Concurrent dynamic epistemic logic for MAS

H.P. van Ditmarsch
Computer Science
University of Otago
Dunedin, New Zealand
hans@cs.otago.ac.nz

W. van der Hoek
Computer Science
University of Liverpool
Liverpool, United Kingdom
wiebe@csc.liv.ac.uk

B.P. Kooi
Computing Science
University of Groningen
Groningen, the Netherlands
barteld@cs.rug.nl

ABSTRACT

When giving an analysis of knowledge in multiagent systems, one needs a framework in which higher-order information and its dynamics can both be represented. A recent tradition starting in original work by Plaza treats all of knowledge, higher-order knowledge, and its dynamics on the same foot. Our work is in that tradition. It also fits in approaches that not only dynamize the epistemics, but also epistemize the dynamics: the actions that (groups of) agents perform are epistemic actions. Different agents may have different information about which action is taking place, including higher-order information. We demonstrate that such information changes require subtle descriptions. The contribution of our paper is that it provides a complete axiomatization for an action language of van Ditmarsch, where an action is interpreted as a relation between states and sets of states. The applicability of the framework is found in every context where multiagent strategic decision making is at stake, and already demonstrated in game-like scenarios such as Cluedo and card games.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

Keywords

agent communication, concurrency, actions, dynamic logic, epistemic logic

1. INTRODUCTION

Since Hintikka's [14] epistemic logic, the logic of knowledge, has been a subject of research in philosophy [15], computer science [9], artificial intelligence [17] and game theory [3]. The latter three application areas made it more and more apparent that in multiagent systems *higher-order information*, knowledge about other agents' knowledge, is crucial.

The famous paper [1] by Alchourrón *et al.* put the *change of information*, or belief revision, as a topic on the philosophical and logical agenda: it was followed by a large stream of publications and much research in: belief revision, finetuning the notion of epistemic entrenchement [18], revising (finite) belief bases [6], differences between belief revision and belief updates [16], and the problem of iterated belief change [8]. However, in all these approaches the dynamics is studied at a level above the informational level, making it impossible to reason about change of agents' knowledge and ignorance within the framework, let alone about the change of other agents' information.

Our work takes these the observations on higher-order knowledge and change of information as a starting point: when giving an analysis of knowledge in multiagent systems, one needs a framework in which higher-order information and its dynamics can be represented.

Although the notion of a run in an interpreted system as described in [9] makes it in principle possible to reason about the dynamics of an agent's knowledge, the interpretation of a run is typically that of a standard program. Further, the pioneering work of Moore [17] also studies the relation between actions and knowledge: there the emphasis is on epistemic preconditions that are needed to perform certain actions in the world, such as knowing a key-combination in order to open a safe.

From the point of view of expressivity, one can say that the work on interpreted systems enables one to reason about the (change of) knowledge over time, and by adding actions to the language, one can also reason about the change of knowledge brought about by performing certain plans. This enables one to express properties like *perfect recall* and *no learning*. Recently, based on work by Alur et al [2], van der Hoek and Wooldridge [25] added a *social* or *coalitional aspect* to an epistemic framework, giving them the possibility to express that for instance a group can establish that some knowledge is eventually obtained, or that two agents can enforce that they exchange a secret, without a third agent getting to know this.

Our work fits in approaches that not only dynamize the epistemics, but also epistemize the dynamics: the actions that (groups of) agents perform are epistemic actions. Different agents may have different information about which action is taking place, including higher-order information. This rather recent tradition treats all of knowledge, higher-order knowledge, and its dynamics on the same foot. Following an original contribution by Plaza in 1989 [22], a stream of publications appeared around the year 2000 [11, 10, 4, 26, 24, 5, 29, 28, 23].

The following, possibly simplest example in the setting of MAS (two agents, one atom) tries to demonstrate that the notions of higher-order information and epistemic actions are indeed important and may be subtle.

> *Anne and Bert are in a bar, sitting at a table. A messenger comes in and delivers a letter to Anne. The letter contains either an invitation for a night out in Amsterdam, or an obligation to give a lecture instead. Anne and Bert commonly know that these are the only alternatives.*

This situation can be modelled as follows: There is one atom p, describing 'the letter contains an invitation for a night out in Amsterdam', so that $\neg p$ stands for the lecture obligation. There are two agents 1 (Anne) and 2 (Bert). Whatever happens in each of the following action scenarios, is publicly known (to Anne and Bert). Also, assume that in fact p is true.

Action scenario 1. *Anne reads the letter aloud. (*tell*)*

Action scenario 2. *Bert is seeing that Anne reads the letter. (*read*)*

Action scenario 3. *Bert orders a drink at the bar so that Anne may have read the letter. (*mayread*)*

Action scenario 4. *Bert orders a drink at the bar while Anne goes to the bathroom. Both may have read the letter. (*bothmayread*)*

After execution of the first scenario it is common knowledge that p: in the resulting state $C_{12}p$ (i.e. $C_{\{1,2\}}p$) holds. This is not the case in the second scenario, but still, some common knowledge is obtained there: $C_{12}(K_1p \vee K_1\neg p)$: it is commonly known that Anne knows the contents of the letter, irrespective of it being p or $\neg p$. Does this higher-order information change in Scenario 3? Yes, in this case Bert does not even know if Anne knows p or knows $\neg p$: $\neg K_2(K_1p \vee K_1\neg p)$. In Scenario 4 something similar happens, that may best be described by saying that the agents concurrently learn that the other may have learnt p or $\neg p$. Now both agents may have learnt p, after which p is generally known – $E_{12}p$ –, but they are in that case unaware of each other's knowledge – $\neg C_{12}p$ –, and *that* is commonly known.

Van Ditmarsch has described such actions as *knowledge actions* (with corresponding dynamic modal operators) in a multiagent dynamic epistemic language [26, 29]. Knowledge actions are interpreted as a relation between states. The contribution of our paper is that it provides a complete axiomatization for the extension of this language with concurrency as found in [27, 28]. This builds on work on concurrency in dynamic logic (PDL) [21, 13, 12] and is partially related to game theoretical semantics for (extensions of) PDL [19, 20].

The applicability of the framework is found in every context where multiagent strategic decision making is at stake, and already demonstrated in game-like scenarios such as Cluedo and card games [26].

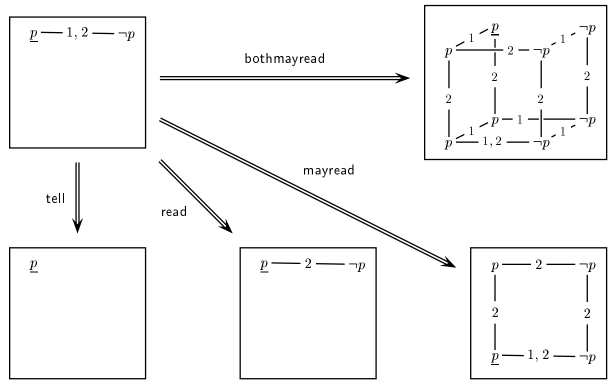

Figure 1: Actions for two agents and one atom. The top left figure represents (Arc, u). Points of models are underlined. Assume transitivity of access. For mayread **and** bothmayread **only one of more executions is shown.**

Section 2 introduces the language and its semantics. Section 3 defines the axioms and derivation rules, its syntactic prerequisites such as syntactic equivalence of actions, and shows the soundness of this proof system. Section 4 shows the completeness of this proof system. Section 5 gives some applications of the language in specifying MAS dynamics, and is followed by the conclusions (Section 6). Almost all proofs have been omitted.

