# Observing Distributed Computation a Dynamic-Epistemic Approach

Radu Mardare

Microsoft Research - University of Trento
Centre for Computational and Systems Biology, Trento, Italy
mardare@cosbi.eu

**Abstract.** We propose a new logic designed for modelling and reasoning about information flow and information exchange between spatially located interconnected *agents* witnessing a distributed computation. The intention is to trace the process of knowledge acquisition and its dynamics in the context of distributed systems. Underpinning on the dual algebraical-coalgebraical characteristics of process calculi, we design a decidable and completely axiomatized logic that combines the process-algebraical/equational and the modal/coequational features and is developed for process-algebraical semantics.

## 1 Introduction

Observation is fast becoming an important topic in computer science. In which manner can observation (in the broad sense of the word) be used for computing? In which way can the partial information available to an external observer of a computational system be used in deriving knowledge about the overall complete system? We will approach these problems by developing a logic designed to handle (partial) information flow and information exchange between external observers (agents) of a distributed system.

In the context of (parallel) distributed computation, a concurrent computational system can be thought of as being composed of a number of *modules*, i.e. spatially localized and independently observable units of behavior and computation (e.g. programs or processors running in parallel), organized in networks of subsystems and being able to interact, collaborate, communicate and interrupt each other. In this context we shall consider *agents* - *external observers* of the modules. As an external observer, an agent witnesses the computation of its module and interacts with the whole system only by means of it. Thus it derives its knowledge about the overall system from the observed behavior of its subsystem and from epistemic reasoning on the knowledge (and reactions) of the other agents witnessing (different) parts of the same computational process.

In this context we are interested in specifying when agents can receive, communicate or protect truthful information, when they improve their knowledge, when they are aware of the knowledge of the others and how they can construct strategies for influencing the others knowledge. Hence, the problem is

T. Mossakowski et al. (Eds.): CALCO 2007, LNCS 4624, pp. 379–393, 2007.

related to issues of privacy, trust, secured communications, authentication, etc. covering many different areas, with potential applications: in Secure Communication (checking privacy and authentication for given communication protocols by studying the knowledge acquisition strategies that an intruder might take), in Debugging and Performance analysis (checking for the cause of errors or of high computational costs in systems where only some modules are accessible), in Artificial Intelligence (endowing artificial agents with good and flexible tools to reason about their changing environment and about each other), in designing and improving strategies for knowledge acquisition over complex networks (such as the Internet), etc.

In approaching this problem we have chosen the process-algebraical representation of (mobile) distributed systems and we developed a logic of information flow for process-algebraical semantics. Taking process calculi as semantics is theoretically challenging due to their dual algebraical/coalgebraical nature. While the algebraical features of processes are naturally approached in equational fashion (that reflects, on logical level, the program constructors), the coalgebraical features (intrinsically related to transition systems via the denotational and the operational semantics of process calculi) ask for a modal/coequational treatment. The modal approach is also needed for the epistemic reasoning.

Consequently, our paper combines two logical paradigms to information flow in distributed systems: *dynamic-epistemic (and doxastic) logics* [14,9,12], semantically based on epistemic-doxastic Kripke models; and the spatial logics for concurrency [6], for which the semantics is usually given in terms of process algebra.

*Epistemic/doxastic logics* [14,9] are multimodal logics that formalize the epistemic notions of *knowledge*, or *belief*, possessed by an agent, or a group of agents, using modalities indexed by agents. We have modalities like $K_A\phi$ (*"A knows that $\phi$"*) or $\Box_A\phi$ (*"A justifiably believes that $\phi$"*) for any agent $A$. The models associate to each basic modality a binary relation interpreted as *"indistinguishability"* relation $\xrightarrow{A}$ for each agent $A$. It expresses the agent's uncertainty about the current state of the system. The states $s'$ such that $s \xrightarrow{A} s'$ are the epistemic alternatives of $s$ to agent $A$, i.e. if the current state is $s$, $A$ thinks that any of the alternatives $s'$ may be the current state.

*Dynamic logics* [12] are closer to process calculi as they have names for *"programs"*, or *"actions"*, and ways to combine them. In this case we have modalities indexed on a signature $\mathbb{A}$ (the set of programs). A dynamic modality $[\pi]\phi$ captures the weakest precondition of such a program w.r.t. a given post-specification $\phi$, and the accessibility relations are interpreted as transitions induced by programs. These logics already combine the coalgebraical features (modalities) with algebraical ones (the modalities have algebraic structures: programs are built using basic program constructors such as sequential composition or iteration).

*Dynamic Epistemic Logics* [1,2,11,3,8] are a class of logics that combine the dynamic and epistemic formalisms for specifying properties of evolving knowledge and beliefs in dynamic systems. The high level of expressivity reaches here a low complexity (decidability and complete axiomatizations).

*Spatial Logics.* Process semantics for modal logics can be considered as a special case of Kripke semantics, since, via transition systems, we can structure a class of processes as a Kripke model, by endowing it with accessibility relations induced by action transitions. Further we can use the standard clauses of Kripke semantics (e.g. Hennessy-Milner logic [13]). In addition, temporal, mobile and concurrent features have been added [22,7,20]. Spatial Logics [6] are the most expressive logics in this class containing equational operators to express spatial properties, such as the *parallel operator* $\phi|\psi$ and the *guarantee operator* $\phi \triangleright \psi$ (the adjoint of parallel), or operators for expressing the *"fresh name features"* inspired by the Gabbay-Pitts quantifier [10], etc.

The intention of this paper is to develop and study a logic that combines these two paradigms proposing a unified one. The new logic combines well with the process algebraical modelling of information flow and can directly express agent-dependent partial information features and their dynamics. We give a spatial interpretation of epistemic modalities in CCS. The intuition is to associate to each "agent" $A$ the process $P$ that describes the behavior of the module observed by $A$. The agent observing a process (possibly running in parallel with many other processes) *"knows"* only the activity and actions of its own process. *"Knowledge"* is thus defined as "information (about the overall, global process) that is locally available (to an agent)". In effect, this organizes any class $\mathcal{M}$ of processes (thought of as "states") as an epistemic Kripke model, with indistinguishability relations $\xrightarrow{A}$ for each agent $A$. Thus, if $A$ observes the subprocess $P$ then $P|P' \xrightarrow{A} P|P''$ for any $P'$, $P''$. Since these are equivalence relations, we obtain a notion of "(truthful) knowledge". The resulting Kripke modality, $K_A\phi$, read *"agent A knows $\phi$"*, holds at a given state (process) $R$ iff the process $P$ is active (as a subprocess) at $R$ and property $\phi$ holds in any context in which $P$ is active.

The resulting logic is completely axiomatizable and decidable. The Hilbert-style axiomatics we propose for it presents this logic as an authentic dynamic-epistemic logic. The classical axioms of knowledge will be present in our system together with spatial-like axioms.

