# Decidable extensions of Hennessy-Milner Logic$^\star$

Radu Mardare[1] and Corrado Priami[1,2]

[1] University of Trento, Italy
[2] Microsoft Research - University of Trento
Center for Computational and Systems Biology, Trento, Italy

**Abstract.** We propose a new class of logics for specifying and model-checking properties of distributed systems - Dynamic Epistemic Spatial Logics. They have been designed as extensions of Hennessy-Milner logic with spatial operators (inspired by Cardelli-Gordon-Caires spatial logic) and epistemic operators (inspired by dynamic-epistemic logics). Our logics focus on observers, agents placed in different locations of the system having access to some subsystems. Treating them as epistemic agents, we develop completely axiomatized and decidable logics that express the information flow between them in a dynamic and distributed environment. The knowledge of an epistemic agent, is understood as the information, locally available to our observer, about the overall-global system.

## 1 Introduction

The development of computer networks came with new paradigms of computation. The concept of *monolithic computational systems* (one-agent system) was replaced by the *concurrent distributed computing systems* (multi-agent systems), representing programs or processors running in parallel and organized in networks of subsystems. They interact, collaborate, communicate and interrupt each other. Underlying this new paradigm is the assumption that each such part has its own identity, as a subsystem. We shall associate to a subsystem an *agent*.

The agents are needed for discriminating between the events of the systems behavior. If we wish to identify a particular event, we have little choice but to identify the agents involved. Hence the agents might be understood as (associated with) separate and independently observable units of behavior and computation. They evolve in a given environment, following some primitive rules, their evolution influencing the structure of the whole (multi-agent) system. The main feature of the agents is their ability to communicate, that is to exchange information inside their environment.

Such a multi-agent system reflects interactive, concurrent and distributed behaviors and computations of agents, thus is extremely complex. The success in dealing with this complexity depends on the mathematical model we choose to abstract the system. Further we focus on two major paradigms.

$^\star$ Work partially supported by EU-IST project 016004 SENSORIA

## 1.1 The agent is nothing more but its behavior

Process Algebra [1] abstracts the agents of the system on the level of their behavior and using some algebraic calculi and operational semantics describes the evolution of the whole system. Further, as the behavior of a concurrent system is a succession of affine states in (possibly branching) time, it was considered the possibility of applying modal (especially temporal) logics for specifying properties of the processes that modelled distributed systems.

In studying security problems, for example, we may want to be able to specify systems composed by agents that deal with fresh or secret resources. We may want to express properties such as "*the agent has the key*", "*eventually the agent crosses the firewall*" or "*there is always at most one agent here able to decrypt the message*".

Hennessy-Milner logic [2] is one of the first modal logics that proposes some dynamic operators, indexed by CCS actions, $\langle\alpha\rangle\phi$ to capture the weakest precondition of a program w.r.t. a given post-specification $\phi$. The idea was further developed in combination with temporal operators [3] and applied to other process calculi [4–6]. All these logics are characterized by their *extensional nature* - they distinguish processes up to their behavior.

The specific applications of mobile computing call for an increased degree of expressiveness for specifying and reasoning about locations, resources, independence, distribution, connectivity or freshness. Thus, *Spatial logics* [7, 8] propose, in addition to the modal-temporal operators, some modal-spatial operators such as the *parallel operator* $\phi|\psi$ (meaning that the current system can be split into a parallel composition of two subsystems, one satisfying $\phi$ and the other satisfying $\psi$), and its adjoint - the *guarantee operator* $\phi\triangleright\psi$, or *ambient-location operators*[3] such as $n[\phi]$ (meaning that the current system can be described as a box $n[P]$ containing a subsystem $P$ that satisfies $\phi$), etc. A formula in a spatial logic describes a property of a particular part of the system at a particular time. These spatial modalities have an *intensional flavor*, the properties they express being invariant only for simple spatial rearrangements of the system.

Still most of the spatial logics face with decidability problems: it was proved that the basic spatial operators, in combination with temporal operators, generate undecidable logics [11–13] even against a finite piece of CCS[14].

### An agent is defined by its "knowledge"

The other paradigm of modelling multi-agent systems is inspired by epistemic logics: reasoning about systems in terms of *knowledge of the agents* [15]. The knowledge of an agent is understood as the sum of actions the agent (subsystem) may take as a function of its local state in a given environment. Thus the agent "knows" its *protocol* in a given system, its knowledge consists in the information related to evolution of this subsystem in an unknown environment.

[3] These operators are characteristic for Ambient Logic [8], a special spatial logic developed for Ambient Calculus [9, 10].

*Epistemic logics* [15] formalize, in a direct manner, notions of knowledge, possessed by an agent, or a group of agents, using modalities like $K_A\phi$ (*$A$ knows $\phi$*), or $Ck\phi$ (*all the agents knows $\phi$, i.e. $\phi$ is a common knowledge*). These logics supports Kripke-model based semantics, each basic modality being associated with a binary *accessibility relation* in these models. Thus for each epistemic agent $A$ we devise an accessibility relation $\xrightarrow{A}$ , called *indistinguishability relation for* $A$, expressing the agent's uncertainty about the current state. The states $s'$ such that $s \xrightarrow{A} s'$ are the *epistemic alternatives* of $s$ to agent $A$: if the current state of the whole system is $s$, $A$ thinks that any of the alternatives $s'$ may be the current state (as it does not have enough information to distinguish them). These logics have been extensively studied and applied to model complex communication-based multi-agent systems.

By mixing dynamic [16] and epistemic [15] formalisms have been developed Dynamic Epistemic Logics [17–19]. These logics combine a rich expressivity with low complexity ensuring decidability and complete axiomatizations.

### Our approach

The two paradigms of modelling concurrent distributed systems presented before were developed in parallel, but to our knowledge, there has been no unified paradigm. We propose such a paradigm in this paper, used for constructing a new logic for concurrency completely axiomatized and decidable that combines the features of spatial logics with the epistemic logics thus obtaining a special type of dynamic epistemic logic equipped with spatial operators. We call it Dynamic Epistemic Spatial Logic. While the dynamic and spatial features allow to express complex spatial/temporal properties, the epistemic operators denote the knowledge state of the agents. Thus we can express, for a security protocol, that Alice knows the key $k$, but she also knows that Bob knows that she knows this key. The hierarchic epistemic statements are relevant for expressing and validating complex security protocols [20, 17].

Formally, we extend Hennessy-Milner logic with the parallel operator and epistemic operators. In our logics the epistemic agents are named by the processes they are related with. Thus $K_P\phi$ means *the agent related with $P$ knows $\phi$* and it holds iff $\phi$ is satisfied by any process having $P$ as subprocess. The intuition is that the agent related with $P$ can see only $P$. So, it cannot differentiate between the global states $P$, $P|Q$ or $P|R$ of the whole system, as in all these states it sees only $P$. Thus its knowledge rests on the properties $\phi$ that are satisfied by each of these states (processes).