2. LANGUAGE AND SEMANTICS

Structures Given a finite set of *agents* N and a set of *atoms* P, a (Kripke) *model* $M = \langle W, R, V\rangle$ consists of a domain W of *worlds* or factual states, for each agent $n \in N$ a binary *accessibility relation* R_n on W, and a *valuation* $V : P \to \mathcal{P}(W)$, or in other words: for each atom $p \in P$, a subset V_p of W. Given a model, the operator gr returns the set of agents: $gr(\langle W, R, V\rangle) = N$; this is called the *group* of the model. The group of a set of models is the union of the groups of these models. In an *epistemic model* (commonly known as an $S5$ model) all accessibility relations are equivalence relations. We then write $\sim_n$ for the equivalence relation for agent n. If $w \sim_n w'$ we say that *w is the same as w'* for n, or that *w is equivalent to w'* for n. Write $\sim_B$ for $(\bigcup_{n\in B} \sim_n)^*$. Write $\mathcal{S}5_N(P)$ for the class of epistemic models for agents N and atoms P, write $\mathcal{S}5_{\subseteq N}(P)$ for $\bigcup_{B\subseteq N} \mathcal{S}5_B(P)$. We drop the '$P$' if it is clear from the context.

For a given model M, $\mathcal{D}(M)$ returns its domain. Instead of $w \in \mathcal{D}(M)$ we also write $w \in M$. Given a model M and a world $w \in M$, (M, w) is called a *modal state*, w the *point* of that state, and M the model *underlying* that state. Also, if M is clear from the context, write $\underline{w}$ for (M, w). Similarly, we *visually* point to a world in a figure by underlining it. If $s = (M, w)$ and $w \in \mathcal{D}(M)$ we also write $w \in s$. All notions for models are assumed to be similarly defined for modal states. Write $\bullet\mathcal{S}5_N(P)$ for the class of epistemic states ('pointed' - '$\bullet$' - models) for agents N and atoms P.

Example 5. *The background setting for 'Lecture or Amsterdam' can be represented by an epistemic state. Arc is the model $\langle\{u, v\}, \sim, V\rangle$ such that both $\sim_1$ and $\sim_2$ are the universal relation on $\{u, v\}$, and $V_p = \{u\}$. The state (Arc, u) corresponds to p being actually the case. After Anne has* *read the letter, a state is reached that is like (Arc, u) but with $\sim_1 = \{(u,u),(v,v)\}$ instead. See Figure 1.*

Syntax To a standard multiagent epistemic language with common knowledge for a set N of agents and a set P of atoms [17, 9], we add dynamic modal operators for programs that are called knowledge actions and that describe actions. The language $\mathcal{L}_N$, the knowledge actions $\mathcal{L}_N^{\mathsf{act}}$, and the group gr are defined by simultaneous induction:

DEFINITION 6 (FORMULAS AND ACTIONS). *The* formulas $\mathcal{L}_N(P)$ *are defined by*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_n\varphi \mid C_B\varphi \mid [\alpha]\psi$$

where $p \in P$, $n \in N$, $B \subseteq N$, $\alpha \in \mathcal{L}_N^{\mathsf{act}}(P)$, *and* $\psi \in \mathcal{L}_{gr(\alpha)}(P)$. *The* actions $\mathcal{L}_N^{\mathsf{act}}(P)$ *are defined by*

$$?\varphi \mid L_B\beta \mid (\alpha \ ! \ \alpha) \mid (\alpha \ ; \ \beta') \mid (\alpha \cup \alpha) \mid (\alpha \cap \alpha)$$

where $\varphi \in \mathcal{L}_N(P)$, $B \subseteq N$, $\beta \in \mathcal{L}_B^{\mathsf{act}}(P)$, *and* $\beta' \in \mathcal{L}_{gr(\alpha)}^{\mathsf{act}}(P)$, *and where the* group $gr(\alpha)$ *of an action* $\alpha \in \mathcal{L}_N^{\mathsf{act}}(P)$ *is defined as:* $gr(?\varphi) := \emptyset$, $gr(L_B\alpha) := B$, *and* $gr(\alpha \bullet \alpha') := gr(\alpha) \cap gr(\alpha')$ *for* $\bullet = \ ! \ , \cap, \cup, \ ; \ .$

Other propositional connectives and modal operators are defined by standard abbreviations, in particular $E_B\varphi := \bigwedge_{n\in B} K_n\varphi$. Outermost parentheses of formulas and actions are deleted whenever convenient. As we may generally assume an arbitrary P, write $\mathcal{L}_N$ instead of $\mathcal{L}_N(P)$, and $\mathcal{L}_N^{\mathsf{act}}$ instead of $\mathcal{L}_N^{\mathsf{act}}(P)$. Instead of, e.g., $C_{\{a,b,c\}}$ we always write C_{abc}. For an arbitrary epistemic ('box') operator K, write $\hat{K}$ for its dual ('diamond'). The dual of $[\alpha]$ is $\Diamond\alpha$.

The program constructor L_B is called *learning.* Action $?\varphi$ is a *test*, $(\alpha \ ; \ \alpha')$ is *sequential execution*, $(\alpha \cup \alpha')$ is *nondeterministic choice*, $(\alpha \ ! \ \alpha')$ is called *local choice*, and $(\alpha \cap \alpha')$ is *concurrent execution.* The construct $L_B?\varphi$ is pronounced as 'B learn *that* φ', the construct $\alpha \ ! \ \alpha'$ is pronounced as 'from α and α', choose the first'. We will see that the interpretation of local choice depends on the context of learning that binds it (as in $L_B(\alpha \ ! \ \alpha')$: everybody in B but not in α, α', is unaware of the choice, that is therefore 'local').

Th group gr was already used for the agents 'occurring' in epistemic states and models. It serves a similar function on actions, whence the overloading. We need it, because in expressions $[\alpha]\varphi$, epistemic operators in φ *must* occur in epistemic states resulting from executing α (or they would have no meaning).

A nondeterministic action can have more than one execution in a given epistemic state. The only way to get such an action is to use $\cup$ (nondeterministic choice) operators in its description. If we use $!$ operators instead, typically, only some but not all of the agents are aware of the choices made. Constructs $\cup$ and $!$ are related by the type of an action:

The *type* t of an action is defined as $t(\alpha) := \alpha[!/\cup]$ (replace all $!$'s by $\cup$'s). Instead of $\alpha \ ! \ \alpha'$ we generally write $!\alpha \cup t(\alpha')$ or $t(\alpha')\cup \, !\alpha$. This expresses more clearly that given choice between α and α', the agents involved in those actions choose α, whereas that choice remains invisible to the agents that learn about these alternatives but are not involved.