### 1.1 Case Study: A Security Attack

For illustrating the problem we approach in this paper, we propose a toy example: a simplified *"Man-in-the-Middle"* type of cryptographic attack. Alice wants to communicate to Bob a secret over some communication channel. More concrete, she wants to inform Bob that a certain event $p$ happened. Before receiving the message from Alice, Bob considers both alternatives, $p$ and $\neg p$ equally possible. For communicating, Alice uses a key $k$ to encrypt her messages while Bob (and only Bob) knows how to decrypt them $(\overline{k})$[1]. But the communication channel is not secure: an evil outsider, Eve, has also access and her purpose is to make

[1] In a public-key cryptographic implementation, one could think of $k$ as being Bob's public key, while $\overline{k}$ is Bob's private key (for decryption). In a different context, $k$ might be Alice's password for communicating with Bob, which can only be authenticated by Bob using $\overline{k}$.

Bob believe $\neg p$. Suppose that Eve is also in possession of $k$ (either because $k$ was Bob's public key, or because Eve has somehow succeeded to steal Alice's password). Hence, she can present herself as Alice and is trying to convince Bob that $\neg p$. The communication of the secret event fails if Bob believes that the received message was from Alice and consequently $\neg p$ happened. In fact Eve manipulated Bob.

We are not concerned here primarily with the cryptographic details of the encryption method, but with the informational, "epistemic" features of this protocol. The main goal of it is to understand the epistemic status of the agents involved. What does Alice know? What does Bob know? What does Alice think that Bob knows? Does Alice know that her communication was unsuccessful? Does Eve know that her attack was successful? In which way the evolution of the system (of the processes involved) influences the information state of the agents? For proving the success or the failure of the protocol one has to show how Bob's knowledge can be influenced by Eve's actions and what can be done in order to avoid this.

One can use process algebras to describe such a scenario and logics for processes to specify properties of this protocol. But for answering to the previous questions, a way of arguing directly on the epistemic status of the agents is needed. We will prove further that despite of the apparent complexity of the epistemic reasoning on such frameworks there is a general approach that can formalize in a decidable manner the agents' reasoning in the above situation.

## 2 On Processes

In this section we introduce a fragment of CCS [19] calculus, that is "the core" of process calculi and will be used for defining the process-algebraical semantics for the logic. For the proofs of the results presented in this section and for additional results on this subject, the reader is referred to [16,18].

### 2.1 CCS Processes

**Definition 1.** *Let $\mathbb{A}$ be a denumerable signature. The syntax of the calculus is given by a grammar with one non-terminal symbol $P$ and the productions $P := 0 \mid \alpha.P \mid P|P$, where $\alpha \in \mathbb{A}$. We denote by $\mathbb{P}$ the language generated by this grammar. We call the elements of* $\mathbb{A}$ (basic) actions *and the objects in* $\mathbb{P}$ processes*.*

**Definition 2.** *Let $\equiv \subseteq \mathfrak{P} \times \mathfrak{P}$ be the smallest equivalence relation on $\mathbb{P}$ s.t.*
- $(\mathbb{P}, |, 0)$ *is a commutative monoid with respect to $\equiv$;*
- *if $P' \equiv P''$ then $\alpha.P' \equiv \alpha.P''$ and $P'|P \equiv P''|P$, for any $\alpha \in \mathbb{A}$ and $P \in \mathbb{P}$.*

*We call $\equiv$ structural congruence.*

**Definition 3.** *We call a process $P$* guarded *if $P \equiv \alpha.Q$ for some $\alpha \in \mathbb{A}$. We denote $P^0 \overset{def}{=} 0$ and $P^k \overset{def}{=} \underbrace{P|...|P}_{k}$.*

**Definition 4.** *We consider, on $\mathfrak{P}$, the labelled transition system defined by the rules in Table 1. We denote $P \longrightarrow Q$ if $P \xrightarrow{\alpha} Q$ or $P \xrightarrow{(\alpha,\overline{\alpha})} Q$ for some $\alpha \in \mathbb{A}$.*

**Table 1.** The transition system

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \qquad \frac{P \equiv Q \qquad P \xrightarrow{\alpha} P'}{Q \xrightarrow{\alpha} P'} \qquad \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\overline{\alpha}} Q'}{P|Q \xrightarrow{(\alpha,\overline{\alpha})} P'|Q}$$

*We write $P \xrightarrow{Q:\alpha} P'$ whenever $P \equiv Q|R$, $P' \equiv Q'|R$ and $Q \xrightarrow{\alpha} Q'$. We write $P|Q \xrightarrow{P:\alpha, Q:\overline{\alpha}} P'|Q'$ to denote the case when $P \xrightarrow{P:\alpha} P'$ and $Q \xrightarrow{Q:\alpha} Q'$. We call $(Q : \alpha)$ and $(P : \alpha, Q : \overline{\alpha})$* composed actions.

**Definition 5.** *We define for any process $P$, its set of actions $Act(P) \subset \mathbb{A}$:*
$1.Act(0) \overset{def}{=} \emptyset \quad 2.Act(\alpha.P) \overset{def}{=} \{\alpha\} \cup Act(P) \quad 3.Act(P|Q) \overset{def}{=} Act(P) \cup Act(Q)$
*For $M \subset \mathbb{P}$ we define $Act(M) \overset{def}{=} \bigcup_{P \in M} Act(P)$.*

**Definition 6.** *We call* action substitution *any mapping $\sigma : \mathbb{A} \longrightarrow \mathbb{A}$. We extend it, syntactically, to processes, $\sigma : \mathbb{P} \longrightarrow \mathbb{P}$, by*
1. $\sigma(0) \overset{def}{=} 0$ 2. $\sigma(P|Q) \overset{def}{=} \sigma(P)|\sigma(Q)$ 3. $\sigma(\alpha.P) \overset{def}{=} \sigma(\alpha).\sigma(P)$
*Let $act(\sigma) \overset{def}{=} \{\alpha, \beta \in \mathbb{A} \mid \alpha \neq \beta,\ \sigma(\alpha) = \beta\}$ and for $M \subset \mathbb{P}$, $\sigma(M) \overset{def}{=} \{\sigma(P) \mid P \in M\}$.*

We will also use $M^\sigma$, $P^\sigma$ for denoting $\sigma(M)$ and $\sigma(P)$.

### 2.2 Size of a Process

**Definition 7.** *The* size $[\![P]\!] = (h, w)$ *of a process $P \in \mathbb{P}$ is given by:*
1. $[\![0]\!] \overset{def}{=} (0, 0)$
2. $[\![P]\!] \overset{def}{=} (h, w)$ *iff $P \equiv (\alpha_1.Q_1)^{k_1}|...|(\alpha_j.Q_j)^{k_j}$ with $\alpha_i.Q_i \not\equiv \alpha_j.Q_j$ for $i \neq j$, where $h = 1 + max(h_1, .., h_k)$, $w = max(k_1, .., k_j, w_1, .., w_j)$ for $[\![Q_i]\!] = (h_i, w_i)$. We write $(h_1, w_1) \leq (h_2, w_2)$ for $h_1 \leq h_2$ and $w_1 \leq w_2$ and $(h_1, w_1) < (h_2, w_2)$ for $h_1 < h_2$ and $w_1 < w_2$.*

The intuition is that the size $(h, w)$ of a process is given by the depth of its syntactic tree (*height* $h$) and by the maximum number of bisimilar processes that can be found in a node of the syntactic tree (*width* $w$). By construction, the size of a process is unique up to structural congruence.