We prove that Dynamic Epistemic Spatial Logic is decidable and we develop sound-complete Hilbert-style axiomatic systems, against process semantics based on a fragment of CCS [14], for two differently expressive such logics.

Concluding, the novelty of our logic with respect to the classical spatial logics is the use of the epistemic operators for expressing global properties while ensuring decidability. The epistemic operators allow to refer directly to agents of our system by mean of their knowledge. By combining the partial knowledge of the agents we can specify complex properties of distributed multi-agent systems.

**Outline of the paper**

The paper is organized as follows. In section 2 we introduce and study a small finite fragment of CCS on which we will focus for the rest of the paper[4]. Some new concepts will be introduced and used further, such as structural bisimulation and pruning processes and sets of processes. Starting with section 3 we define our logics. Two such systems will be introduced $\mathcal{L}_{DS}$ and its extension $\mathcal{L}_{DES}$. For both we will prove the bounded finite model property and develop sound complete Hilbert-style axiomatic systems against the chosen semantics. Eventually we end the paper with some concluding remarks.

For the proofs of the theorems presented in this paper, and for additional results the reader is referred to [21] for Dynamic Epistemic Spatial Logic and to [22] for Dynamic Spatial Logic. Some extensions of these logics have been presented in [23]

## 2 Processes and contexts

In this section, focusing on the fragment of CCS introduced in definition 1, we develop some concepts on which we will base the further constructs.

**Definition 1 (Processes).** *Consider the fragment of CCS generated by the next syntax, where $\mathbb{A}$ is a denumerable set of actions and $\alpha \in \mathbb{A}$:*

$$P ::= 0 \mid \alpha.P \mid P|P$$

*Hereafter this calculus[5] is the object of our paper. We will use $\alpha, \beta$ to range over $\mathbb{A}$ and we will denote by $\mathfrak{P}$ the class of processes. As standard, we consider defined over $\mathfrak{P}$ a structural congruence, Table 1, and a labelled transition system, Table 2.*

$$P|0 \equiv P \qquad P|Q \equiv Q|P \qquad P|(Q|R) \equiv (P|Q)|R$$

**Table 1.** The axioms the structural congruence

**Assumption [Representativeness modulo structural congruence]:** As the structural congruence is the ultimate level of expressivity we want for our logic, hereafter we will speak about processes up to structural congruence.

[4] This calculus provides a semantics against which the classical spatial logic is undecidable [11].

[5] We can, additionally, consider an involution on $\mathbb{A}$ that associate to each action $\alpha \in \mathbb{A}$ an action $\overline{\alpha} \in \mathbb{A}$, as usual in CCS, and also consider the silent action $\tau$. But all these represent just syntactic sugar, irrelevant from the point of view of the logic we discuss.

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \qquad \frac{P \equiv Q \qquad P \xrightarrow{\alpha} P'}{Q \xrightarrow{\alpha} P'} \qquad \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}$$

**Table 2.** The transition system

**Definition 2.** *We call a process $P$* guarded *iff $P \equiv \alpha.Q$ for $\alpha \in \mathbb{A}$. We introduce the notation $P^k \overset{def}{=} \underbrace{P|...|P}_{k}$, and convey to denote $P^0 \equiv 0$.*

We extend the operators from processes to sets of processes.

**Definition 3.** *For any sets of processes $M, N \subset \mathfrak{P}$ and any $\alpha \in \mathbb{A}$ we define:*

$$\alpha.M \overset{def}{=} \{\alpha.P \mid P \in M\} \qquad\qquad M|N \overset{def}{=} \{P|Q \mid P \in M, Q \in N\}$$

*As we speak about processes up to structural congruence, the parallel operator on sets of processes will be commutative, associative and will have $\{0\}$ as null.*

Now we define the *contexts*. The intuition is that a *context* $\mathcal{M}$ is a (possibly infinite) set of processes that contains, in a maximal manner, any process representing a possible state of our system or of a subsystem of our system. Hence if a process belongs to a context then any process obtained by pruning its syntactic tree should belong to the context, as it might represent a possible state of a. For the same reason, the context should be also closed to transitions. $\pi(P)$ denotes the set of all processes obtained by pruning the syntactic tree of $P$.

**Definition 4 (Pruning the syntactic tree).** *For $P \in \mathfrak{P}$ define*[6] *$\pi(P) \subset \mathfrak{P}$:*

1. $\pi(0) \overset{def}{=} \{0\}$ 2. $\pi(\alpha.P) \overset{def}{=} \{0\} \cup \alpha.\pi(P)$ 3. $\pi(P|Q) \overset{def}{=} \pi(P)|\pi(Q)$

*We extend the definition of $\pi$ to sets of processes $M \subset \mathfrak{P}$ by $\pi(M) \overset{def}{=} \bigcup_{P \in M} \pi(P)$.*

**Definition 5 (Context).** A context *is a nonempty set $\mathcal{M} \subseteq \mathfrak{P}$ such that:*

*1. if $P \in \mathcal{M}$ and $P \longrightarrow P'$ then $P' \in \mathcal{M}$* *2. if $P \in \mathcal{M}$ then $\pi(P) \subset \mathcal{M}$*

### 2.1 Size of a process

Further we define the *size of a process*, following a similar idea developed in [24] for sizes of trees. The intuition is that the process has a *height* given by the vertical size of its syntactic tree, and a *width* equal to the maximum number of bisimilar subprocesses that can be found in a node of the syntactic tree.

**Definition 6 (Size of a process).** *We define, inductively, the* size *$(h, w)$ ($h$ stays for* height *and $w$ for* width*) of a process $P$, denoted by $[\![P]\!]$:*

1. $[\![0]\!] \overset{def}{=} (0, 0)$ 2. $[\![P]\!] \overset{def}{=} (h, w)$ *iff*
   - $P = (\alpha_1.Q_1)^{k_1}|(\alpha_2.Q_2)^{k_2}|...|(\alpha_j.Q_j)^{k_j}$, $[\![Q_i]\!] = (h_i, w_i)$, $i \in 1..j$
   - $h = 1 + max(h_1, ..., h_k)$, $w = max(k_1, ..., k_j, w_1, ..., w_j)$

*We convey to 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 $w_1 < w_2$.*

[6] We consider also $\pi(P)$ defined up to structural congruence.

**Definition 7 (Size of a set of processes).** *Let $M \subset \mathfrak{P}$. We write $[\![M]\!] = (h,w)$ iff $(h,w) = max\{[\![P]\!] \mid P \in M\}$*[7].

