

Dynamic Epistemic Logic with Communication Actions

Mario Roberto Folhadela Benevides¹

*PESC/COPPE - Inst. de Matemática/DCC
Federal University of Rio de Janeiro
Rio de Janeiro, Rio de Janeiro*

Isaque Macalam Saab Lima²

*PESC/COPPE
Federal University of Rio de Janeiro
Rio de Janeiro, Rio de Janeiro*

Abstract

This work proposes a Dynamic Epistemic Logic with Communication Actions that can be performed concurrently. Unlike Concurrent Epistemic Action Logic introduced by Ditmarsch, Hoek and Kooi [15], where the concurrency mechanism is the so called *true concurrency*, here we use an approach based on process calculus, like CCS and CSP, and Action Models Logic. Our approach makes possible the proof of soundness, completeness and decidability, different from the others approaches. We present an axiomatisation and show that the proof of soundness, completeness and decidability can be done using a reduction method.

Keywords: Epistemic Logic, Dynamic Logic, Action Models, Dynamic Epistemic Logic, Concurrent Actions, Communication Action.

1 Introduction

Multi-Agent Epistemic Logic has been investigated in Computer Science [6] to represent and reason about agents (or groups of agents') knowledge and beliefs. Dynamic Logic aims to reason about actions (programs) and their effects [8]. Dynamic Epistemic Logic [16] is conceived to reason about actions that change agents (or groups of agents') epistemic state, i.e., actions which change agent's knowledge and beliefs.

The first Dynamic Epistemic Logic was proposed independently by [11] and [7] it is called Public Announcement Logic(PAL) . There are many other approaches but the one that is used in this work is the Action Model Logic proposed by [1,2].

¹ Email:mario@cos.ufrj.br

² Email:isaque@cos.ufrj.br

Concurrent Dynamic Epistemic Logic was introduced in [15] and it was intended to extend Epistemic Action Logic proposed by Van Ditmarsch in [14] with concurrent epistemic actions. In this extension they use a mechanism to deal with concurrency called "true concurrency" which is inspired on the Concurrent Propositional Dynamic Logic proposed by Peleg in [10]. An interesting work, entitled Logics of Communication and Knowledge, presented in [12], proposes a framework for modelling message passing situations that combines properties of dynamic epistemic semantics and history-based approaches, which consists of Kripke models with records of sent messages in their valuations. Another work that inspired us to represent communication actions as private epistemic action is [7].

Example: Consider that there are two students waiting for a message from a teacher to send back the homework and that one student does not know if the other received or responded the message. To represent this we need to model the following actions: teacher sending the message (send action), each student receiving (receive action) and responding (response action) the message independently. We also need to guarantee that: the receive action can not be performed before the send action, the response action can not be performed before the receive action and the students actions can be performed concurrently. Can we model this using Action Models Logic? Since this is a very small example one can argue that this can be done by using pre conditions and non deterministic choice to model all the possible paths. Now imagine the same situation with 100 students. It would be not so easy to model.

This work proposes a way to deal with concurrency and communication with Dynamic Epistemic Logic. We use an approach based on action models and process calculus, like CCS and CSP, which allow us to prove soundness, completeness and decidability. Different from [15], that implements concurrency on top of Epistemic Action Logic, we extend Action Models to deal with concurrency and communication. The proofs of soundness, completeness and decidability can be done using a reduction method.

In order to facilitate the proof of soundness, completeness, and decidability we restricted our concurrency approach. We do not deal with "true concurrency" like in [15]. Instead, we adopt the interleaving (non-deterministic choices of all possible paths) approach used in process algebras like CCS and CSP. Since we are based on Action Models we can use the pre-conditions to restrict actions that must be executed after another action. We do not deal with Common Knowledge, because this would make the proofs a little more tricky.

In sections 2, 3 and 4 we give a brief introduction to Multi-agent Epistemic Logic, Action Model Logic and Concurrent Dynamic Epistemic Logic. Next we present the Dynamic Epistemic Logic that we propose in this paper. The last section is the conclusion.

2 Multi-Agent Epistemic Logic

This section presents the Multi-Agent Epistemic Logic **S5_a**. All the definitions and theorems of this section are based on [16].