**Definition 8.** *For a set $M \subset \mathbb{P}$ we define*[2] $[\![M]\!] \overset{def}{=} max\{[\![P]\!] \mid P \in M\}$.

[2] The size of a set of processes is not always well-defined. An infinite set, for example, might not have the maximum required. However we will use this definition only where it is well-defined.

### 2.3 Structural Bisimulation

We introduce the *structural bisimulation*, a relation on processes that is an approximation of the structural congruence defined on size. It analyzes the behavior of a process focusing on a boundary of its syntactic tree. This relation is similar with the pruning relation proposed in [4].

**Definition 9.** *Let $P, Q \in \mathbb{P}$. We define $P \approx_h^w Q$ inductivelly by:*
*$P \approx_0^w Q$ always*
*$P \approx_{h+1}^w Q$ iff $\forall i \in 1..w$ and $\forall \alpha \in \mathbb{A}$ we have*

- *if $P \equiv \alpha.P_1|...|\alpha.P_i|P'$ then $Q \equiv \alpha.Q_1|...|\alpha.Q_i|Q'$ with $P_j \approx_h^w Q_j$, $j = 1..i$*
- *if $Q \equiv \alpha.Q_1|...|\alpha.Q_i|Q'$ then $P \equiv \alpha.P_1|...|\alpha.P_i|P'$ with $Q_j \approx_h^w P_j$, $j = 1..i$*

*We call $\approx_h^w$* structural bisimulation *on dimension $(h, w)$.*

**Proposition 1.** *$\approx_h^w$ is a congruence relation on processes having the properties:*
*1.(Antimonotonicity) if $P \approx_h^w Q$ and $(h', w') \leq (h, w)$ then $P \approx_{h'}^{w'} Q$.*
*2.(Inversion) if $P'|P'' \approx_h^{w_1+w_2} Q$ then $Q \equiv Q'|Q''$ and $P' \approx_h^{w_1} Q'$, $P'' \approx_h^{w_2} Q''$.*

**Proposition 2.** *1. If $[\![P]\!] \leq (h, w)$ and $[\![P']\!] \leq (h, w)$ then $P \approx_h^w P'$ iff $P \equiv P'$.*
*2. If $P \approx_h^w Q$ and $[\![P]\!] < (h, w)$ then $P \equiv Q$.*

Hence, for a well-chosen size which depends on the processes involved, the structural bisimulation guarantees the structural congruence. Reverse, the structural congruence implies the structural bisimulation.

**Proposition 3 (Behavioral simulation).** *Let $P \approx_h^w Q$.*
*1. If $P \xrightarrow{\alpha} P'$ then there exists a transition $Q \xrightarrow{\alpha} Q'$ s.t. $P' \approx_{h-1}^{w-1} Q'$.*
*2. If $[\![R]\!] < (h, w)$ and $P \xrightarrow{R:\alpha} P'$ then $Q \xrightarrow{R:\alpha} Q'$ and $P' \approx_{h-1}^{w-1} Q'$.*

This states that the structural bisimulation is preserved by transitions with the price of decreasing the size.

### 2.4 Bound Pruning Processes

In this subsection we prove that for a given process $P$ and a given size $(h, w)$ we can always find a process $Q$, having the size at most equal with $(h, w)$, such that $P \approx_h^w Q$. We will present a method for constructing $Q$ from $P$, by pruning the syntactic tree of $P$ to the given size.

**Theorem 1 (Bound pruning theorem).** *For any process $P \in \mathbb{P}$ and any size $(h, w)$ there exists a process $Q \in \mathbb{P}$ with $P \approx_h^w Q$ and $[\![Q]\!] \leq (h, w)$.*

*Proof.* We construct [3] $Q$ inductively on $h$.

**Case** $h = 0$**:** we take $Q \equiv 0$, as $P \approx_0^w Q$ and $[\![0]\!] = (0, 0)$.

**Case** $h + 1$**:** suppose $P \equiv \alpha_1.P_1|...|\alpha_n.P_n$.

Let $P_i'$ be the result of pruning $P_i$ by $(h, w)$ (the inductive step of construction)

[3] This construction is not necessarily unique.

and $P' \equiv \alpha_1.P'_1|...|\alpha_n.P'_n$. As for any $i = 1..n$ we have $P_i \approx^w_h P'_i$ (by the inductive hypothesis), we obtain, using Proposition 1, that $\alpha_i.P_i \approx^w_{h+1} \alpha_i.P'_i$, hence $P \approx^w_{h+1} P'$. Consider now $P' \equiv (\beta_1.Q_1)^{k_1}|...|(\beta_m.Q_m)^{k_m}$. Let $l_i = min(k_i, w)$ for $i = 1..m$. Further we define $Q \equiv (\beta_1.Q_1)^{l_1}|...|(\beta_m.Q_m)^{l_m}$. Obviously $Q \approx^w_{h+1} P'$ and as $P \approx^w_{h+1} P'$, we obtain $P \approx^w_{h+1} Q$. By construction, $[\![Q]\!] \leq (h+1, w)$.

**Definition 10.** *For a process $P$ and a tuple $(h, w)$ we denote by $P_{(h,w)}$ the process obtained by pruning $P$ to the size $(h, w)$ by the method described in the proof of Theorem 1.*

## 3 Sets of processes

In this section we study the *closed sets of processes* that will play an essential role in proving the finite model property for the logic we will introduce. Intuitively, a closed set of processes is a set that whenever contains a process contains also any future "state" of that process and any "observable" subpart of it (what an observer might see from it). Syntactically this means that whenever we have a process in a closed set, we will also have all the processes that can be obtained by arbitrarily pruning the syntactic tree of our process. For the proofs of the results presented in this section the reader is referred to [18].

**Definition 11.** *For $M, N \subset \mathbb{P}$ and $\alpha \in \mathbb{A}$ we define:*
$\alpha.M \overset{def}{=} \{\alpha.P \mid P \in M\}$ $\qquad$ $M|N \overset{def}{=} \{P|Q \mid P \in M, Q \in N\}$.

**Definition 12.** *For $P \in \mathbb{P}$ we define $\pi(P) \subset \mathbb{P}$ inductively by:*
1. $\pi(0) \overset{def}{=} \{0\}$ $\qquad$ 2. $\pi(\alpha.P) \overset{def}{=} \{0\} \cup \alpha.\pi(P)$ $\qquad$ 3. $\pi(P|Q) \overset{def}{=} \pi(P)|\pi(Q)$
*We extend the definition of $\pi$ to sets of processes $M \subset \mathbb{P}$ by $\pi(M) \overset{def}{=} \bigcup_{P \in M} \pi(P)$.*

Thus, we associate to each process $P$ the set $\pi(P)$ of all processes obtained by arbitrarily pruning the syntactic tree of $P$.