EXAMPLE 7. *The description in* $\mathcal{L}_{12}^{\mathsf{act}}(\{p\})$ *of the actions in the introduction are:*

tell	$L_{12}?p \cup L_{12}?\neg p$
read	$L_{12}(L_1?p \cup L_1?\neg p)$
mayread	$L_{12}(L_1?p \cup L_1?\neg p \cup \ ?\top)$
bothmayread	$L_{12}(\ (L_1?p \cap L_2?p) \cup (L_1?\neg p \cap L_2?\neg p)$ $\cup L_1?p \cup L_1?\neg p \cup L_2?p \cup L_2?\neg p \cup \ ?\top)$

For example, the description of read *(Anne reads the letter) reads as follows: 'Anne and Bert learn that either Anne learns that she is invited for a night out in Amsterdam or that Anne learns that she has to give a lecture instead.' In the last two actions, instead of* $?\top$ *(for 'nothing happens') we may as well write* $?p\cup?\neg p$.

EXAMPLE 8. *The action* read *where Bert is seeing that Anne reads the letter is different from the action where Bert is seeing that Anne reads the letter* and Anne is actually invited for a night out. *The last is described as* $L_{12}(L_1?p \ ! \ L_1?\neg p)$*: of the two alternatives* $L_1?p$ *and* $L_1?\neg p$ *the first is chosen, but agent 2 is unaware of that choice. A different way of writing that action is* $L_{12}(!L_1?p\cup L_1?\neg p)$. *The action* read *is its* type. *Somewhat similarly, the action* bothmayread *has four different executions if* p *is true and another four if* p *is false: there are eight 'concrete' actions (state transformers) of that type.*

Semantics The semantics of $\mathcal{L}_N$ (on epistemic models) is defined as usual [17], plus an additional clause for the meaning of dynamic operators. The interpretation of a dynamic operator is a relation between an epistemic state and a set of epistemic states. The *composition* $(R \circ R')$ of two relations $R, R' : W \to \mathcal{P}(W)$ (such as $[\![\cdot]\!]$) is defined as follows: let $v \in W, V \subseteq W$, then: $(R \circ R')(v, V) :\Leftrightarrow \exists V' : R(v, V')$ and $\forall v' \in V' : \exists V'' \subseteq V : R'(v', V'')$ and $V = \bigcup_{v'\in V'}\{V'' \mid R'(v', V'')\}$. We write vR for $\{V \mid R(v,V)\}$. Further, $R \sqcup R' := \{(v,V) \mid \exists V', V'' : R(v,V'), R'(v,V''),$ and $V = V' \cup V''\}$.

In the semantics, we need a notion of equivalence between sets of epistemic states. We lift equivalence of worlds in a state to equivalence of states and to equivalence of sets of states. Sets of states will occur as worlds in definition 11 of action interpretation, and equivalence of such worlds for an agent will be defined as equivalence of those sets.

DEFINITION 9 (EQUIVALENCE OF SETS OF STATES). *Let* $M, M' \in \mathcal{S}5_N$, $v, w \in M$, *and* $w' \in M'$. *Let* $S, S' \subseteq \bullet\mathcal{S}5_{\subseteq N}$. *Let* $n \in N$. *Then:*

$$\begin{array}{rll} (M,w) \sim_n (M,v) & : \textit{iff} & w \sim_n v \\ (M,w) \sim_n (M',w') & : \textit{iff} & \exists v \in M : (M,v) \leftrightarrow (M',w') \\ & & \textit{and } (M,w) \sim_n (M,v) \\ S \sim_n S' & : \textit{iff} & [\ \forall s \in S : n \in gr(s) \Rightarrow \exists s' \in S' : \\ & & s \sim_n s' \ ] \textit{ and } [\ \forall s' \in S' : \\ & & n \in gr(s') \Rightarrow \exists s \in S : s \sim_n s' \ ] \end{array}$$

In the second clause of the definition, $\leftrightarrow$ stands for 'is bisimilar to' [7]. Bisimilarity is a notion of sameness between states that implies equivalence of their logical descriptions (theories), though not vice versa. The implicit symmetric closure in the third clause of the definition is needed to keep $\sim_n$ an equivalence relation.

We now continue with the semantics. The interpretation of formulas and actions is defined simultaneously.

Definition 10 (Interpretation of formulas). *Let $s = (M, w) \in \bullet\mathcal{S}5_N$ and $\varphi \in \mathcal{L}_N$, where $M = \langle W, \sim, V\rangle$. We define $M, w \models \varphi$ by inductive cases.*

$$
\begin{array}{lll}
M, w \models p & :\textit{iff} & w \in V(p) \\
M, w \models \neg\varphi & :\textit{iff} & M, w \not\models \varphi \\
M, w \models \varphi \wedge \psi & :\textit{iff} & M, w \models \varphi \textit{ and } M, w \models \psi \\
M, w \models K_n\varphi & :\textit{iff} & \forall w' : w' \sim_n w \Rightarrow M, w' \models \varphi \\
M, w \models C_B\varphi & :\textit{iff} & \forall w' : w' \sim_B w \Rightarrow M, w' \models \varphi \\
M, w \models [\alpha]\varphi & :\textit{iff} & \forall S \subseteq \bullet\mathcal{S}5_{\subseteq N} : (M, w)[\![\alpha]\!]S \Rightarrow \\
& & \exists s' \in S : s' \models \varphi
\end{array}
$$

Definition 11 (Interpretation of actions). *Let $\alpha \in \mathcal{L}_N^{\mathsf{act}}$. The interpretation $[\![\alpha]\!]$ of α in an arbitary epistemic state $s = (M, w) \in \bullet\mathcal{S}5_N$, where $M = \langle W, \sim, V\rangle$, is defined by inductive cases:*

$$
\begin{array}{rcl}
[\![?\varphi]\!] & := & \{(s, S) \mid S = \{(\langle W_\varphi, \emptyset, V|W_\varphi\rangle, w)\}\ \} \\
[\![L_B\alpha]\!] & := & \{(s, S) \mid \exists S' : s[\![\alpha]\!]S' \textit{ and} \\
& & S = \{(\langle W', \sim', V'\rangle, S')\}\ \} \\
[\![\alpha\ ;\ \alpha']\!] & := & [\![\alpha]\!] \circ [\![\alpha']\!] \\
[\![\alpha \cup \alpha']\!] & := & [\![\alpha]\!] \cup [\![\alpha']\!] \\
[\![\alpha\ !\ \alpha']\!] & := & [\![\alpha]\!] \\
[\![\alpha \cap \alpha']\!] & := & [\![\alpha]\!] \sqcup [\![\alpha']\!]
\end{array}
$$

In the clause for 'test', $W_\varphi = \{v \in \mathcal{D}(M) \mid M, v \models \varphi\}$. In the clause for 'learning': $W' := \{(M', v') \mid \exists v \in M : (M, v)[\![t(\alpha)]\!](M', v')\}$; for an arbitrary agent n, $\sim'_n := \sim_n$, seen as equivalence between sets of epistemic states; and for an arbitrary atom p: $S'' \in V'_p$ iff [for all $(\langle W'', \sim'', V''\rangle, w'') = s'' \in S''$, $w'' \in V''_p$].