2.1 Language and Semantics

Definition 2.1 The Epistemic language consists of a countable set Φ of proposition symbols, a finite set \mathcal{A} of agents, a modality K_a for each agent a and the boolean connectives \neg and \wedge . The formulas are defined as follows:

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid K_a\varphi$$

where $p \in \Phi$, $a \in \mathcal{A}$.

Definition 2.2 A multi-agent epistemic *frame* is a tuple $\mathcal{F} = (S, R_a)$ where:

- S is a non-empty set of states;
- R_a is a binary relation over S , for each agent $a \in \mathcal{A}$;

Definition 2.3 A multi-agent epistemic *model* is a pair $\mathcal{M} = (\mathcal{F}, \mathbf{V})$, where \mathcal{F} is a frame and \mathbf{V} is a valuation function $\mathbf{V} : \Phi \rightarrow 2^S$. We call a rooted multi-agent epistemic model (\mathcal{M}, s) an epistemic state.

Definition 2.4 Given a multi-agent epistemic model $\mathcal{M} = ((S, R_a), \mathbf{V})$. The notion of satisfaction $\mathcal{M}, s \models \varphi$ is defined as follows:

1. $\mathcal{M}, s \models p$ iff $s \in \mathbf{V}(p)$
2. $\mathcal{M}, s \models \neg\phi$ iff $\mathcal{M}, s \not\models \phi$
3. $\mathcal{M}, s \models \phi \wedge \psi$ iff $\mathcal{M}, s \models \phi$ and $\mathcal{M}, s \models \psi$
4. $\mathcal{M}, s \models K_a\phi$ iff for all $s' \in S : sR_as' \Rightarrow \mathcal{M}, s' \models \phi$

2.2 Axiomatisation

- (i) All instantiations of propositional tautologies,
- (ii) $K_a(\varphi \rightarrow \psi) \rightarrow (K_a\varphi \rightarrow K_a\psi)$,
- (iii) $K_a\varphi \rightarrow \varphi$,
- (iv) $K_a\varphi \rightarrow K_aK_a\varphi$ (+ introspection),
- (v) $\neg K_a\varphi \rightarrow K_a\neg K_a\varphi$ (− introspection),

Inference Rules

M.P. $\varphi, \varphi \rightarrow \psi / \psi$ U.G. $\varphi / K_a\varphi$

Theorem 2.5 $\mathbf{S5}_a$ is sound and complete w.r.t its semantics.

Example 2.6 This example is from [16].

Suppose we have a card game with three cards: **0**, **1** and **2**, and three players **a**, **b** and **c**. Each player receives a card and do not know the other players cards.

We use proposition symbols $0_x, 1_x, 2_x$ for $x \in \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ meaning “player x has card **0**, **1** or **2**”. We name each state by the cards that each player has in that state, for instance 012 is the state where player **a** has card **0**, player **b** has card **1** and

player **c** has card **2**³. The following epsitemic model represents the epistemic state of each agent⁴.

$Hexa1 = \langle (S, R), \mathbf{V} \rangle$:

- $S = \{012, 021, 102, 120, 201, 210\}$
- $R = \{(012, 012), (012, 021), (021, 021), \dots\}$
- $\mathbf{V}(0_a) = \{012, 021\}, \mathbf{V}(1_a) = \{102, 120\}, \dots$

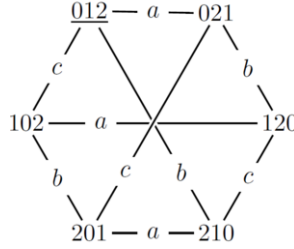


Fig. 1. Epistemic Model $Hexa1$

3 Action Models

All the definitions and theorems of this section are based on [16].

3.1 Language and Semantics

Definition 3.1 An action model \mathbf{M} is a structure $\langle S, \sim_a, \text{pre} \rangle$, where:

- S is a finite domain of action points or events;
- \sim_a is an equivalence relation on S , for each agent $a \in \mathcal{A}$;
- $\text{pre} : S \mapsto \mathcal{L}$ is a precondition function that assigns a precondition to each $s \in S$.

Routed action models is an action model with a distinguished state (\mathbf{M}, s) .