**Definition 13.** *A set of processes $\mathcal{M} \subseteq \mathbb{P}$ is closed if it satisfies the conditions*
*1. if $P \in \mathcal{M}$ and $P \longrightarrow P'$ then $P' \in \mathcal{M}$ $\qquad$ 2. if $P \in \mathcal{M}$ then $\pi(P) \subset \mathcal{M}$.*
*We say that $\mathcal{M}$ is the closure of $M \subset \mathbb{P}$ if $\mathcal{M}$ is the smallest closed set of processes that contains $M$. We write $\overline{M} = \mathcal{M}$.*
*For any closed set $\mathcal{M}$ and any $(h, w)$ we define $\mathcal{M}_{(h,w)} \overset{def}{=} \overline{\{P_{(h,w)} \mid P \in \mathcal{M}\}}$.*
*For $A \subset \mathbb{A}$ we define $\mathfrak{M}^A_{(h,w)} \overset{def}{=} \{\overline{M} \subset \mathbb{P} \mid Act(M) \subseteq A,\ [\![M]\!] \leq (h, w)\}$.*

**Lemma 1.** *If $A \subset \mathbb{A}$ is a finite set of actions, then the following hold:*
*1. If $\mathcal{M} \in \mathfrak{M}^A_{(h,w)}$ then $\mathcal{M}$ is a finite closed set of processes.*
*2. $\mathfrak{M}^A_{(h,w)}$ is finite.*

The previous result shows that the set of closed sets of processes with actions from a given finite signature $A$ and dimension not bigger than $(h, w)$ is finite.

**Definition 14.** *Let $\mathcal{M},\mathcal{N} \subset \mathbb{P}$ be closed sets. We write $\mathcal{M} \approx_h^w \mathcal{N}$ iff*
*1. for any $P \in \mathcal{M}$ there exists $Q \in \mathcal{N}$ with $P \approx_h^w Q$*
*2. for any $Q \in \mathcal{N}$ there exists $P \in \mathcal{M}$ with $P \approx_h^w Q$*
*We write $(\mathcal{M}, P) \approx_h^w (\mathcal{N}, Q)$ when $P \in \mathcal{M}$, $Q \in \mathcal{N}$, $P \approx_h^w Q$ and $\mathcal{M} \approx_h^w \mathcal{N}$.*

Further we state that having a closed set $\mathcal{M}$ with actions from $A$ and a dimension $(h, w)$ we can always find, in the finite set $\mathfrak{M}^{A}_{(h,w)}$, a closed set $\mathcal{N}$ structural bisimilar with $\mathcal{M}$ at the dimension $(h, w)$.

**Proposition 4.** *For any closed set of processes $\mathcal{M}$, and any size $(h, w)$ we have $\mathcal{M}_{(h,w)} \approx_w^h \mathcal{M}$.*

**Theorem 2 (Bound pruning theorem).** *Let $\mathcal{M}$ be a closed set of processes. Then for any $(h, w)$ there is a closed set $\mathcal{N} \in \mathfrak{M}^{Act(\mathcal{M})}_{(h,w)}$ such that $\mathcal{M} \approx_h^w \mathcal{N}$.*

## 4 The Logic $\mathcal{L}^{\mathfrak{A}}_{\mathbb{A}}$

In this section we introduce the logic $\mathcal{L}^{\mathfrak{A}}_{\mathbb{A}}$ with multimodal operators indexed by "epistemic agents" from a signature $\mathfrak{A}$ and by "transition actions" from a signature $\mathbb{A}$. The proofs of the results presented further can be consulted in [18].

### 4.1 Syntax of $\mathcal{L}^{\mathfrak{A}}_{\mathbb{A}}$

**Definition 15.** *Consider a set $\mathcal{A}$ and its extension $\mathcal{A}^+$ generated, for arbitrary $\alpha \in \mathbb{A}$ and $e \in \mathcal{A}$, by $E := e \mid \alpha.E \mid E|E$. In addition, on $\mathcal{A}^+$ it is defined the smallest congruence relation $\equiv$ for which $|$ is commutative and associative. We call the $\equiv$-equivalence classes of $\mathcal{A}^+$* epistemic agents *and we call* atomic agents *the classes corresponding to elements of $\mathcal{A}$. For $E \in \mathcal{A}^+$ we denote by $\overline{E}$ its $\equiv$-equivalence class.*
*A* society of agents *is a set $\mathfrak{A}$ of epistemic agents satisfying the conditions*
*1. if $\overline{E_1|E_2} \in \mathfrak{A}$ then $\overline{E_1}, \overline{E_2} \in \mathfrak{A}$ 2. if $\overline{\alpha.E} \in \mathfrak{A}$ then $\overline{E} \in \mathfrak{A}$*
*Hereafter we denote by $A, A', A_1, ...$ arbitrary epistemic agents and we consider the canonical extension of the operators $|$ and $\alpha.$ from $\mathcal{A}^+$ to the epistemic agents.*

**Definition 16.** *Let $\mathfrak{A}$ be a society of epistemic agents defined for the set $\mathbb{A}$ of actions. We define the language $\mathcal{F}^{\mathfrak{A}}_{\mathbb{A}}$ of $\mathcal{L}^{\mathfrak{A}}_{\mathbb{A}}$, for $A, A' \in \mathfrak{A}$ and $\alpha \in \mathbb{A}$ by:*
$\phi := 0 \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi|\phi \mid \langle a\rangle\phi \mid K_A\phi$, *where*
$\langle a\rangle := \langle\alpha\rangle \mid \langle\alpha, \overline{\alpha}\rangle \mid \langle A : \alpha\rangle \mid \langle A, A' : \alpha\rangle$.

### 4.2 Process Semantics

A formula of $\mathcal{F}^{\mathfrak{A}}_{\mathbb{A}}$ will be evaluated to processes in a given closed set of processes, by using the satisfaction relation $\mathcal{M}, P \models \phi$.

**Definition 17.** *A model of* $\mathcal{L}^{\mathfrak{A}}_{\mathbb{A}}$ *is a pair* $(\mathcal{M}, I)$ *where* $\mathcal{M}$ *is a closed set of processes and* $I : (\mathfrak{A}, |, \alpha.) \longrightarrow (\mathcal{M}, |, \alpha.)$ *a homomorphism*[4] *of structures such that* $I(A) = 0$ *iff* $A = \overline{e}$ *for some* $e \in \mathcal{A}$.

