we say that $\operatorname{reveal}(\xi)_j$ implements non-opportunism iff $a_i^*(s)|\operatorname{reveal}(\xi)_j \cap OPP(i, j, s)|\operatorname{reveal}(\xi)_j = \emptyset$.

A revealing update implements non-opportunism if and only if the intersection between rational alternatives and opportunistic behaviors becomes an empty set after the revealing update is performed. Clearly, this concerns the update that a revealing update brings to the system. With our update logic of revealing updates, we now discuss how a revealing update influences an agent's decision-making and the identification of opportunistic behavior.

Proposition 5.2. *Given a multi-agent system $\mathcal{M}$ with two different agents i and j in state s, and a revealing update* $\operatorname{reveal}(\xi)_j$*, agent i's rational alternatives will remain the same after* $\operatorname{reveal}(\xi)_j$ *is performed in state s, which is formalized as* $a_i^*(s) = a_i^*(s)|\operatorname{reveal}(\xi)_j$.

Proof. Since revealing update $\operatorname{reveal}(\xi)_j$ is performed by the system to agent j, agent i's epistemic structure will remain the same after $\operatorname{reveal}(\xi)_j$ is performed. Hence, according to Definition 2.6 and 2.7, agent i's subjectively available actions and rational alternatives will remain the same after $\operatorname{reveal}(\xi)_j$ is performed. □

Proposition 5.3. *Given a multi-agent system $\mathcal{M}$ with two different agents i and j in state s, and a revealing update* $\operatorname{reveal}(\xi)_j$*, opportunistic behaviors performed by agent i to agent j will not become more after* $\operatorname{reveal}(\xi)_j$ *is performed, which is formalized as* $OPP(i, j, s) \supseteq OPP(i, j, s)|\operatorname{reveal}(\xi)_j$.

Proof. Given a value system profile for agent i and j, we can identify the set of opportunistic behaviors $OPP(i, j, s)$ in a state. Because $\operatorname{reveal}(\xi)_j$ causes update of agent j' knowledge, knowledge asymmetry will become false after $\operatorname{reveal}(\xi)_j$, and thus some actions will become non-opportunistic. Because the system might reveal the information that is not relevant to any opportunistic behavior, it is possible that all the opportunistic behaviors remain unchanged. □

If we limit a revealing update to the one that is performed to agent j, agent i's rational alternatives will remain the same while opportunistic behaviors performed by agent i to agent j will remain the same or become less, after $\operatorname{reveal}(\xi)_j$ is performed. Therefore, if a revealing update can eliminate all the actions in the intersection of rational alternatives and opportunistic behavior, it implements non-opportunism. Notice that action a, which was opportunistic behavior, is still in agent i's rational alternatives, but it is not opportunistic any more because knowledge asymmetry regarding opportunistic behavior a is already removed. As for Example 4.6, we see that $\operatorname{reveal}(broken)_b$ can eliminate opportunistic behavior *sell*. Even though the seller can still sell the broken cup to the buyer, it is not opportunistic behavior any more because the buyer already know that he will have a broken cup. Therefore, we can conclude that given a set of value system profiles $\hat{V} = \{(V_s, V_b)\}$ revealing update $\operatorname{reveal}(broken)_b$ implements non-opportunism.

6 CONCLUSION

Opportunism is a behavior that takes advantage of relevant knowledge asymmetry and results in promoting an agent's own value and demoting another agent's value. As opportunistic behavior has undesirable results for other agents who participate in the system, it is important to design mechanisms to eliminate opportunism. In this paper we developed an epistemic approach to eliminate opportunism in multi-agent systems: we eliminated opportunism by removing the precondition of opportunism *knowledge asymmetry*, which disables the performance of opportunism. *Knowledge asymmetry* is removed by agents' revealing updates, which might reveal the information that the system wanted to keep private between agents through setting privacy norms. So we investigated the balance between eliminating opportunism and respecting agents' privacy. Finally, we related our approach to the theory of mechanism design. However, we do recognize some downsides of our proposed mechanism. Firstly, in order to eliminate opportunism, the system has to reveal to an agent the information that he might care about for the transition given his possible value systems. This is indeed an ideal practice because it will be difficult to achieve it when the set of possible value systems become large. Secondly, in order to reveal useful information to agents, the system has to first identify if a given action is opportunistic behavior with a set of value system profiles for the agents involved, and then reveal appropriate information to the agents to eliminate opportunism. Those revealing updates should not be demotivated by the system through setting privacy norms. This indeed puts a burden on the designer before implementing any privacy norms, as agents' value systems are initially unknown. An agent performs opportunistic behavior when he has the ability and the desire of doing that. Instead of removing the ability, future work can be done by removing the desire, namely making the choice of being opportunistic not optimal. As there exists trade-off between eliminating opportunism and respecting agents' privacy, it will be interesting to eliminate opportunism through removing privacy norms.

ACKNOWLEDGMENTS

The authors would like to thank Chinese Scholarship Council and Centrum Wiskunde & Informatica for their financial support.

REFERENCES

[1] Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. 2007. Normative system games. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 129.

[2] Alexandru Baltag and Bryan Renne. 2016. Dynamic Epistemic Logic. In *The Stanford Encyclopedia of Philosophy* (winter 2016 ed.), Edward N. Zalta (Ed.). Metaphysics Research Lab, Stanford University.

[3] Nils Bulling and Mehdi Dastani. 2016. Norm-based mechanism design. *Artificial Intelligence* 239 (2016), 97–142.

[4] Max Knobbout, Mehdi Dastani, and John-Jules Ch. Meyer. 2016. Formal Frameworks for Verifying Normative Multi-agent Systems. In *Theory and Practice of Formal Methods - Essays Dedicated to Frank de Boer on the Occasion of His 60th Birthday*. 294–308. https://doi.org/10.1007/978-3-319-30734-3_20

[5] Jieting Luo and John-Jules Meyer. 2017. A formal account of opportunism based on the situation calculus. *AI & SOCIETY* 32, 4 (01 Nov 2017), 527–542. https://doi.org/10.1007/s00146-016-0665-4

[6] Jieting Luo, John-Jules Meyer, and Max Knobbout. 2016. Monitoring opportunism in multi-agent systems. In *Proceedings of the Coordination, Organizations, Institutions and Norms workshop@ECAI2016*.

[7] Jieting Luo, John-Jules Meyer, and Max Knobbout. 2017. Reasoning about Opportunistic Propensity in Multi-agent Systems. In *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers*. to appear.

[8] Eric S Maskin. 2008. Mechanism design: How to implement social goals. *The American Economic Review* 98, 3 (2008), 567–576.

[9] Robert C Moore. 1980. *Reasoning about knowledge and action*. SRI International Menlo Park, CA.

[10] Robert C Moore. 1984. *A formal theory of knowledge and action*. Technical Report. DTIC Document.

[11] Noam Nisan. 2007. Introduction to mechanism design (for computer scientists). *Algorithmic game theory* 9 (2007), 209–242.

[12] Jeremy Pitt and Alexander Artikis. 2015. The open agent society: retrospective and prospective views. *Artificial Intelligence and Law* 23, 3 (2015), 241–270.

[13] TL Van der Weide. 2011. *Arguing to motivate decisions*. Ph.D. Dissertation. Utrecht University.

[14] Hans Van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. 2007. *Dynamic epistemic logic*. Vol. 337. Springer Science & Business Media.

[15] O.E. Williamson. 1975. *Markets and hierarchies, analysis and antitrust implications: a study in the economics of internal organization*. Free Press. https://books.google.nl/books?id=3Fi3AAAAIAAJ