*Example 1.* We show the size for some processes:

1. $[\![0]\!] = (0,0)$
2. $[\![\alpha.0]\!] = (1,1)$
3. $[\![\alpha.0|\beta.0]\!] = (1,1)$
4. $[\![\alpha.0|\alpha.0]\!] = (1,2)$
5. $[\![\alpha.\alpha.0]\!] = [\![\alpha.\beta.0]\!] = (2,1)$
6. $[\![\alpha.(\beta.0|\beta.0)]\!] = (2,2)$

### 2.2 Substitutions

For the future constructs is also useful to introduce the substitutions of actions in a process.

**Definition 8 (The set of actions of a process).** *We define $Act(P) \subset \mathbb{A}$ by:*
1.$Act(0) \overset{def}{=} \emptyset$ 2.$Act(\alpha.P) \overset{def}{=} \{\alpha\} \cup Act(P)$ 3.$Act(P|Q) \overset{def}{=} Act(P) \cup Act(Q)$
*For a set $M \subset \mathfrak{P}$ of processes we define $Act(M) \overset{def}{=} \bigcup_{P \in M} Act(P)$.*

**Definition 9 (Action substitution).** *We call* action substitution *any function $\sigma : \mathbb{A} \longrightarrow \mathbb{A}$. We syntactically extend it, from actions to processes, by:*
1. $\sigma(0) \overset{def}{=} 0$ 2. $\sigma(P|Q) \overset{def}{=} \sigma(P)|\sigma(Q)$ 3. $\sigma(\alpha.P) \overset{def}{=} \sigma(\alpha).\sigma(P)$
*For $M \subset \mathfrak{P}$ let $\sigma(M) \overset{def}{=} \{\sigma(P) \mid P \in M\}$. We also use notation $M^\sigma$, $P^\sigma$ for $\sigma(M)$ and $\sigma(P)$. The* set of actions of $\sigma$, *$act(\sigma)$, is defined as*

$$act(\sigma) \overset{def}{=} \{\alpha, \beta \in \mathbb{A} \mid \alpha \neq \beta,\ \sigma(\alpha) = \beta\}$$

### 2.3 Structural bisimulation

The *structural bisimulation* is a congruence on processes (then extended to contexts) defined as an approximation of the structural congruence bound by two sizes: the *height* (the depth of the syntactic tree) and the *weight* (the maximum number of bisimilar subprocesses that can be found in a node of the syntactic tree) of a process. A conceptually similar congruence was proposed in [24] for analyzing trees of location for the static ambient calculus.

The structural bisimulation analyzes the behavior of a process focusing on a boundary $(h,w)$ of its syntactic tree. The intuition is that $P \approx_h^w Q$ ($P$ and $Q$ are structurally bisimilar on size $(h,w)$) iff when we consider for both processes their syntactic trees up to the depth $h$ only (we prune them on the height $h$) and we ignore the presence of more than $w$ parallel bisimilar subprocesses in any node of the syntactic trees (we prune the trees on weight $w$), we obtain identical syntactic trees.

[7] Observe that not all the sets of processes have a size, as for an infinite one it might be not possible to have the maximum required.

**Definition 10 (Structural bisimulation).** *For $P, Q \in \mathfrak{P}$ we define $P \approx_h^w Q$ by:*

*$P \approx_0^w Q$ always*

*$P \approx_{h+1}^w Q$ iff for any $i \in 1..w$ and any $\alpha \in \mathbb{A}$ we have*

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

**Theorem 1 (Congruence).** *$\approx_h^w$ is a congruence on processes.*

We extend the definitions of structural bisimulation from processes to contexts.

**Definition 11 (Structural bisimulation over contexts).** *Let $\mathcal{M}, \mathcal{N}$ be two contexts. We write $\mathcal{M} \approx_h^w \mathcal{N}$ iff*

*1. for any $P \in \mathcal{M}$ there is a $Q \in \mathcal{N}$ with $P \approx_h^w Q$*

*2. for any $Q \in \mathcal{N}$ there is a $P \in \mathcal{M}$ with $P \approx_h^w Q$*

*We convey to write $(\mathcal{M}, P) \approx_h^w (\mathcal{N}, Q)$ for the case when $P \in \mathcal{M}$, $Q \in \mathcal{N}$, $P \approx_h^w Q$ and $\mathcal{M} \approx_h^w \mathcal{N}$.*

*Example 2.* Consider the processes $R \equiv \alpha.(\beta.0|\beta.0|\beta.0)|\alpha.\beta.0$ and $S \equiv \alpha.(\beta.0|\beta.0)|\alpha.\beta.\alpha.0$. We can verify the requirements of the definition 10 and decide that $R \approx_2^2 S$. But $R \not\approx_3^2 S$ because on the depth 2 $R$ has an action $\alpha$ (in figure 1 marked with a dashed arrow) while $S$ does not have it (because the height of $S$ is only 2). Also $R \not\approx_2^3 S$ because $R$ contains only 2 (bisimilar) copies of $\beta.0$ while $S$ contains 3 (the extra one is marked with a dashed arrow). Hence,

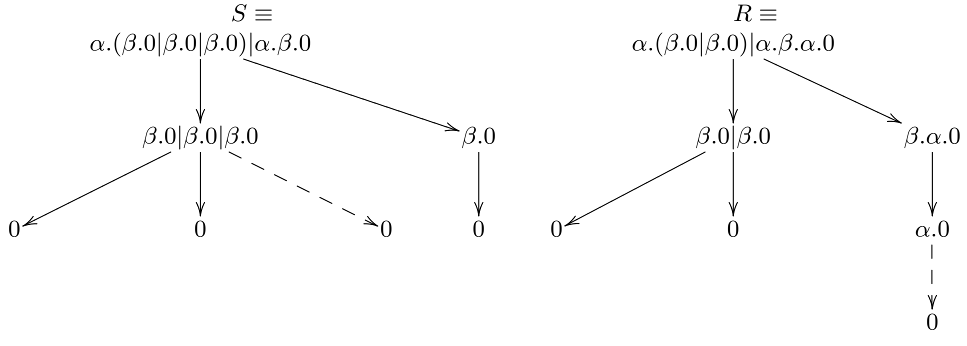
$S \equiv$
$\alpha.(\beta.0|\beta.0|\beta.0)|\alpha.\beta.0$
$\beta.0|\beta.0|\beta.0$
$\beta.0$
0
0
0
0
$R \equiv$
$\alpha.(\beta.0|\beta.0)|\alpha.\beta.\alpha.0$
$\beta.0|\beta.0$
$\beta.\alpha.0$
0
0
$\alpha.0$
0

**Fig. 1.** Syntactic trees

for any weight bigger than 2 this feature will show the two processes as different. But if we remain on depth 1 we have $R \approx_1^3 S$, as on this deep the two processes have the same number of bisimilar subprocesses, i.e. any of them can perform $\alpha$ in two ways giving, further, processes in the relation $\approx_0^3$. Indeed $R \equiv \alpha R'|\alpha R''$, where $R' \equiv \beta.0|\beta.0|\beta.0$ and $R'' \equiv \beta.0$ and $S \equiv \alpha.S'|\alpha.S''$, where $S' \equiv \beta.0|\beta.0$ and $S'' \equiv \beta.\alpha.0$. By definition, $R' \approx_0^3 S'$ and $R'' \approx_0^3 S''$.