*We denote* $P \overset{I(A):\alpha}{\longrightarrow} Q$ *by* $P \overset{A:\alpha}{\longrightarrow} Q$ *and* $P \overset{(I(A):\alpha, I(B):\overline{\alpha})}{\longrightarrow} Q$ *by* $P \overset{A,B:\alpha}{\longrightarrow} Q$. *Let* $\mathbb{A}^{\mathfrak{A}} = \mathbb{A} \cup \{(\alpha, \overline{\alpha}) \mid \alpha \in \mathbb{A}\} \cup \{(A : \alpha), (A, A' : \alpha) \mid \alpha \in \mathbb{A}, A, A' \in \mathfrak{A}\}$ *and* $a \in \mathbb{A}^{\mathfrak{A}}$ *an arbitrary element. For* $P \in \mathcal{M}$ *and* $\phi \in \mathcal{F}^{\mathfrak{A}}_{\mathbb{A}}$ *we define* $\mathcal{M}, P \models \phi$ *by:*
$\mathcal{M}, P \models \top$ *always.*
$\mathcal{M}, P \models 0$ *iff* $P \equiv 0$.
$\mathcal{M}, P \models \neg\phi$ *iff* $\mathcal{M}, P \not\models \phi$.
$\mathcal{M}, P \models \phi \wedge \psi$ *iff* $\mathcal{M}, P \models \phi$ *and* $\mathcal{M}, P \models \psi$.
$\mathcal{M}, P \models \phi|\psi$ *iff* $P \equiv Q|R$ *and* $\mathcal{M}, Q \models \phi$, $\mathcal{M}, R \models \psi$.
$\mathcal{M}, P \models \langle a \rangle \phi$ *iff there exists a transition* $P \overset{a}{\longrightarrow} P'$ *such that* $\mathcal{M}, P' \models \phi$.
$\mathcal{M}, P \models K_A\phi$ *iff* $P \equiv I(A)|R$ *and for all* $I(A)|R' \in \mathcal{M}$ *we have* $\mathcal{M}, I(A)|R' \models \phi$.

**Definition 18 (Derived operators).** *In addition to the classical boolean operators, we introduce other derived operators:*

$$1 \overset{def}{=} \neg((\neg 0) \mid (\neg 0)) \qquad \alpha.\psi \overset{def}{=} (\langle \alpha \rangle \psi) \wedge 1$$
$$[a]\phi \overset{def}{=} \neg(\langle a \rangle(\neg\phi)) \qquad \widetilde{K}_A\phi \overset{def}{=} \neg K_A \neg\phi.$$

*We use the precedence order* $\neg$, $K_A$, $\langle a \rangle$, $|, \wedge, \vee, \rightarrow$ *for the operators, where* $\neg$ *has precedence over all.*

The semantics of the derived operators will be:
$\mathcal{M}, P \models [a]\phi$ iff for any transition $P \overset{a}{\longrightarrow} P'$ (if any) we have $\mathcal{M}, P' \models \phi$
$\mathcal{M}, P \models 1$ iff $P \equiv 0$ or $P \equiv \alpha.Q$
$\mathcal{M}, P \models \alpha.\phi$ iff $P \equiv \alpha.Q$ and $\mathcal{M}, Q \models \phi$
$\mathcal{M}, P \models \widetilde{K}_A\phi$ iff either $P \not\equiv I(A)|R$ for any $R$, or $\exists I(A)|S \in \mathcal{M}$ such that $\mathcal{M}, I(A)|S \models \phi$

Remark the interesting semantics of the operators $K_A$ and $\widetilde{K}_A$ for $A \in I^{-1}(0)$:
$\mathcal{M}, P \models K_A\phi$ iff $\forall Q \in \mathcal{M}$ we have $\mathcal{M}, Q \models \phi$
$\mathcal{M}, P \models \widetilde{K}_A\phi$ iff $\exists Q \in \mathcal{M}$ such that $\mathcal{M}, Q \models \phi$

Hence $K_A\phi$ and $\widetilde{K}_A\phi$ for an atomic agent $A$ encode, in syntax, the validity and the satisfiability with respect to a given model.

### 4.3 Bounded Finite Model Property

**Definition 19.** *The* sizes of a formula *(height and width)* $(\!|\phi|\!) = (h, w)$ *w.r.t. the homomorphism* $I$ *is given inductively in Table 2.*

**Lemma 2.** *If* $(\!|\phi|\!) = (h, w)$, $\mathcal{M}, P \models \phi$ *and* $(\mathcal{M}, P) \approx^w_h (\mathcal{N}, Q)$ *then* $\mathcal{N}, Q \models \phi$.

Hence $\phi$ is *"sensitive"* via satisfaction only up to size $(\!|\phi|\!)$. In other words, the relation $\mathcal{M}, P \models \phi$ is conserved by substituting the pair $(M, P)$ with any other

[4] The function $I$ associates to each agent the process it observes. An atomic agent sees always the process 0.

**Table 2.** Sizes of formulas

Suppose $\llparenthesis\phi\rrparenthesis = (h, w)$, $\llparenthesis\psi\rrparenthesis = (h', w')$, $\llbracket I(A)\rrbracket = (h_A, w_A)$ and $\llbracket I(A,)\rrbracket = (h_{A'}, w_{A'})$, then

$1.\llparenthesis 0\rrparenthesis = \llparenthesis\top\rrparenthesis \overset{def}{=} (0, 0)$ $\quad 2.\llparenthesis\neg\phi\rrparenthesis \overset{def}{=} \llparenthesis\phi\rrparenthesis$ $\quad 3.\llparenthesis\phi \wedge \psi\rrparenthesis \overset{def}{=} (max(h, h'), max(w, w'))$

$4.\llparenthesis\phi|\psi\rrparenthesis \overset{def}{=} (max(h, h'), w + w')$ $\quad 5.\llparenthesis\langle\alpha\rangle\phi\rrparenthesis = \llparenthesis\langle\alpha, \overline{\alpha}\rangle\phi\rrparenthesis \overset{def}{=} (1 + h, 1 + w)$

$6.\llparenthesis\langle A : \alpha\rangle\phi\rrparenthesis \overset{def}{=} (1 + max(h, h_A), 1 + max(w, w_A))$

$7.\llparenthesis K_A\phi\rrparenthesis \overset{def}{=} (1 + max(h, h_A), 1 + max(w, w_A))$

$8.\llparenthesis\langle A, A' : \alpha\rangle\phi\rrparenthesis \overset{def}{=} (1 + max(h, h_A, h_{A'}), 1 + max(w, w_A, w_{A'}))$

pair $(N, P)$ structurally bisimilar to it at the size $\llparenthesis\phi\rrparenthesis$. Using this result, we conclude that if a process satisfies $\phi$ w.r.t. a given closed set of processes, then by pruning the process and the closed set on the size $\llparenthesis\phi\rrparenthesis$, we preserve the satisfiability for $\phi$. Indeed the theorems 1 and 4 prove that if $\llparenthesis\phi\rrparenthesis = (h, w)$ then $(\mathcal{M}, P) \approx_w^h (\mathcal{M}_{\llparenthesis\phi\rrparenthesis}, P_{\llparenthesis\phi\rrparenthesis})$. Hence $\mathcal{M}, P \models \phi$ implies $\mathcal{M}_{\llparenthesis\phi\rrparenthesis}, P_{\llparenthesis\phi\rrparenthesis} \models \phi$.

**Definition 20.** *The set of actions of a formula $\phi$ is defined in Table 3.*

**Table 3.** The set of actions of a formula

1. $act(0) = act(\top) \overset{def}{=} \emptyset$
2. $act(\langle\alpha\rangle\phi) \overset{def}{=} \{\alpha\} \cup act(\phi)$
3. $act(\neg\phi) = act(\phi)$
4. $act(\langle\alpha, \overline{\alpha}\rangle\phi) \overset{def}{=} \{\alpha, \overline{\alpha}\} \cup act(\phi)$
5. $act(\phi \wedge \psi) = act(\phi|\psi) \overset{def}{=} act(\phi) \cup act(\psi)$
6. $act(\langle A : \alpha\rangle\phi) = act(K_A\phi) \overset{def}{=} Act(I(A)) \cup act(\phi)$
7. $act(\langle A, A' : \alpha\rangle\phi) \overset{def}{=} Act(I(A)) \cup Act(I(A')) \cup act(\phi)$

The next result states that a formula $\phi$ does not reflect properties that involve more then the actions in its syntax. Thus if $\mathcal{M}, P \models \phi$ then any substitution $\sigma$ having the elements of $act(\phi)$ as fix points preserves the satisfaction relation.