The notion $\langle\alpha\rangle$ is dual to $[\alpha]$ and is defined as $s \models \langle\alpha\rangle\varphi$ iff [$\exists S \subseteq \bullet\mathcal{S}5_{\subseteq N} : s[\![\alpha]\!]S$ and $\forall s' \in S : s' \models \varphi$]. This may be intuitively more appealing: from the given state s, we can reach a set of of states S where φ holds everywhere ('concurrently'). Our treatment of the dynamic operators is similar to that in dynamic logic [21, 12].

A test results in an epistemic state without access for any agent. This is appropriate: how knowledge changes is only expressed in 'learning', so before we encounter a learn operator we cannot say anything at all about the knowledge of the agents in the state resulting from action execution: no access. One might as well say that, while compositionally interpreting an action, the computation of agents' knowledge is *deferred* until L operators are encountered.

Learning $L_B\alpha$ is defined in terms of $t(\alpha)$, and *this* is how local choice constructions $\alpha\ !\ \alpha'$ get their meaning from being bound by a learning operator. To execute an action $L_B\alpha$ in a state s, we do not just have to execute the *actual* action α in the *actual* state s, but also any *other* action of the same type $t(\alpha)$ as α in any *other* state s' with the same underlying model as s. The results are the *worlds* in the state that results from executing $L_B\alpha'$ in s. Such worlds (that are sets of states) cannot be distinguished from each other by an agent $n \in B$ if it could not distinguish their origins and it could not distinguish the actions resulting in those sets of states either. An appealing way to define the equivalence between worlds in the clause for 'learning' is to say that, for state transformers β, β' of type α (and by further lifting the notion of equivalence of sets of states to equivalence of relations – such as induced by action interpretation):

$$(M, v)[\![\beta]\!] \sim'_n (M, v')[\![\beta']\!] \ \text{ iff } \ v \sim_n v' \text{ and } [\![\beta]\!] \sim_n [\![\beta']\!]$$

The semantics may *appear* complex, because worlds in the model resulting from learning are actually sets of epistemic

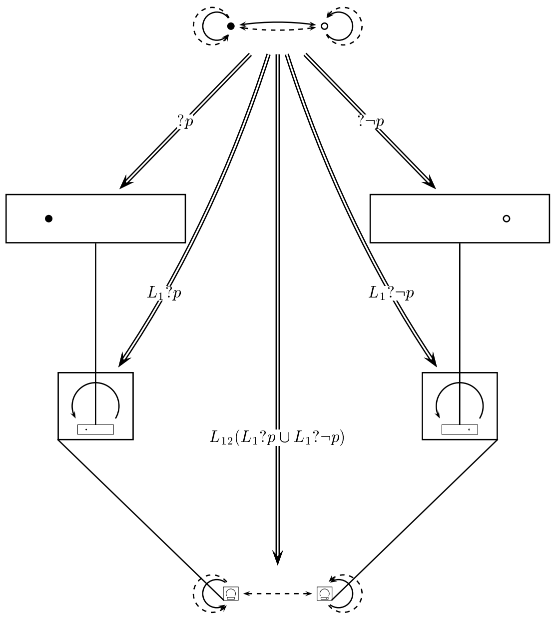

Figure 2: Details of the interpretation of action read **in** (Arc, u) **(see Figure 1). All access is visualized: access for 1 is a straight arrow, access for 2 is a dotted arrow. We have abstracted from the point of** Arc**, the left two and middle actions are actually executed in** (Arc, u) **and the right two in** (Arc, v)**. Linked boxes are identical.**

states. It is therefore important to realize that this is *merely a complex naming device* for worlds, but that the semantics is simple where it matters: the accessibility between worlds (simple: use $\sim_n$), and the value of atoms (simple: keep current value).

If the interpretation of α in s is not empty, we say that α is *executable* in s. For all actions except concurrent knowledge actions it is more intuitive to think of their interpretation as a relation between states than as a relation between a state and a set of states: if $s[\![\alpha]\!]\{s'\}$, we like to think of s' as the result of executing α in s. The notational abbreviation $s[\![\alpha]\!]s' :\Leftrightarrow s[\![\alpha]\!]\{s'\}$ allows us to keep using this helpful intuition. Further, if the interpretation is functional as well, write $s[\![\alpha]\!]$ for the unique s' such that $s[\![\alpha]\!]s'$. If this is the case for arbitrary s, we call α a *state transformer*. Note that tests are state transformers.

Example 12. *The interpretation of* $\mathsf{read} = L_{12}(L_1?p \cup L_1?\neg p)$ *on (Arc, u) (see Example 5) is defined in terms of the interpretation of $L_1?p \cup L_1?\neg p$ on (Arc, u) and (Arc, v). To interpret $L_1?p \cup L_1?\neg p$ on (Arc, u) we may either interpret $L_1?p$ or $L_1?\neg p$. Only the first can be executed. The interpretation of $L_1?p$ on (Arc, u) is defined in terms of the interpretation of $?p$ on any state (Arc, x) where $?p$ can be executed, i.e. where p holds, that is on (Arc, u); $(Arc, u)[\![?p]\!]$ is the singleton state consisting of world u without access. This epistemic state is therefore the single world in the domain of $(Arc, u)[\![L_1?p]\!]$. That world has reflexive access for 1, because*

$$(Arc, u)[\![?p]\!] \sim_1 (Arc, u)[\![?p]\!]$$

Etc. In the next and final stage of the interpretation, note that (as worlds)

$$(Arc, u)[\![L_1?p]\!] \sim_2 (Arc, u)[\![L_1?\neg p]\!]$$

because agent 2 does not occur in those states, but that

$$(Arc, u)[\![L_1?p]\!] \not\sim_1 (Arc, u)[\![L_1?\neg p]\!]$$

because $(Arc, u)[\![L_1?p]\!]$ is not bisimilar to $(Arc, u)[\![L_1?\neg p]\!]$. Further details have been omitted. See Figure 2, and Figure 1 for the other example actions.

Various algebraic action properties hold, such as associativity of $\cup$ and of $\cap$. The two main theorems of interest for the semantics are (for proofs see [29]):

THEOREM 13 (BISIM. IMPLIES MODAL EQUIVALENCE). *Let $\varphi \in \mathcal{L}_N$. Let s, s' be epistemic states. If $s \underline{\leftrightarrow} s'$, then $s \models \varphi \Leftrightarrow s' \models \varphi$.*

THEOREM 14 (ACTION EXECUTION PRESERVES BISIM.). *Let $\alpha \in \mathcal{L}_N^{\mathsf{act}}$ and $s, s' \in \bullet\mathcal{S}5_N$. For each $S \in \bullet\mathcal{S}5_{\subseteq N}$ there is a $S' \in \bullet\mathcal{S}5_{\subseteq N}$ and a bijection $f : S \to S'$ such that: If $s \underline{\leftrightarrow} s'$ and $s[\![\alpha]\!]S$, then $s'[\![\alpha]\!]S'$ and for all $s'' \in S : s'' \underline{\leftrightarrow} f(s'')$.*

A corollary of theorem 14 is the following:

COROLLARY 15. *Let $s, s' \in \bullet\mathcal{S}5_N$ and let $\alpha \in \mathcal{L}_N^{\mathsf{act}}$ be a state transformer that is executable in s. If $s \underline{\leftrightarrow} s'$, then $s[\![\alpha]\!] \underline{\leftrightarrow} s'[\![\alpha]\!]$.*

3. PROOF SYSTEM

In this section we present the proof system for concurrent dynamic epistemic logic. It is based on the dynamic epistic logics of [10] and [5], and on Concurrent PDL [21].