### 2.4 Pruning processes and contexts

We introduce an effective method to construct, given a process $P$, a minimal process $Q$ that has an established size $(h, w)$ and is structurally bisimilar to $P$ on this size. Because the construction is based on pruning the syntactic tree of $P$ on a given size, we call this method *bound pruning*, and we refer to $Q$ as *the pruned of $P$ on the size* $(h, w)$.

**Theorem 2 (Bound pruning theorem).** *For any process $P \in \mathfrak{P}$ and any $(h, w)$ exists a process $Q \in \mathfrak{P}$ with $P \approx_h^w Q$ and $\llbracket Q \rrbracket \leq (h, w)$.*

*Proof.* We describe the construction[8] of $Q$ by induction on $h$.

**For** $h = 0$**:** we just take $Q \equiv 0$, because $P \approx_0^w Q$ and $\llbracket 0 \rrbracket = (0, 0)$.

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

Let $P_i'$ be the result of pruning $P_i$ by $(h, w)$ (we use the inductive step of construction) and $P' \equiv \alpha_1.P_1'|...|\alpha_n.P_n'$. As for any $i = 1..n$ we have $P_i \approx_h^w P_i'$ (by the inductive hypothesis), we obtain, using theorem 1, that $\alpha_i.P_i \approx_{h+1}^w \alpha_i.P_i'$ and further $P \approx_{h+1}^w P'$.

Consider the canonical representation of $P' \equiv (\beta_1.Q_1)^{k_1}|...|(\beta_m.Q_m)^{k_m}$.

Let $l_i = min(k_i, w)$ for $i = 1..m$. Then we define $Q \equiv (\beta_1.Q_1)^{l_1}|...|(\beta_m.Q_m)^{l_m}$. Obviously $Q \approx_{h+1}^w P'$ and as $P \approx_{h+1}^w P'$, we obtain $P \approx_{h+1}^w Q$. By construction, $\llbracket Q \rrbracket \leq (h + 1, w)$.

**Definition 12 (Bound pruning processes).** *For a process $P$ and for 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 2.*

*Example 3.* Consider the process $P \equiv \alpha.(\ \beta.(\gamma.0|\gamma.0|\gamma.0)\ |\ \beta.\gamma.0\ )\ |\ \alpha.\beta.\gamma.0$
Observe that $\llbracket P \rrbracket = (3, 3)$, hence $P_{(3,3)} \equiv P$. For constructing $P_{(3,2)}$ we have to prune the syntactic tree of $P$ such that to not exist, in any node, more than two bisimilar branches. Hence $P_{(3,2)} = \alpha.(\ \beta.(\gamma.0|\gamma.0)\ |\ \beta.\gamma.0)\ |\ \alpha.\beta.\gamma.0$
If we want to prune $P$ on the size $(3, 1)$, we have to prune its syntactic tree such that, in any node, there are no bisimilar branches. The result is $P_{(3,1)} = \alpha.\beta.\gamma.0$.
For pruning $P$ on the size $(2, 2)$, we have to prune all the nodes on depth 2 and in the new tree we have to let, in any node, a maximum of two bisimilar branches. As a result of these modifications, we obtain $P_{(2,2)} = \alpha.(\beta.0|\beta.0)\ |\ \alpha.\beta.0$. Going further we obtain the smaller processes $P_{(0,0)} = 0$, $P_{(1,1)} = \alpha.0$, $P_{(1,2)} = \alpha.0|\alpha.0$, $P_{(2,1)} = \alpha.\beta.0$.

Further we define the bound pruning of a context $\mathcal{M}$ as the context generated by the set of pruned processes of $\mathcal{M}$.

[8] This construction is not necessarily unique.

**Definition 13 (Bound pruning contexts).** *We say that the set $M \subset \mathfrak{P}$ is a system of generators for the context $\mathcal{M}$ if $\mathcal{M}$ is the smallest context that contains $M$. We denote this by $\overline{M} = \mathcal{M}$. For any context $\mathcal{M}$ and any $(h, w)$ we define $\mathcal{M}_{(h,w)} \overset{def}{=} \overline{\{P_{(h,w)} \mid P \in \mathcal{M}\}}$.*

**Theorem 3.** *For any context $\mathcal{M}$, any $P \in \mathcal{M}$, and any size $(h, w)$ we have $(\mathcal{M}, P) \approx^h_w (\mathcal{M}_{(h,w)}, P_{(h,w)})$.*

**Definition 14.** *Let $A \subset \mathbb{A}$. Consider the sets:*

$$\mathfrak{P}^A_{(h,w)} \overset{def}{=} \{P \in \mathfrak{P} \mid Act(P) \subseteq A,\ [\![P]\!] \leq (h, w)\}$$

$$\mathfrak{M}^A_{(h,w)} \overset{def}{=} \{\overline{M} \subset \mathfrak{P} \mid Act(M) \subseteq A,\ [\![M]\!] \leq (h, w)\}$$

**Theorem 4.** *If $A \subset \mathbb{A}$ is a finite set of actions, then the following hold:*

*1. $\mathfrak{P}^A_{(h,w)}$ is finite  2. any $\mathcal{M} \in \mathfrak{M}^A_{(h,w)}$ is a finite context  3. $\mathfrak{M}^A_{(h,w)}$ is finite.*

**Theorem 5 (Bound pruning theorem).** *Let $\mathcal{M}$ be a context. Then for any $(h, w)$ there is a context $\mathcal{N} \in \mathfrak{M}^{Act(\mathcal{M})}_{(h,w)}$ such that $\mathcal{M} \approx^w_h \mathcal{N}$. Moreover, $\mathcal{N} = \mathcal{M}_{(h,w)}$ has this property.*

## 3 Logics for specifying distributed systems

In this section we introduce Dynamic Spatial Logic, $\mathcal{L}_{DS}$, as an extension of Hennessy-Milner logic with the parallel operator and Dynamic Epistemic Spatial Logic, $\mathcal{L}_{DES}$, which extends $\mathcal{L}_{DS}$ with the epistemic operators. The intuition is to define the knowledge of the process $P$ in the context $\mathcal{M}$ as the common properties of the processes in $\mathcal{M}$ that contain $P$ as subprocess. Hence the knowledge implies a kind of universal quantifier over $\mathcal{M}$.