**Lemma 3.** *If $\mathcal{M}, P \models \phi$ and $\sigma$ a substitution with $act(\sigma) \bigcap act(\phi) = \emptyset$ then $\mathcal{M}^\sigma, P^\sigma \models \phi$.*

Consider a lexicographical order $\ll$ on $\mathbb{A}$. For a finite set $B \subset \mathbb{A}$ there exists a unique maximal element. We denote by $B_+$ the set obtained by adding to $B$ the successor, w.r.t. $\ll$, of its maximal element.

**Lemma 4 (Finite model property).** *If $\mathcal{M}, P \models \phi$ then $\exists\mathcal{N} \in \mathfrak{M}_{\llparenthesis\phi\rrparenthesis}^{act(\phi)+}$ and $Q \in \mathcal{N}$ such that $\mathcal{N}, Q \models \phi$.*

Because $act(\phi)$ is finite, Theorem 1 states that $\mathfrak{M}_{\llparenthesis\phi\rrparenthesis}^{act(\phi)+}$ is finite and any closed set $\mathcal{M} \in \mathfrak{M}_{\llparenthesis\phi\rrparenthesis}^{act(\phi)+}$ is finite as well. Thus we obtain the finite model property for our logic. A consequence of theorem 4 is the decidability for satisfiability, validity and model checking against the process semantics.

**Theorem 3 (Decidability).** *For $\mathcal{L}_{\mathbb{A}}^{\mathfrak{A}}$ validity, satisfiability and model checking are decidable against process semantics.*

### 4.4 Characteristic Formulas

In this subsection we use the peculiarities of the dynamic and epistemic operators to define characteristic formulas for processes and for finite closed sets of processes.

**Definition 21.** *Consider the class of logical formulas indexed by ($\equiv$-equivalence classes of) processes $\mathcal{F}_{\mathbb{P}} = \{(f_P) \mid P \in \mathbb{P}\}$ defined as follows*[5]*:*

1. $f_0 \overset{def}{=} 0$ 2. $f_{P|Q} \overset{def}{=} f_P|f_Q$ 3. $f_{\alpha.P} \overset{def}{=} \alpha.f_P$

**Proposition 5.** *$f_P$ is a characteristic formula for $P$, i.e. $\mathcal{M}, P \models f_Q$ iff $P \equiv Q$.*

**Definition 22.** *Consider the class of logical formulas indexed by epistemic agents $\mathcal{F}_{\mathfrak{A}}$ defined as follows*[6]*: Similarly we introduce a class of logical formulas $(f_A)_{A\in\mathfrak{A}}$, on epistemic agents*

1. $f_A \overset{def}{=} 0$ *if $A$ is atomic agent* 2. $f_{A_1|A_2} \overset{def}{=} f_{A_1}|f_{A_2}$ 3. $f_{\alpha.A} \overset{def}{=} \alpha.f_A$

**Proposition 6.** *$\mathcal{M}, P \models f_A$ iff $P \equiv I(A)$.*

**Definition 23.** *Let $\Phi \subset \mathcal{F}^{\mathfrak{A}}$ be a finite set of formulas and $A \in \mathfrak{A}$ an atomic agent. Let $\Delta\Phi \overset{def}{=} K_A(\bigvee_{\phi\in\Phi}\phi) \wedge (\bigwedge_{\phi\in\Phi}\tilde{K}_A\phi)$.*

Observe that $\mathcal{M}, P \models \Delta\Phi$ iff for any $Q \in \mathcal{M}$ there exists $\phi \in \Phi$ such that $\mathcal{M}, Q \models \phi$ and for any $\phi \in \Phi$ there exists $Q \in \mathcal{M}$ such that $\mathcal{M}, Q \models \phi$. Observe also that it is irrelevant which atomic agent $A$ we choose to define $\Delta$, as the epistemic operators of any atomic agent can encode validity and satisfiability.

Further we exploit the semantics of this operator for defining characteristic formulas for finite closed sets of processes.

**Definition 24.** *For a finite closed set of processes $\mathcal{M}$ let $f_{\mathcal{M}} = \Delta\{f_P \mid P \in \mathcal{M}\}$.*

**Proposition 7.** *If $\mathcal{M}$,$\mathcal{N}$ are finite closed sets of processes and $P \in \mathcal{M}$ then $\mathcal{M}, P \models f_{\mathcal{N}}$ iff $\mathcal{N} = \mathcal{M}$.*

### 4.5 Axiomatic System

Consider the subset of logical formulas given by $f := \alpha.0 \mid \alpha.f \mid f|f$ for $\alpha \in \mathbb{A}$. We denote the class of these formulas by $\mathcal{F}$[7]. Hereafter we use $f, g, h$ for denoting arbitrary formulas from $\mathcal{F}$, while $\phi, \psi, \rho$ will be used for formulas in $\mathcal{F}_{\mathbb{A}}^{\mathfrak{A}}$.

**Proposition 8.** $\mathcal{F} \cup \{0\} = \mathcal{F}_{\mathbb{P}}$.

In table 4 is proposed a Hilbert-style axiomatic system for $\mathcal{L}_{\mathbb{A}}^{\mathfrak{A}}$. We assume the axioms and the rules of propositional logic. In addition we have a set of spatial axioms