We first introduce some syntactic notions to be used in the proof system: the *set of state transformers* that are the concretizations or instances of an action, *syntactic equivalence* of actions, and the *precondition* of an action.

A nondeterministic action can have more than one execution in a given epistemic state. By replacing nondeterministic choice operators in a given action with local choice operators we get a concrete action, i.e. a state tranformer, of the same type.

DEFINITION 16. *Set $T(\alpha)$ is defined by induction on action structure (no details) with only nontrivial clause $T(\alpha \cup \alpha') := \{\beta \;!\; \beta' \mid \beta \in T(\alpha)$ and $\beta' \in T(\alpha')\} \cup \{\beta' \;!\; \beta \mid \beta \in T(\alpha)$ and $\beta' \in T(\alpha')\}$.*

LEMMA 17. *The interpretation of every action is equivalent to nondeterministic choice between all the state transformers that are its concretizations: $[\![\alpha]\!] = [\![\bigcup_{\beta \in T(\alpha)} \beta]\!]$.*

Given some α, set $T(t(\alpha))$ is the set of all concretizations of $t(\alpha)$, i.e. the *set of state transformers of type* α. We now use $T(t(\alpha))$ to define *syntactic* accessibility between actions: given the finite and effectively computable set of all actions of the same type, which of those are syntactically indistinguishable for an agent n? We omit details:

DEFINITION 18 (SYNTACTIC ACCESSIBILITY). *Given $\alpha \in \mathcal{L}_N^{\mathsf{act}}$ and $n \in N$, an equivalence relation $\sim_n$ induces a partition on $T(t(\alpha))$.*

EXAMPLE 19. *Consider action $L_{12}(!L_1?p \cup L_1?\neg p)$ (i.e., $L_{12}(L_1?p \;!\; L_1?\neg p)$). We have that $T(L_{12}(!L_1?p \cup L_1?\neg p)) = \{L_{12}(!L_1?p \cup L_1?\neg p)\}$ (it is already a state transformer), that $t(L_{12}(!L_1?p \cup L_1?\neg p)) = L_{12}(L_1?p \cup L_1?\neg p)$ and that $T(t(L_{12}(!L_1?p \cup L_1?\neg p))) = \{L_{12}(!L_1?p \cup L_1?\neg p), L_{12}(L_1?p \cup \;!L_1?\neg p)\}$: there are two state transformers of that type. Agent 1, but not agent 2, can tell learning p apart from learning $\neg p$:*

$$\begin{array}{lcl} L_{12}(!L_1?p \cup L_1?\neg p) & \not\sim_1 & L_{12}(L_1?p \cup \;!L_1?\neg p) \\ L_{12}(!L_1?p \cup L_1?\neg p) & \sim_2 & L_{12}(L_1?p \cup \;!L_1?\neg p) \end{array}$$

PROPOSITION 20. *Given are $\alpha \in \mathcal{L}_N^{\mathsf{act}}$, $n \in N$, and $\beta, \beta' \in T(t(\alpha))$. If $\beta \sim_n \beta'$, then $[\![\beta]\!] \sim_n [\![\beta']\!]$.*

In other words: syntactic equivalence of actions implies semantic equivalence. The converse does not hold! But we will see that there are 'just enough' indistinguishable actions to ensure completeness.

THEOREM 21 (PRESERVATION OF ACCESSIBILITY). *If $v, v' \in M \in \mathcal{S}5_N$ such that $v \sim_n v'$, and $\beta, \beta' \in \mathcal{L}_N^{\mathsf{act}}$ such that $\beta \sim_n \beta'$, and $(M, v)[\![\beta]\!]S$, then there is an S' such that $(M, v')[\![\beta']\!]S'$ and $S \sim_n S'$.*

COROLLARY 22. *For state transformers β, β' we have: if $v \sim_n v'$ and $\beta \sim_n \beta'$, then $(M, v)[\![\beta]\!] \sim_n (M, v')[\![\beta']\!]$.*

The last notion that we need in the proof system is that of the precondition of an action. If the precondition of an action holds, the action can be executed.

DEFINITION 23 (PRECONDITION).

$$\begin{array}{lcl} \mathsf{pre}(?\varphi) & := & \varphi \\ \mathsf{pre}(\alpha \;;\; \beta) & := & \mathsf{pre}(\alpha) \wedge \langle\alpha\rangle\mathsf{pre}(\beta) \\ \mathsf{pre}(\alpha \cup \beta) & := & \mathsf{pre}(\alpha) \vee \mathsf{pre}(\beta) \\ \mathsf{pre}(\alpha \cap \beta) & := & \mathsf{pre}(\alpha) \wedge \mathsf{pre}(\beta) \\ \mathsf{pre}(\alpha \;!\; \beta) & := & \mathsf{pre}(\alpha) \\ \mathsf{pre}(L_B\alpha) & := & \mathsf{pre}(\alpha) \end{array}$$

DEFINITION 24 (PROOF SYSTEM).

a *All propositional tautologies*
b $K_n(\varphi \to \psi) \to (K_n\varphi \to K_n\psi)$
c $K_n\varphi \to \varphi$
d $K_n\varphi \to K_nK_n\varphi$
e $\neg K_n\varphi \to K_n\neg K_n\varphi$
f $C_B\varphi \to (\varphi \wedge E_BC_B\varphi)$
g $[?\varphi]\psi \leftrightarrow (\varphi \to \psi)$
h $[\alpha \;;\; \alpha']\varphi \leftrightarrow [\alpha][\alpha']\varphi$
i $[\alpha \cup \alpha']\varphi \leftrightarrow ([\alpha]\varphi \wedge [\alpha']\varphi)$
j $[\alpha \cap \alpha']\varphi \leftrightarrow ([\alpha]\varphi \vee [\alpha']\varphi)$
k $\langle L_B\alpha\rangle\top \leftrightarrow \mathsf{pre}(L_B\alpha)$
l $[\alpha \;!\; \alpha']\varphi \leftrightarrow [\alpha]\varphi$
m $[\alpha]\varphi \leftrightarrow \bigwedge_{\beta \in T(\alpha)}[\beta]\varphi$
n $[\alpha]p \leftrightarrow (\mathsf{pre}(\alpha) \to p)$
o $(\varphi \wedge C_B(\varphi \to E_B\varphi)) \to C_B\varphi$
p $[\alpha]K_n\varphi \leftrightarrow (\mathsf{pre}(\alpha) \to K_n \bigwedge_{\alpha' \sim_n \alpha}[\alpha']\varphi)$

q *if φ and $\varphi \to \psi$, then ψ*
r *if φ, then $K_n\varphi$*
s *if $\varphi \to \psi$, then $[\alpha]\varphi \to [\alpha]\psi$*
t *if : for all $\alpha \sim_B \beta$ there is a χ_β such that:*
$\chi_\beta \to [\beta]\varphi$, and such that
$\beta \sim_n \alpha'$ implies $(\chi_\beta \wedge \mathsf{pre}(\beta)) \to E_B\chi_{\alpha'}$,
then $\chi_\alpha \to [\alpha]C_B\varphi$

A formula φ is deducable, abbreviated as $\vdash \varphi$, iff there exists a finite sequence of formulas such that each formula is either an instantiation of one of the axioms a – p, or if it is obtained by applying one of the rules q – t to formulas that appear earlier in the sequence.