The satisfiability relations will evaluate a formula to a process in a context.

For our logics, we propose Hilbert-style axiomatic systems proved to be sound and complete with respect to process semantics. $\mathcal{L}_{DS}$ and $\mathcal{L}_{DES}$ satisfy the bond finite model property against the process semantics that entails the decidability for satisfiability, validity and model checking for both logics.

### 3.1 Syntax

**Definition 15 (Languages).** *We define the language of Dynamic Spatial Logic, $\mathcal{F}_{DS}$, and the language of Dynamic Epistemic Spatial Logic, $\mathcal{F}_{DES}$, for $\alpha \in \mathbb{A}$:*

$$\phi := \ 0 \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi|\phi \mid \langle\alpha\rangle\phi \qquad (\mathcal{F}_{DS})$$

$$\phi := \ 0 \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi|\phi \mid \langle\alpha\rangle\phi \mid K_Q\phi \qquad (\mathcal{F}_{DES})$$

**Definition 16 (Derived operators).** *In addition we have derived operators:*

1. $\bot \overset{def}{=} \neg\top$
2. $\phi \vee \psi \overset{def}{=} \neg((\neg\phi) \wedge (\neg\psi))$
3. $\phi \rightarrow \psi \overset{def}{=} (\neg\phi) \vee \psi$
4. $[\alpha]\phi \overset{def}{=} \neg(\langle\alpha\rangle(\neg\phi))$
5. $1 \overset{def}{=} \neg((\neg 0) \mid (\neg 0))$
6. $\langle !\alpha\rangle\psi \overset{def}{=} (\langle\alpha\rangle\psi) \wedge 1$
7. $\widetilde{K}_Q\phi \overset{def}{=} \neg K_Q\neg\phi$

We could also introduce, for each action $\alpha$, a derived operator[9] $\langle \alpha, \overline{\alpha} \rangle$ to express communication by $\alpha$, supposing that we have defined an involution $co : \mathbb{A} \longrightarrow \mathbb{A}$ which associates to each action $\alpha$ its co-action $\overline{\alpha}$:

$$\langle \alpha, \overline{\alpha} \rangle \phi \overset{def}{=} \bigvee_{\phi \leftrightarrow \phi_1 | \phi_2} \langle \alpha \rangle \phi_1 | \langle \overline{\alpha} \rangle \phi_2$$

### 3.2 Process semantics

A formula of $\mathcal{F}_{DS}$, or of $\mathcal{F}_{DES}$, will be evaluated to processes in a given context, by mean of a satisfaction relation $\mathcal{M}, P \models \phi$.

**Definition 17 (Models and satisfaction).** *A model of $\mathcal{L}_{DS}$ or of $\mathcal{L}_{DES}$ is a context $\mathcal{M}$ for which we define the satisfaction relation, for $P \in \mathcal{M}$, as follows:*
$\mathcal{M}, P \models \top$ *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* $\mathcal{M}, Q \models \phi$, $\mathcal{M}, R \models \psi$
$\mathcal{M}, P \models \langle \alpha \rangle \phi$ *iff there exists a transition* $P \xrightarrow{\alpha} P'$ *and* $\mathcal{M}, P' \models \phi$
$\mathcal{M}, P \models K_Q \phi$ *iff* $P \equiv Q|R$ *and* $\forall Q|R' \in \mathcal{M}$ *we have* $\mathcal{M}, Q|R' \models \phi$

Then the semantics of the derived operators will be:
$\mathcal{M}, P \models [\alpha]\phi$ iff for any $P' \in \mathcal{M}$ such that $P \xrightarrow{\alpha} P'$ (if any), $\mathcal{M}, P' \models \phi$
$\mathcal{M}, P \models 1$ iff $P \equiv 0$ or $P \equiv \alpha.Q$ ($P$ is null or guarded)
$\mathcal{M}, P \models \langle !\alpha \rangle \phi$ iff $P \equiv \alpha.Q$ and $\mathcal{M}, Q \models \phi$
$\mathcal{M}, P \models \widetilde{K}_Q \phi$ iff either $P \not\equiv Q|R$, or it exists $Q|S \in \mathcal{M}$ such that $\mathcal{M}, Q|S \models \phi$

Remark the interesting semantics of the operators $K_0$ and $\widetilde{K}_0$ that allow to encode, in syntax, the validity and the satisfiability w.r.t. a context:
$\mathcal{M}, P \models K_0 \phi$ iff for any $Q \in \mathcal{M}$ we have $\mathcal{M}, Q \models \phi$
$\mathcal{M}, P \models \widetilde{K}_0 \phi$ iff it exists a process $Q \in \mathcal{M}$ such that $\mathcal{M}, Q \models \phi$

### 3.3 Characteristic formulas

In this subsection we use the peculiarities of the dynamic and epistemic operators to define characteristic formulas for processes and for finite contexts. Such formulas will be useful in providing appropriate axiomatic systems for our logics and, eventually, for proving the completeness.

**Definition 18 (Characteristic formulas for processes).** *In $\mathcal{F}_{DS}$ we define a class of formulas $(c_P)_{P \in \mathfrak{P}}$, indexed by ($\equiv$-equivalence classes of) processes, by:*

1. $c_0 \overset{def}{=} 0$ 2. $c_{P|Q} \overset{def}{=} c_P | c_Q$ 3. $c_{\alpha.P} \overset{def}{=} \langle !\alpha \rangle c_P$

[9] The disjunction is considered up to logically-equivalent decompositions $\phi \leftrightarrow \phi_1 | \phi_2$ that ensures the use of a finitary formula.

**Theorem 6.** $\mathcal{M}, P \models c_Q$ *iff* $P \equiv Q$.

As $\mathcal{F}_{DES}$ is an extension of $\mathcal{F}_{DS}$, $(c_P)_{P\in\mathfrak{P}}$ characterize processes also in $\mathcal{F}_{DES}$. Specific for $\mathcal{F}_{DES}$ only is the possibility to exploit the semantics of the operators $K_0$ and $\widetilde{K}_0$, as they can describe validity and satisfiability w.r.t a model, in defining characteristic formulas for finite contexts.