**Table 4.** The axiomatic system $\mathcal{L}_{\mathbb{A}}^{\mathfrak{A}}$

| | |
|---|---|
| **Spatial axioms** | |
| S1: $\vdash \top \mid \bot \rightarrow \bot$ | S4: $\vdash \phi \mid (\psi \vee \rho) \rightarrow (\phi \mid \psi) \vee (\phi \mid \rho)$ |
| S2: $\vdash (\phi \mid \psi) \mid \rho \rightarrow \phi \mid (\psi \mid \rho)$ | S5: $\vdash \phi \mid \psi \rightarrow \psi \mid \phi$ |
| S3: $\vdash \phi \mid 0 \leftrightarrow \phi$ | S6: $\vdash (f \wedge \phi \mid \psi) \rightarrow \bigvee_{f \leftrightarrow g \mid h} (g \wedge \phi) \mid (h \wedge \psi)$ |
| **Spatial rules** | |
| SR1: If $\vdash \phi \rightarrow \psi$ then $\vdash \phi \mid \rho \rightarrow \psi \mid \rho$ | |
| **Dynamic axioms** | |
| D1: $\vdash \langle a \rangle \phi \mid \psi \rightarrow \langle a \rangle (\phi \mid \psi)$ | D3: $\vdash 0 \vee \langle !\alpha \rangle \top \rightarrow [\beta] \bot$, for $\alpha \neq \beta$ |
| D2: $\vdash [a](\phi \rightarrow \psi) \rightarrow ([a]\phi \rightarrow [a]\psi)$ | D4: $\vdash \langle !\alpha \rangle \phi \rightarrow [\alpha]\phi$ |
| **Dynamic rules** | |
| DR1: If $\vdash \phi$ then $\vdash [a]\phi$ | DR2: If $\vdash \phi_1 \rightarrow [a]\phi_1'$ and $\vdash \phi_2 \rightarrow [a]\phi_2'$<br>*then* $\vdash \phi_1 \mid \phi_2 \rightarrow [a](\phi_1' \mid \phi_2 \vee \phi_1 \mid \phi_2')$ |
| **Epistemic axioms** | |
| E1: $\vdash K_A\phi \wedge K_A(\phi \rightarrow \psi) \rightarrow K_A\psi$ | |
| E2: $\vdash K_A\phi \rightarrow \phi$ | E4: $\vdash K_A\top \rightarrow (\neg K_A\phi \rightarrow K_A \neg K_A \phi)$ |
| E3: $\vdash K_A\phi \rightarrow K_A K_A \phi$ | E5: $\vdash K_A\top \leftrightarrow f_A \mid \top$ |
| **Axioms involving atomic agents** $A_0 \in \mathcal{A}$ | |
| E6: $\vdash K_A\phi \leftrightarrow (K_A\top \wedge K_{A_0}(K_A\top \rightarrow \phi))$ | E8: $\vdash K_{A_0}\phi \rightarrow [a]K_{A_0}\phi$ |
| E7: $\vdash K_{A_0}\phi \wedge \psi \mid \rho \rightarrow (K_{A_0}\phi \wedge \psi) \mid (K_{A_0}\phi \wedge \rho)$ | E9: $\vdash K_{A_0}\phi \rightarrow (K_A\top \rightarrow K_A K_{A_0}\phi)$ |
| **Epistemic rules** | |
| ER1: If $\vdash \phi$ then $\vdash K_A\top \rightarrow K_A\phi$ | |
| **Mixed axioms** | |
| M1: $\vdash \langle A : \alpha \rangle \top \rightarrow K_A\top$ | M3: $\vdash \langle A : \alpha \rangle \phi \wedge \langle A \mid A' : \alpha \rangle \top \rightarrow \langle A \mid A' : \alpha \rangle \phi$ |
| M2: $\vdash f_A \rightarrow (\langle \alpha \rangle \phi \leftrightarrow \langle A : \alpha \rangle \phi)$ | M4: $\vdash \langle A : \alpha \rangle \phi \mid \langle A' : \overline{\alpha} \rangle \psi \rightarrow \langle A, A' : \alpha \rangle (\phi \mid \psi)$ |
| **Mixed rules** | |
| MR1: If $\vdash (\bigvee_{\mathcal{M} \in \mathfrak{M}_{(\!\mid \phi \mid\!)}^{act(\phi)_+}} f_{\mathcal{M}}) \rightarrow \phi$ then $\vdash \phi$ | |

and rules, of dynamic axioms and rules and of epistemic axioms and rules. We will also have a class of mixed axioms and rules that combine different operators.

Observe that the disjunctions in axiom $S6$ and in the rule $MR1$ are finitary.

**Definition 25.** *We say that a formula* $\phi \in \mathcal{F}^{\mathfrak{A}}$ *is* provable in $\mathcal{L}_{\mathbb{A}}^{\mathfrak{A}}$ *if* $\phi$ *can be derived, as a theorem, using the axioms and the rules of* $\mathcal{L}_{\mathbb{A}}^{\mathfrak{A}}$. *We denote this by* $\vdash \phi$. *We say that a formula* $\phi \in \mathcal{F}_{\mathbb{A}}^{\mathfrak{A}}$ *is* consistent in $\mathcal{L}_{\mathbb{A}}^{\mathfrak{A}}$ *if* $\neg\phi$ *is not* $\mathcal{L}_{\mathbb{A}}^{\mathfrak{A}}$*-provable.*

*We call a formula* $\phi \in \mathcal{F}_{\mathbb{A}}^{\mathfrak{A}}$ satisfiable *if there exists a context* $\mathcal{M}$ *and a process* $P \in \mathcal{M}$ *such that* $\mathcal{M}, P \models \phi$. *We call a formula* $\phi \in \mathcal{F}_{\mathbb{A}}^{\mathfrak{A}}$ validity *if for any context* $\mathcal{M}$ *and any process* $P \in \mathcal{M}$ *we have* $\mathcal{M}, P \models \phi$. *In such a situation we write* $\models \phi$.

**Theorem 4 (Soundness).** *The system* $\mathcal{L}_{\mathbb{A}}^{\mathfrak{A}}$ *is sound w.r.t. process semantics, i.e. if* $\vdash \phi$ *then* $\models \phi$.

**Theorem 5 (Completeness).** *The axiomatic system of* $\mathcal{L}_{\mathbb{A}}^{\mathfrak{A}}$ *is complete w.r.t. process semantics, i.e. if* $\models \phi$ *then* $\vdash \phi$.

[5] Note that $\mathcal{F}_{\mathbb{P}} \subset \mathcal{F}_{\mathbb{A}}^{\mathfrak{A}}$.

[6] Note that $\mathcal{F}_{\mathfrak{A}} \subset \mathcal{F}_{\mathbb{A}}^{\mathfrak{A}}$.

[7] By construction, $\mathcal{F} \subset \mathcal{F}_{\mathbb{A}}^{\mathfrak{A}}$.

The proof of this theorem uses the characteristic formulas for processes and finite closed sets and consists in proving the equivalence equivalence between $\mathcal{M}, P \models \phi$ and $\vdash f_{\mathcal{M}} \wedge f_P \rightarrow \phi$.

## 5 Formalizing the Security Scenario

We return to the security scenario proposed in subsection 1.1 and we will use CCS to encode the process and the logic to analyze it. The entire process can be represented as the process $P \equiv k.\alpha.A \mid \overline{k}.(\overline{\alpha}.P'|\overline{\beta}.P'') \mid k.\beta.E$, where $P'$ is interpreted as *"event p happened"* while $P''$ as *"event $\neg p$ happened"*. $k.\alpha.A$ is the process of Alice, $\overline{k}(\overline{\alpha}.P'|\overline{\beta}.P'')$ is Bob's and $k.\beta.E$ is the process of Eve. We associate to Alice three epistemic agents, $A_1, A_2, A_3$ that represent the three successive states of Alice in our scenario. Similarly $E_1, E_2, E_3$ are the agents representing different instances of Eve, while $B_1, B_2, B_3, B_4$ represent instances of Bob. The model is given by $\mathcal{M} = \overline{\{P\}}$ and the interpretation $I$ in Table 5.