THEOREM 25 (SOUNDNESS). *If $\vdash \varphi$, then $\models \varphi$.*

The soundness of most of the axioms is easily seen. For m we use lemma 17. For rule t some more work needs to be done. For this we need the following lemma (based on [5]).

LEMMA 26 (WITNESS PATH). *[$M, w \models \langle\alpha\rangle\hat{C}_B\neg\psi$] iff [there is a $\beta \sim_B \alpha$ such that $M, w \models \hat{C}_B\langle\beta\rangle\neg\psi$].*

EXAMPLE 27. $\vdash [L_{12}?p]C_{12}p$

1	$p \rightarrow p$	a
2	$[L_{12}?p]p \leftrightarrow (p \rightarrow p)$	n
3	$[L_{12}?p]p$	a,q 1, 2
4	$\top \rightarrow [L_{12}?p]p$	a,q 3
5	$\top$	a
6	$K_1\top$	r 5
7	$K_2\top$	r 5
8	$((\top \wedge p) \rightarrow K_1\top) \wedge ((\top \wedge p) \rightarrow K_2\top)$	a,q 6, 7
9	$\top \rightarrow [L_{12}?p]C_{12}p$	t 4, 8
10	$[L_{12}?p]C_{12}p$	q 5, 9

One might have expected a distribution axiom for $[\alpha]$, but this is not sound. Such an axiom is also unsound in the logic presented in [21], for the same reason: the interpretation of actions are relations between states and sets of states. The modality $[\alpha]$ correspond to a $\forall\exists$ quantifier and distribution does not hold for that. We do have a weaker form of distribution in the form of rule s. This is all we need in the completeness proof.

4. COMPLETENESS

The completeness proof is based on [5], [9], and [21]. Space does not permit us to go into much detail. We generally follow the structure of the proof in [5]. The main difficulty in the proof is the truth lemma, which is proven by induction on formulas. We show that every formula is provably equivalent to a formula in a sublanguage of the full language. The induction follows the structure of the formulas in the sublanguage. We first define the translation to the sublanguage.

DEFINITION 28 (TRANSLATION).

$$\begin{array}{lll}
f(p) & := & p \\
f(\neg\varphi) & := & \neg f(\varphi) \\
f(\varphi \wedge \psi) & := & f(\varphi) \wedge f(\psi) \\
f(K_n\varphi) & := & K_n f(\varphi) \\
f(C_B\varphi) & := & C_B f(\varphi) \\
f([?\varphi]\psi) & := & f(\varphi) \rightarrow f(\psi) \\
f([\alpha\ ;\ \beta]\psi) & := & f([\alpha]f([\beta]\psi)) \\
f([\alpha \cup \beta]\psi) & := & f([\alpha]\psi) \wedge f([\beta]\psi) \\
f([\alpha \cap \beta]\psi) & := & f([\alpha]\psi) \vee f([\beta]\psi) \\
f([\alpha\ !\ \beta]\psi) & := & f([\alpha]\psi) \\
f([L_B\alpha]p) & := & f(\mathsf{pre}(\alpha)) \rightarrow p \\
f([L_B\alpha]\neg\varphi) & := & \bigvee_{\beta \in T(L_B\alpha)} \neg f([\beta]\varphi) \\
f([L_B\alpha](\varphi \wedge \psi)) & := & f([L_B\alpha]\varphi) \wedge f([L_B\alpha]\psi) \\
f([L_B\alpha]K_n\varphi) & := & f(\mathsf{pre}(\alpha)) \rightarrow K_n \bigwedge_{\beta \sim_n L_B\alpha} f([\beta]\varphi) \\
f([L_B\alpha]C_B\varphi) & := & [L_B f(\alpha)]C_B f(\varphi) \\
f([L_B\alpha][\beta]\varphi) & := & f([L_B\alpha]f([\beta]\varphi)) \\
f(?\varphi) & := & ?f(\varphi) \\
f(\alpha\ ;\ \beta) & := & f(\alpha)\ ;\ f(\beta) \\
f(\alpha \cup \beta) & := & f(\alpha) \cup f(\beta) \\
f(\alpha \cap \beta) & := & f(\alpha) \cap f(\beta) \\
f(\alpha\ !\ \beta) & := & f(\alpha)\ !\ f(\beta) \\
f(L_B\alpha) & := & L_B f(\alpha)
\end{array}$$

LEMMA 29. $\vdash f(\varphi) \leftrightarrow \varphi$

The sublanguage has the following structure.

LEMMA 30. *Given a formula $\varphi \in \mathcal{L}_N$, we have that $f(\varphi) \in \mathcal{L}_N^f$, where $\mathcal{L}_N^f$ is defined by the following BNF:*

$$\omega ::= p \mid \neg\omega \mid \omega \wedge \omega \mid K_n\omega \mid C_B\omega \mid [L_B\alpha]C_B\omega$$

where, if ω' occurs in $L_B\alpha$, it is also of the form $\omega \in \mathcal{L}_N^f$.

The next lemma shows we can apply induction on this sublanguage in the truth lemma.

LEMMA 31 (WELL-FOUNDED ORDER). *There is a well-founded order $<$ on the language $\mathcal{L}_N$ with the following properties: (1) for any subformula ψ of φ, $\psi < \varphi$, (2) $<$ is transitive, (3) $f(\varphi) \leq \varphi$.*

We can now start constructing the canonical model. Because logics with reflexive transitive closure operators are generally not compact we need to construct a finite canonical model for every formula. That means we only look at maximally consistent sets with respect to some finite set of sentences. This set of sentences is called the closure.