**Definition 19 (Characteristic formulas for contexts).** *In $\mathcal{F}_{DES}$, if $\mathcal{M}$ is a finite context, we can define its* characteristic formula *by:*

$$c_{\mathcal{M}} = K_0(\bigvee_{Q\in\mathcal{M}} c_Q) \wedge (\bigwedge_{Q\in\mathcal{M}} \widetilde{K}_0 c_Q)$$

Suppose that $\mathcal{N}, P \models c_{\mathcal{M}}$. Then the first conjunct $K_0(\bigvee_{Q\in\mathcal{M}} c_Q)$ tells us that $\bigvee_{Q\in\mathcal{M}} c_Q$ is a validity in $\mathcal{N}$, hence each element of $\mathcal{N}$ is an element of $\mathcal{M}$, $\mathcal{N} \subseteq \mathcal{M}$. The second conjunct tells us that for each $Q \in \mathcal{M}$, $\mathcal{N}, P \models \widetilde{K}_0 c_Q$. By the semantics of $\widetilde{K}_0$ this means that it exists a process $P' \in \mathcal{N}$ such that $\mathcal{N}, P' \models c_Q$, i.e. $P' \equiv Q$. As the processes are identified up to structural congruence, $\mathcal{M} \subseteq \mathcal{N}$. Hence $\mathcal{M} = \mathcal{N}$.

**Theorem 7.** *If $\mathcal{M}$ is a finite context and $P \in \mathcal{M}$ then $\mathcal{M}, P \models c_{\mathcal{N}}$ iff $\mathcal{N} = \mathcal{M}$.*

### 3.4 Bound finite model property and decidability

Now we prove the finite model property for Dynamic Epistemic Spatial Logic that will entail the decidability against the process semantics. As a consequence, we obtain decidability for Dynamic Spatial Logic (being less expressive). Anticipating, we define a size for formulas $\phi$; then we prove that if $\mathcal{M}, P \models \phi$ then substituting, by $\sigma$, all the actions in $\mathcal{M}$ (and implicitly in $P$) that are not in the syntax of $\phi$ (as indexes of dynamic or epistemic operators) by a fixed action with the same property, and then pruning $\mathcal{M}^\sigma$ and $P^\sigma$ to the size of $\phi$ we will obtain a couple $(\mathcal{N}, Q)$ such that $\mathcal{N}, Q \models \phi$. The fixed action of substitution can be chosen as the successor[10] of the maximum action of $\phi$, which is unique. Hence $\mathcal{N} \in \mathfrak{M}^A_{(h,w)}$ where $(h, w)$ is the size of $\phi$ and $A$ is the set of actions of $\phi$ augmented with the successor of its maximum, thus $A$ is finite. But then theorem 4 ensures that the set of pairs $(\mathcal{N}, Q)$, with this property, is finite.

**Definition 20 (Size of a formula).** *We define* the sizes of a formula, $(\!|\phi|\!)$ *(height and width), inductively on $\mathcal{F}_{DES}$, by:*
1.$(\!|0|\!) = (\!|\top|\!) \stackrel{def}{=} (0, 0)$ 2.$(\!|\neg\phi|\!) \stackrel{def}{=} (\!|\phi|\!)$
*and supposing that $(\!|\phi|\!) = (h, w)$, $(\!|\psi|\!) = (h', w')$ and $[\![R]\!] = (h_R, w_R)$, further:*
3.$(\!|\phi|\psi|\!) \stackrel{def}{=} (max(h, h'), w + w')$ 4.$(\!|\phi \wedge \psi|\!) \stackrel{def}{=} (max(h, h'), max(w, w'))$
5.$(\!|\langle\alpha\rangle\phi|\!) \stackrel{def}{=} (1{+}h, 1{+}w)$ 6.$(\!|K_R\phi|\!) \stackrel{def}{=} (1{+}max(h, h_R), 1{+}max(w, w_R))$

[10] We consider defined, on the class of actions $\mathbb{A}$, a lexicographical order.

The next theorem states that $\phi$ is *"sensitive"* via satisfaction only up to size $\llparenthesis\phi\rrparenthesis$. In other words, the relation $\mathcal{M}, P \models \phi$ is conserved by substituting the couple $(M, P)$ with any other couple $(N, P)$ structurally bisimilar to it at the size $\llparenthesis\phi\rrparenthesis$.

**Theorem 8.** *If* $\llparenthesis\phi\rrparenthesis = (h, w)$*,* $\mathcal{M}, P \models \phi$ *and* $(\mathcal{M}, P) \approx_h^w (\mathcal{N}, Q)$ *then* $\mathcal{N}, Q \models \phi$*.*

Using this theorem, we conclude that if a process, in a context, satisfies $\phi$ then by pruning the process and the context on the size $\llparenthesis\phi\rrparenthesis$, we still have satisfiability for $\phi$. Indeed the theorems 2 and 3 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 21 (The set of actions of a formula).** *We define the set of actions of a formula* $\phi$*,* $act(\phi) \subset \mathbb{A}$*, inductively by:*

1.$act(0) \overset{def}{=} \emptyset$

2.$act(\top) \overset{def}{=} \emptyset$

3.$act(\phi \wedge \psi) = act(\phi|\psi) \overset{def}{=} act(\phi) \cup act(\psi)$

4.$act(\neg\phi) = act(\phi)$

5.$act(K_R\phi) \overset{def}{=} Act(R) \cup act(\phi)$

6.$act(\langle\alpha\rangle\phi) \overset{def}{=} \{\alpha\} \cup act(\phi)$

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.

**Theorem 9.** *If* $\mathcal{M}, P \models \phi$ *and* $act(\sigma) \bigcap act(\phi) = \emptyset$ *then* $\mathcal{M}^\sigma, P^\sigma \models \phi$*.*

Suppose that on $\mathbb{A}$ we have a lexicographical order $\ll$. So, for a finite set $A \subset \mathbb{A}$ we can identify a maximal element that is unique. Hence the successor of this element is unique as well. We convey to denote by $A_+$ the set obtained by adding to $A$ the successor of its maximal element. Moreover, for a context $\mathcal{N} \ni P$, for a size $(h, w)$ and for a finite set of actions $A \subset \mathbb{A}$ we denote by $\mathcal{N}_{(h,w)}^A$ (and by $P_{(h,w)}^A$ respectively) the context (respectively the process) obtained by substituting all the actions $\alpha \in Act(\mathcal{N}) \setminus A$ ($\alpha \in Act(P) \setminus A$ respectively) by the successor of the maximum element of $A$ and then pruning the context (the process) obtained to size $(h, w)$.

**Theorem 10 (Bound finite model property).**

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

*Moreover* $\mathcal{N} = \mathcal{M}_{\llparenthesis\phi\rrparenthesis}^{act(\phi)}$ *and* $Q = P_{\llparenthesis\phi\rrparenthesis}^{act(\phi)}$ *fulfill the requirements of the theorem.*

Because $act(\phi)$ is finite implying $act(\phi)_+$ finite, we apply theorem 4 ensuring that $\mathfrak{M}_{\llparenthesis\phi\rrparenthesis}^{act(\phi)_+}$ is finite and any context $\mathcal{M} \in \mathfrak{M}_{\llparenthesis\phi\rrparenthesis}^{act(\phi)_+}$ is finite as well. Thus we obtain the bound finite model property for our logic. A consequence of theorem 10 is the decidability for satisfiability, validity and model checking against the process semantics.