**Table 5.** The interpretation of epistemic agents

| *Alice* | *Bob* | *Eve* |
|---|---|---|
| $I(A_1) = k.\alpha.A$ | $I(B_1) = \overline{k}.(\overline{\alpha}.P'\|\overline{\beta}.P'').\beta.E$ | $I(E_1) = k.\beta.E$ |
| $I(A_2) = \alpha.A$ | $I(B_2) = \overline{\alpha}.P'\|\overline{\beta}.P''$ | $I(E_2) = \beta.E$ |
| $I(A_3) = A$ | $I(B_3) = P'\|\overline{\beta}.P''$ | $I(E_3) = E$ |
| | $I(B_4) = \overline{\alpha}.P'\|P''$ | |

Now we can express that Alice and Bob can recognize each other and that Alice can inform Bob about $p$ by $\mathcal{M}, P \models \langle A_1, B_1 : k\rangle\langle A_2, B_2 : \alpha\rangle(P'|\top)$.

But Bob can also communicate with Eve, as Eve has the encryption key: $\mathcal{M}, P \models \langle A_1, E_1 : k\rangle\langle A_2, E_2 : \alpha\rangle(P''|\top)$.

Alice knows that she can communicate with Bob, using the key $k$, and as a result Bob will be informed about the event $p$: $\mathcal{M}, P \models K_{A_1}\langle k\rangle\langle\alpha\rangle(P'|\top)$. But if Alice is aware of the possibility of an attack she cannot be sure that, after she sent the messages to Bob, Bob does know that $p$ happened; it might be the case that Bob did not receive Alice's message and that he communicated instead with the impersonator:
$\mathcal{M}, P \models \neg[k][\alpha]K_{A_3}(P'|\top)$ or $\mathcal{M}, P \models \langle k\rangle\langle\alpha\rangle\neg K_{A_3}K_{B_3}(P'|\top)$.

Alice knows that Bob knows that $p$ happened only if Bob did the two communications with her: $\mathcal{M}, P \models K_{A_1}\langle A_1, B_1 : k\rangle\langle A_2, B_2 : \alpha\rangle K_{B_3}(P'|\top)$.

Before communication Bob knows only that whatever Alice will say it will be true: $\mathcal{M}, P \models K_{B_1}[A_1, B_1 : k][A_2, B_2 : \alpha]\top$.

Eve knows that she can present herself as Alice (i.e. can send $k$) but she can be sure that will communicate with Bob:
$\mathcal{M}, P \models K_{E_1}\langle E_1 : k\rangle\top$ and $\mathcal{M}, P \models \neg K_{E_1}\langle E_1, B_1\rangle\top$

In the same way we can express many complex properties. Further we can use model checking or theorem proving to play with such properties.

## 6 Concluding Remarks and Future Works

In this paper we introduced a new dynamic-epistemic logic, $\mathcal{L}_{\mathbb{A}}^{\mathfrak{A}}$, with a process-algebraical semantics that combines well with process algebraical modelling of information flow, but that can also directly express agent-dependent partial information features and their dynamics. This logic is meant to be used for expressing properties of multiagent distributed systems. In this respect the society of agents $\mathfrak{A}$ came with an algebraical structure that depicts the distribution of the modules which are observed by the agents. In expressing this we used operators from spatial logics together with operators characteristic for dynamic-epistemic logics.

The logic is presented with a complete and decidable axiomatic system containing similar axioms with the logics it combines.

With respect to dynamic-epistemic logics, the novelty of our logic consists in assuming an algebraical structure on the class of agents. Thus, we can speak about the knowledge of agents $A', A''$ but also about the knowledge of the agent $A'|A''$ which subsumes the knowledge of $A'$, of $A''$, and the knowledge derived from the fact that what $A'$ and $A''$ see are modules running in parallel as parts of the same system.

With respect to logics for processes (spatial logics), our logic focuses on agents and their knowledge proposing a direct way of encoding epistemic properties that are relevant for many applications and which, using the logics of processes only can be encoded in a difficult or unnatural way. Thus we can trace the evolution of the agent's knowledge and we can express properties concerning their dynamics. Such properties are important e.g. in analyzing communication protocols where the success of a protocol depends on the knowledge of the agents involved.

## References

1. Baltag, A., Moss, L.S.: Logics for Epistemic Programs. Synthese 139(2) (2004)
2. Baltag, A., Moss, L.S., Solecki, S.: The Logic of Public Announcements. Common Knowledge and Private Suspicions, CWI Tech. Rep. SEN-R9922 (1999)
3. van Benthem, J.F.A.K.: Logic for information update. In: Proc. of TARK'01 (2001)
4. Calcagno, C., Cardelli, L., Gordon, A.D.: Deciding validity in a spatial logic for trees. Journal of Functional Programming 15 (2005)
5. Caires, L., Lozes, E.: Elimination of Quantifiers and Decidability in Spatial Logics for Concurrency. In: Gardner, P., Yoshida, N. (eds.) CONCUR 2004. LNCS, vol. 3170, Springer, Heidelberg (2004)
6. Caires, L., Cardelli, L.: A Spatial Logic for Concurrency (Part I). Information and Computation 186(2) (2003)
7. Dam, M.: Model checking mobile processes. Information and Computation 129(1) (1996)
8. van Ditmarsch, H.: Knowledge games. Bull. of Economic Research 53(4) (2001)
9. Fagin, R., et al.: Reasoning about Knowledge. MIT Press, Cambridge (1995)
10. Gabbay, M., Pitts, A.: A New Approach to Abstract Syntax Involving Binders. Formal Aspects of Computing 13(3-5), 341–363 (2002)
11. Gerbrandy, J., Groeneveld, W.: Reasoning about information change. Journal of Logic, Language and Information 6 (1997)

12. Harel, D., et al.: Dynamic Logic. MIT Press, Cambridge (2000)
13. Hennessy, M., Milner, R.: Algebraic laws for Nondeterminism and Concurrency. Journal of JACM 32(1) (1985)
14. Hintikka, J.: Knowledge and Belief, Ithaca, N.Y.: Cornell University Press (1962)
15. Mardare, R.: Logical analysis of Complex Systems. Dynamic Epistemic Spatial Logics, Ph.D. thesis, DIT, University of Trento (2006)
16. Mardare, R., Priami, C.: Decidable extensions of Hennessy-Milner Logic. In: Najm, E., Pradat-Peyre, J.F., Donzeau-Gouge, V.V. (eds.) FORTE 2006. LNCS, vol. 4229, Springer, Heidelberg (2006)
17. Mardare, R., Priami, C.: Dynamic Epistemic Spatial Logics, Technical Report, 03/, Microsoft Research Center for Computational and Systems Biology, Trento, Italy (2006) available from http://www.cosbi.eu
18. Mardare, R.: Dynamic-Epistemic reasoning on distributed systems, Technical Report 2007, Microsoft Research Center for Computational and Systems Biology, Trento, Italy (2006) available from http://www.cosbi.eu
19. Milner, R.: A Calculus of Communicating Systems. Springer-Verlag, New York, Inc. (1982)
20. Milner, R., Parrow, J., Walker, D.: Modal logics for mobile processes. TCS 114 (1993)
21. Sangiorgi, D.: Extensionality and Intensionality of the Ambient Logics. In: Proc. of the 28th ACM Annual Symposium on Principles of Programming Languages (2001)
22. Stirling, C.: Modal and temporal properties of processes. Springer-Verlag, New York, Inc. (2001)