DEFINITION 32 (CLOSURE). *Let $\omega \in \mathcal{L}_N^f$. The* closure *of ω is the minimal set $Cl(\omega) \subseteq \mathcal{L}_N^f$ such that*

1. *$\omega \in Cl(\omega)$*
2. *If $\omega' \in Cl(\omega)$ and ω'' is a subf. of ω', then $\omega'' \in Cl(\omega)$.*
3. *If $\omega' \in Cl(\omega)$ and ω' is not a negation, then $\neg\omega' \in Cl(\omega)$*
4. *If $C_B\omega' \in Cl(\omega)$, then $K_nC_B\omega' \in Cl(\omega)$, for all $a \in B$.*
5. *If $[\alpha]C_B\omega' \in Cl(\omega)$, then for all β and all $n \in B$ such that $\alpha \sim_n \beta$: $K_n[\beta]C_B\omega', [\beta]\omega' \in Cl(\omega)$.*

For any formula $\omega \in \mathcal{L}_N^f$, the closure of ω is finite. We will often write Ω for $Cl(\omega)$. Now we only look at maximally consistent sets in Ω. It is clear that any consistent subset of Ω can be expanded to a maximally consistent subset.

DEFINITION 33 (Ω-CANONICAL MODEL). *The Ω-canonical model is $M^\Omega = (W^\Omega, \sim^\Omega, V^\Omega)$ where*

$$\begin{array}{lll}
W^\Omega & = & \{\Gamma \subseteq \Omega : \Gamma \text{ is maximally } \Omega\text{-consistent}\} \\
\Gamma \sim_n^\Omega \Delta & \text{iff} & \{\omega' \in \Omega \mid K_n\omega' \in \Gamma\} \\
 & & = \\
 & & \{\omega' \in \Omega \mid K_n\omega' \in \Delta\} \\
V^\Omega(p) & = & \{\Gamma : p \in \Gamma\}
\end{array}$$

Note that M^Ω is finite: it contains at most $2^{|\Omega|}$ elements. Moreover, note that it is a model in S_B, where B is the group of agents that occur in φ, because the accessibility relations $\sim_n^\Omega$ are all equivalence relations. To ensure that the truth lemma holds for sentences of the form $[L_B\alpha]C_B\omega$ we need the following definition and lemma, which is also based on a similar definition and lemma found in [5].

DEFINITION 34 (GOOD PATH). *A good path from $\Gamma \in M^\Omega$ for $\langle\alpha\rangle\hat{C}_B\psi$ is a path in M^Ω*

$$\Gamma = \Gamma_0 \sim_{n_1} \Gamma_1 \sim_{n_2} \cdots \sim_{n_k} \Gamma_k$$

such that $k \geq 0$, $n_i \in N$ and there are actions α_i such that

$$\alpha = \alpha_0 \sim_{n_1} \alpha_1 \sim_{n_2} \cdots \sim_{n_k} \alpha_k$$

such that $\langle \alpha_i \rangle \top \in \Gamma_i$ $(0 \leq i \leq k)$ and $\langle \alpha_k \rangle \psi \in \Gamma_k$.

LEMMA 35. *Suppose $[\alpha] C_B \chi \in \Omega$. Then: there is a good path from Γ_0 for $\langle \alpha \rangle \hat{C}_B \neg \psi$, iff $\langle \alpha \rangle \hat{C}_B \neg \psi \in \Gamma_0$.*

This ensures that the truth lemma holds.

LEMMA 36 (TRUTH LEMMA). *If $\Gamma \in W_\Omega$, then for all $\gamma \in \Omega$ it holds that $(M_\Omega, \Gamma) \models \gamma$ iff $\gamma \in \Gamma$.*

Completeness follows in the standard way from the truth lemma.

THEOREM 37 (COMPLETENESS). *If $\not\vdash \varphi$, then there is a model (M, w) such that $(M, w) \not\models \varphi$*

PROOF. Suppose $\not\vdash \varphi$. Then, $\neg\varphi$ is consistent. Take $\omega = f(\neg\varphi)$. Note that $\neg\varphi$ and ω are provably equivalent (Lemma 29). Now, there is a maximally consistent set Γ in the closure Ω of ω such that $\omega \in \Gamma$. Because of the finite truth lemma we may conclude that $(M^\Omega, \Gamma) \models \omega$, and therefore (Lemma 29), $(M_\Omega, \Gamma) \models \neg\varphi$ and thus $(M_\Omega, \Gamma) \not\models \varphi$ □

5. APPLICATIONS

In various publications this language has been applied to describe the dynamics of concrete multiagent systems [26, 29, 28, 30]. We merely give an overview of application areas by examples, without much detail.

EXAMPLE 38 (CARD GAME ACTIONS). *Assume three players $1, 2, 3$ and three cards a, b, c. Each player is dealt one card. Atom a_1 represents the fact where card a is held by player 1, etc. The action where player 1 picks up his card, so that the others cannot see which card it is, is described by the action*

$$\mathsf{pickup} = L_{123}(L_1?a_1 \cup L_1?b_1 \cup L_1?b_1)$$

In some state s where each player is dealt one card and all players have picked up their cards (for details, see [29]), player 1 puts his card face up on the table. This is described by the action

$$\mathsf{table} = L_{123}?a_1 \cup L_{123}?b_1 \cup L_{123}?c_1$$

Note that in a given epistemic state only one of these alternatives can be executed. Now in that same state s we can also execute two rather different actions: firstly, player 1 can show his card to player 2 without player 3 seeing which card is shown. This action is described by

$$\mathsf{show} = L_{123}(L_{12}?a_1 \cup L_{12}?b_1 \cup L_{12}?c_1)$$

Next, player 2 can ask player 1 "please whisper in my ear the name of a card that you do not have," after which player 1 responds to 2's request. That action is described by

$$\mathsf{whisper} = L_{123}(L_{12}?\neg a_1 \cup L_{12}?\neg b_1 \cup L_{12}?\neg c_1)$$

In this case, whatever the actual state, 1 can choose one of two cards to whisper (and indeed, the complexity of the resulting epistemic state has now increased).

EXAMPLE 39 (CLUEDO). *The 'murder game' Cluedo is a card game where actions as in the previous example can take place. Other typical actions in Cluedo are 'ending your move' and 'winning the game'. For a perfect logician, ending a move in Cluedo is publicly announcing that you cannot win the game yet. This is the action $L_N?\neg\mathsf{win}_n$, where win_n is an epistemic formula describing knowledge of the 'murder cards', the cards 'held by the table' (agent 0) so to speak, i.e. $\mathsf{win}_n = K_n(\mathsf{scarlett}_0 \wedge \mathsf{knife}_0 \wedge \mathsf{kitchen}_0) \vee K_n \ldots$ (all murder cards combinations).*

EXAMPLE 40 (DIFFERENT CARDS). *Two players $1, 2$ face three cards a, b, c lying face-down in two stacks on the table. Let a be the atom describing 'card a is in the stack with two cards', etc. Some outsider hands player 1 one of the two-stack cards and, 'at the same time, with his other hand', player 2 the other ('different') card. The action is described by*

$$\mathsf{differentcards} = L_{12}(\bigcup_{x \neq y = a,b,c} (L_1?x \cap L_2?y))$$