**Theorem 11 (Decidability of $\mathcal{L}_{DES}$).** *For* $\mathcal{L}_{DES}$ *validity, satisfiability and model checking are decidable against the process semantics.*

**Corollary 1 (Decidability of $\mathcal{L}_{DS}$).** *For* $\mathcal{L}_{DS}$ *validity, satisfiability and model checking are decidable against the process semantics.*

### 3.5 Axiomatic Systems

In Table 3 we propose a Hilbert-style axiomatic system for $\mathcal{L}_{DS}$. We assume the axioms and the rules of propositional logic. In addition we will have a set of spatial axioms and rules, and a set of dynamic axioms and rules.

**Spatial axioms**

S1: $\vdash \top|\bot \rightarrow \bot$
S2: $\vdash (\phi|\psi)|\rho \rightarrow \phi|(\psi|\rho)$
S3: $\vdash \phi|0 \leftrightarrow \phi$
S4: $\vdash \phi|(\psi \vee \rho) \rightarrow (\phi|\psi) \vee (\phi|\rho)$
S5: $\vdash \phi|\psi \rightarrow \psi|\phi$
S6: $\vdash (c_P \wedge \phi|\psi) \rightarrow \bigvee_{P \equiv Q|R}(c_Q \wedge \phi)|(c_R \wedge \psi)$

**Spatial rules**

SR1: $\vdash \phi \rightarrow \psi$ then $\vdash \phi|\rho \rightarrow \psi|\rho$

**Dynamic axioms**

D7: $\vdash \langle\alpha\rangle\phi|\psi \rightarrow \langle\alpha\rangle(\phi|\psi)$
D8: $\vdash [\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [a]\psi)$
D9: $\vdash 0 \rightarrow [\alpha]\bot$
D10: For $\alpha_i \neq \beta$, $\vdash \langle !\alpha_1\rangle\top|...|\langle !\alpha_n\rangle\top \rightarrow [\beta]\bot$
D11: $\vdash \langle !\alpha\rangle\phi \rightarrow [\alpha]\phi$

**Dynamic rules**

DR2: $\vdash \phi$ then $\vdash [\alpha]\phi$
DR4: $\vdash \bigvee_{P \in \mathfrak{P}^{act(\phi)+}_{(\!|\phi|\!)}} c_P \rightarrow \phi$ then $\vdash \phi$
DR3: If $\vdash \phi_1 \rightarrow [\alpha]\phi'_1$ and $\vdash \phi_2 \rightarrow [\alpha]\phi'_2$
*then* $\vdash \phi_1|\phi_2 \rightarrow [\alpha](\phi'_1|\phi_2 \vee \phi_1|\phi'_2)$

**Table 3.** The axiomatic system of $\mathcal{L}_{DS}$

Concerning the axioms and rules we make two observations. The disjunction involved in Axiom S6 is finitary, as we considered the processes up to structural congruence level. Also the disjunction involved in Rule DR4 has a finite number of terms, as a consequence of the finite model property.

The axiomatic system for $\mathcal{L}_{DES}$ is just an extension of the axiomatic system of $\mathcal{L}_{DS}$ with the set of epistemic axioms and rules presented in Table 4. Observe that Rule DR4 has been replaced by Rule DR'4, as this logic is sensitive to contexts (due to universal quantifier involved by the semantics of the epistemic operator).

For the epistemic axioms and rules we point on their similarities with the classic axioms of knowledge. Thus Axiom E12 is the classical (K)-axiom stating that our epistemic operator is a normal one, while Axiom E13 is just the necessity axiom, for the epistemic operator. Also Axiom E14 is well known in epistemic logics. It states that our epistemic agents satisfy *the positive introspection property*: if $P$ knows $\phi$ then it knows that it knows $\phi$. Axiom E15 states a variant of the *negative introspection*, saying that if an agent $P$ is active and if it doesn't know $\phi$, then it knows that it doesn't know $\phi$. These axioms are present in all the epistemic logics [15]. Axiom E16 is also interesting as it states the equivalence between *to be active* and *to know* for our epistemic agents.

**Dynamic rule**

DR'4: $\vdash \bigvee_{\mathcal{M} \in \mathfrak{M}_{(\!|\phi|\!)}^{act(\phi)_+}} c_{\mathcal{M}} \rightarrow \phi$ then $\vdash \phi$

**Epistemic axioms**

E12: $\vdash K_Q\phi \wedge K_Q(\phi \rightarrow \psi) \rightarrow K_Q\psi$
E13: $\vdash K_Q\phi \rightarrow \phi$
E14: $\vdash K_Q\phi \rightarrow K_QK_Q\phi$
E15: $\vdash K_Q\top \rightarrow (\neg K_Q\phi \rightarrow K_Q\neg K_Q\phi)$

E16: If $P \in \mathfrak{S}$ then $\vdash K_P\top \leftrightarrow c_P|\top$
E17: $\vdash K_Q\phi \leftrightarrow (K_Q\top \wedge K_0(K_Q\top \rightarrow \phi))$
E18: $\vdash K_0\phi \wedge \psi|\rho \rightarrow (K_0\phi \wedge \psi)|(K_0\phi \wedge \rho)$
E19: $\vdash K_0\phi \rightarrow [\alpha]K_0\phi$
E20: $\vdash K_0\phi \rightarrow (K_Q\top \rightarrow K_QK_0\phi)$

**Epistemic rules**

ER5: $\vdash \phi$ then $\vdash K_Q\top \rightarrow K_Q\phi$

ER6: If $\mathcal{M} \ni P$ is a finite context and
$\vdash c_{\mathcal{M}} \wedge c_P \rightarrow K_0\phi$ then $\vdash c_{\mathcal{M}} \rightarrow \phi$

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

### 3.6 Soundness and Completeness

The choice of the axioms is motivated by the soundness theorem.

**Theorem 12 (Soundness).** *The systems $\mathcal{L}_{DS}$ and $\mathcal{L}_{DES}$ are sound w.r.t. process semantics.*

Hence everything expressed by our axioms and rules about the process semantics is correct and, in conclusion, using our system, we can derive only theorems that can be meaningfully interpreted in CCS.

Further we state the completeness of $\mathcal{L}_{DS}$ and of $\mathcal{L}_{DES}$ with respect to process semantics. The intuition is that, because $c_P$ is a characteristic formulas, we should have an equivalence between $\mathcal{M}, P \models \phi$ and $\vdash c_P \rightarrow \phi$ for $\mathcal{L}_{DS}$, and between $\mathcal{M}, P \models \phi$ and $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \phi$ for $\mathcal{L}_{DES}$ (when $\mathcal{M}$ is a finite context). Using this intuition we proved the completeness theorem. Observe that $\mathcal{L}_{DS}$ logic is not sensitive to contexts, while $\mathcal{L}_{DES}$ is, because of the universal quantifier involved in the semantics of the epistemic operator.

**Theorem 13 (Completeness).** *The $\mathcal{L}_{DS}$ and $\mathcal{L}_{DES}$ are complete with respect to process semantics.*

The completeness ensures that everything that can be derived in the semantics can be proved as theorem. In this way we have the possibility to syntactically verify (prove) properties of distributed systems.

## 4 Concluding remarks

In this paper we developed two decidable and complete axiomatized logics for specifying and model-checking concurrent distributed systems: Dynamic Spatial Logic - $\mathcal{L}_{DS}$ and Dynamic Epistemic Spatial Logic - $\mathcal{L}_{DES}$. They extend

Hennessy-Milner logic with the parallel operator and respectively with epistemic operators. The lasts operators are meant to express global properties over contexts. We propose these operators as alternative to the guarantee operator of the classical spatial logics, in order to obtaining a logic adequately expressive and decidable.

$\mathcal{L}_{DES}$ is less expressive than the classic spatial logic. Using the guarantee operator and the characteristic formulas, we can express our epistemic operators in classic spatial logic, while guarantee operator cannot be expressed by using our logic: $K_Q\phi \stackrel{def}{=} c_Q|\top \wedge\ (\neg(c_Q|\top \rightarrow \phi) \triangleright \bot)$.

Validity and satisfiability in a model can be syntactically expressed in $\mathcal{L}_{DES}$. Combining this feature with the possibility to characterize processes and finite contexts, we may argue on utility of this logic.

In the context of decidability, our sound and complete Hilbert-style axiomatic systems provide powerful tools for making predictions on the evolution of the concurrent distributed systems. Knowing the current state or a sub-state of a system, we can characterize it syntactically. And because any other state can be characterized, we can project any prediction-like problem in syntax and verify its satisfiability. Hence if the system we considered can reach the state we check, we will obtain that the formula is satisfiable and this method will provide also a minimal model.

The axioms and rules considered are very similar to the classical axioms and rules in epistemic logic, and some derivable theorems state meaningful properties of epistemic agents. All these relates our logic with the classical epistemic/doxastic logics and focus the specifications on external observers as epistemic agents. This interpretation is consistent with the spirit of process algebras.

Further researches are to be considered such as adding other operators in logics to fit with more complex process calculi. Challenging will be also the perspective of considering recursion in semantics.

**Acknowledgements.** We thank to Alexandru Baltag for contributing with valuable comments, since the beginning, on the construction of this logic. Thanks also to Luca Cardelli for comments and related discussions. The name *structural bisimulation* was suggested to us by Gordon Plotkin.

## References

1. Bergstra, J.A.: Handbook of Process Algebra. Elsevier Science Inc., New York, NY, USA (2001)
2. Hennessy, M., Milner, R.: Algebraic laws for nondeterminism and concurrency. JACM **vol: 32(1)** (1985) 137–161
3. Stirling, C.: Modal and temporal properties of processes. Springer-Verlag New York, Inc., New York, NY, USA (2001)
4. Milner, R., Parrow, J., Walker, D.: Modal logics for mobile processes. Theoretical Computer Science **vol:114** (1993) 149–171
5. Dam, M.: Proof systems for $\pi$-calculus. (In de Queiroz, editor, Logic for Concurrency and Synchronisation, Studies in Logic and Computation. Oxford University Press. To appear)

6. Dam, M.: Model checking mobile processes. Information and Computation **vol:129(1)** (1996) 35–51
7. Caires, L., Cardelli, L.: A spatial logic for concurrency (part i). Information and Computation **Vol: 186/2** (November 2003) 194–235
8. Cardelli, L., Gordon, A.D.: Ambient logic. To appear in Mathematical Structures in Computer Science (2003)
9. Cardelli, L., Gordon, A.D.: Anytime, anywhere: Modal logics for mobile ambients. (2000) 365–377
10. Cardelli, L., Gordon, A.D.: Mobile ambients. In: Foundations of Software Science and Computation Structures: First International Conference, FOSSACS '98, Springer-Verlag, Berlin Germany (1998)
11. Caires, L., Lozes, E.: Elimination of quantifiers and decidability in spatial logics for concurrency. Volume vol:3170. (2004)
12. Charatonik, W., Talbot, J.M.: The decidability of model checking mobile ambients. Volume 2142 of Lecture Notes in Computer Science. (2001) 339–354
13. Charatonik, W., Gordon, A.D., Talbot, J.M.: Finite-control mobile ambients. In: ESOP '02: Proceedings of the 11th European Symposium on Programming Languages and Systems, Springer-Verlag (2002) 295–313
14. Milner, R.: A Calculus of Communicating Systems. Springer-Verlag New York, Inc. (1982)
15. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIT Press (1995)
16. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press (2000)
17. Baltag, A., Moss, L.S.: Logics for epistemic programs. In: Synthese: J. Symons, J. Hintikka. (Eds.), Knowledge, Rationality and Action, Springer **139 (2)** (2004) 165–224
18. J. Gerbrandy, W.G.: Reasoning about information change. Journal of Logic, Language and Information **6** (1997) 146–169
19. van Benthem, J.F.A.K.: Games in dynamic epistemic logic. Bulletin of Economic Research, Los Altos **53(4)** (2001) 219–248
20. Syverson, P., Cervesato, I.: The logic of authentication protocols. In: Riccardo Focardi, Roberto Gorrieri (Eds.): Foundations of Security Analysis and Design, Springer **LNCS 2117** (2001)
21. Mardare, R., Priami, C.: Dynamic epistemic spatial logics. Technical Report, 03/2006, Microsoft Research Center for Computational and Systems Biology, Trento, Italy (2006)
22. Mardare, R., Priami, C.: A decidable extension of hennessy-milner logic with spatial operators. Technical Report DIT-06-009, Informatica e Telecomunicationi, University of Trento (2006)
23. Mardare, R.: Logical analysis of complex systems: Dynamic epistemic spatial logics. PhD. thesis, DIT, University of Trento, Italy, available from http://www.dit.unitn.it/∼mardare/publications.htm (March 2006)
24. Calcagno, C., Cardelli, L., Gordon, A.D.: Deciding validity in a spatial logic for trees. (2003) 62–73