EXAMPLE 41 (MUDDY CHILDREN). *We assume familiarity with the 'muddy children problem' [9]. Public announcement of φ corresponds in $\mathcal{L}_N$ to a knowledge action $L_N?\varphi$. Suppose there are three children 1, 2, and 3. First 'father' tells them that at least one of them is muddy. This is described by $L_{123}?(m_1 \vee m_2 \vee m_3)$ (where m_i stand for 'child i is muddy'). And then father tells them, that who knows whether (s)he is muddy may step forward. When nobody steps forward, that action $\mathsf{noforward}$ is generally [22, 11, 4] analysed as the public announcement of a conjunction describing that none of the children knows whether he/she is muddy: $L_{123}?((\neg K_1 m_1 \wedge \neg K_1 \neg m_1) \wedge (\neg K_2 m_2 \wedge \neg K_2 \neg m_2) \wedge (\neg K_3 m_3 \wedge \neg K_3 \neg m_3))$. We prefer an analysis where 'nobody steps forward' is composed of subprograms '1 does not step forward', '2 does not step forward' and '3 does not step forward':*

$$\begin{array}{rcll} \mathsf{noforward} & = & L_{123}(& L_{123}?(\neg K_1 m_1 \wedge \neg K_1 \neg m_1) \cap \\ & & & L_{123}?(\neg K_2 m_2 \wedge \neg K_2 \neg m_2) \cap \\ & & & L_{123}?(\neg K_3 m_3 \wedge \neg K_3 \neg m_3) \quad) \end{array}$$

EXAMPLE 42 (SECURITY PROTOCOLS). *From a pack of seven known cards $(0, 1, 2, 3, 4, 5, 6)$ two players Anne (a) and Bill (b) each draw three cards and a third player Crow (c) gets the remaining card. How can Anne and Bill openly (publicly) inform each other about their cards, without Crow learning from any of their cards who holds it? There are many solutions to this problem [30]. Suppose Anne actually holds {0,1,2} (012_a), Bill {3,4,5}, and Crow card 6. One of the solutions consists of Anne saying "My hand is one of $012, 034, 056, 135, 246$" after which Bill says "Crow has card 6". This is described by the sequence of two public announcements*

$$L_{abc}?K_a(012_a \vee 034_a \vee 056_a \vee 135_a \vee 246_a) \; ; \; L_{abc}?K_b 6_c$$

Hereafter, it is common knowledge that Anne knows Bill's cards, Bill knows Anne's cards, and Crow doesn't know any of Anne's or Bill's cards.

6. CONCLUSIONS

We have presented a proof system, and proved it to be sound and complete, for a dynamic epistemic logic in which higher-order information and belief change, and even higher-order belief change, can all be elegantly expressed. The crucial technical features of the language are, (1) that the notion of epistemic accessibility is lifted from one between worlds of an epistemic state to one between more complex semantic objects, such as sets of states, (2) the notion of the group of models, states, and actions, and (3) that actions are interpreted as a relation between states and sets of states. In view of proving completeness, we introduced a useful notion of syntactic access between actions. We gave an overview of the wide range of applications of this language for concrete MAS specification. We intend to continue this research by generalizing the semantics to include (not just knowledge but also) belief.

7. ACKNOWLEDGMENTS

Wiebe van der Hoek has carried out part of this research while visiting the University of Otago on a William Evans Fellowship. Barteld Kooi has been carrying out this research with the assistance of the Netherlands Organization for Scientific Research (NWO).

8. REFERENCES

[1] C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 100–109, Florida, October 1997.

[3] R. Aumann and A. Brandenburger. Epistemic conditions for Nash equilibrium. *Econometrica*, (63):1161–1180, 1995.

[4] A. Baltag. A logic for suspicious players: Epistemic actions and belief updates in games. *Bulletin of Economic Research*, 54(1):1–45, 2002.

[5] A. Baltag, L. Moss, and S. Solecki. The logic of public announcements, common knowledge and private suspicions. Presented at TARK 98, accepted for Annals of Pure and Applied Logic, 2002.

[6] S. Benferhat, D. Dubois, H. Prade, and M. Williams. A practical approach to revising prioritized knowledge bases. *Studia Logica*, 70(1), 2002.

[7] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge, 2001. Cambridge Tracts in Theoretical Computer Science 53.

[8] A. Darwiche and J. Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89(1-2):1–29, 1997.

[9] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge MA, 1995.

[10] J. Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, University of Amsterdam, 1999. ILLC Dissertation Series DS-1999-01.

[11] J. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language, and Information*, 6:147–169, 1997.

[12] R. Goldblatt. *Logics of Time and Computation*. CSLI Publications, Stanford, 2 edition, 1992. CSLI Lecture Notes No. 7.

[13] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, Cambridge MA, 2000. Foundations of Computing Series.

[14] J. Hintikka. *Knowledge and Belief*. Cornell University Press: Ithaca, NY, 1962.

[15] J. Hintikka. Reasoning about knowledge in philosophy. In J. Y. Halpern, editor, *Proceedings of TARK 1986*, pages 63–80. Morgan Kaufmann: San Mateo, CA, 1986.

[16] H. Katsuno and A. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings of KR '91*, pages 387–394, 1991.

[17] J.-J. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge Tracts in Theoretical Computer Science 41. Cambridge University Press, Cambridge, 1995.

[18] T. Meyer, W. Labuschagne, and J. Heidema. Refined epistemic entrenchment. *Journal of Logic, Language and Information*, 9:237–259, 2000.

[19] R. Parikh. The logic of games and its applications. In M. Karpinski and J. van Leeuwen, editors, *Topics in the theory of computation* Annals of Discrete Mathematics 24, pages 111–139, Amsterdam, 1985. Elsevier Science.

[20] M. Pauly. *Logic for social software*. PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series DS-2001-10.

[21] D. Peleg. Concurrent dynamic logic. *Journal of the ACM*, 34(2):450–479, 1987.

[22] J. Plaza. Logics of public communications. In M. Emrich, M. Pfeifer, M. Hadzikadic, and Z. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.

[23] B. ten Cate. Internalizing epistemic actions. In M. Martinez, editor, *Proceedings of the NASSLLI-2002 student session*, Stanford University, 2002.

[24] J. van Benthem. Logics for information update. In J. van Benthem, editor, *Proceedings of TARK VIII*, pages 51–88, Los Altos, 2001. Morgan Kaufmann.

[25] W. van der Hoek and M. Wooldridge. Tractable multi agent planning for epistemic goals. In C. Castelfranchi and W. Johnson, editors, *Proceedings of AAMAS*, pages 1167–1174, New York, USA, 2002. ACM Press.

[26] H. van Ditmarsch. *Knowledge games*. PhD thesis, University of Groningen, 2000. ILLC Dissertation Series DS-2000-06.

[27] H. van Ditmarsch. The semantics of concurrent knowledge actions. In M. Pauly and G. Sandu, editors, *ESSLLI 2001 workshop on Logic and Games*, 2001.

[28] H. van Ditmarsch. The description of game actions in cluedo. In L. Petrosian and V. Mazalov, editors, *Game Theory and Applications*, volume 8, pages 1–28, Commack, NY, USA, 2002. Nova Science Publishers.

[29] H. van Ditmarsch. Descriptions of game actions. *Journal of Logic, Language and Information*, 11:349–365, 2002.

[30] H. van Ditmarsch. The russian cards problem: a case study in cryptography with public announcements. Technical report, Department of Computer Science, University of Otago, 2002. OUCS-2002-08.