Note that S is different from S , \mathbf{M} is different from \mathcal{M} and s is different from s .

Definition 3.2 The Action Model language consists of a countable set Φ of proposition symbols, a finite set \mathcal{A} of agents, the boolean connectives \neg and \wedge , a modality K_a for each agent $a \in \mathcal{A}$ and a modality $[\alpha]$. The formulas are defined as follows:

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid K_a\varphi \mid [\alpha]\varphi,$$

$$\alpha ::= (\mathbf{M}, s) \mid \alpha_1; \alpha_2 \mid \alpha_1 \cup \alpha_2$$

where $p \in \Phi$, $a \in \mathcal{A}$, (\mathbf{M}, s) a rooted action model and $\langle \alpha \rangle \leftrightarrow \neg[\neg\alpha]$

Definition 3.3 Given an epistemic state (\mathcal{M}, s) with $\mathcal{M} = \langle (S, R_a), \mathbf{V} \rangle$ and a rooted action model (\mathbf{M}, s) with $\mathbf{M} = \langle S, \sim_a, \text{pre} \rangle$. The result of executing (\mathbf{M}, s) in (\mathcal{M}, s) is $(\mathcal{M} \otimes \mathbf{M}, (s, s))$ where $\mathcal{M} \otimes \mathbf{M} = \langle (S', R'_a), \mathbf{V}' \rangle$ such that:

³ A state name underlined means current state

⁴ We omit the reflexive loops in the picture.

- (i) $S' = \{(s, s) \text{ such that } s \in S, s \in S, \text{ and } \mathcal{M}, s \models \text{pre}(s)\}$
- (ii) $(s, s)R'_a(t, t) \text{ iff } (s R_a t \text{ and } s \sim_a t)$
- (iii) $(s, s) \in \mathbf{V}'(p) \text{ iff } s \in \mathbf{V}(p)$

Definition 3.4 Composition of rooted action models

Given rooted action models (\mathcal{M}, s) with $\mathcal{M} = \langle S, \sim, \text{pre} \rangle$ and (\mathcal{M}', s') with $\mathcal{M}' = \langle S', \sim', \text{pre}' \rangle$, their composition is the action model $(\mathcal{M}; \mathcal{M}', (s, s'))$ with $\mathcal{M}; \mathcal{M}' = \langle S'', \sim'', \text{pre}'' \rangle$:

- $S'' = \{(s, s') \text{ such that } s \in S, s' \in S' \}$
- $(s, s') \sim''_a (t, t') \text{ iff } (s \sim_a t \text{ and } s' \sim'_a t')$
- $\text{pre}''(s, s') = \langle (\mathcal{M}, s) \rangle \text{pre}'(s')$

Definition 3.5 Given a rooted epistemic state (\mathcal{M}, s) with $\mathcal{M} = \langle (S, R_a), \mathbf{V} \rangle$ and a rooted action model (\mathcal{M}, s) with $\mathcal{M} = \langle S, \sim, \text{pre} \rangle$. The notion of satisfaction $\mathcal{M}, s \models \varphi$ extends from 2.4 and is defined as follows

1, 2, 3, 4 as in definition 2.4

5. $\mathcal{M}, s \models [(M, s)]\phi \text{ iff } \mathcal{M}, s \models \text{pre}(s) \Rightarrow \mathcal{M} \otimes M, (s, s) \models \phi$

6. $\llbracket \alpha \cup \beta \rrbracket \text{ iff } \llbracket \alpha \rrbracket \cup \llbracket \beta \rrbracket$

7. $\llbracket (M, s); (M', s') \rrbracket \text{ iff } (M; M', (s, s')) \text{ Composition of action models}$

Where $\llbracket \cdot \rrbracket$ is the interpretation on a action model.

3.2 Axiomatisation

Epistemic Logic Axioms

Axioms (i), (ii), (iii), (iv) and (v) of section 2.2,

Action Model Logic Axioms

- (vi) $[(M, s)]p \leftrightarrow (\text{pre}(s) \rightarrow p),$
- (vii) $[(M, s)]\neg\phi \leftrightarrow (\text{pre}(s) \rightarrow \neg[(M, s)]\phi)$
- (viii) $[(M, s)](\phi \wedge \psi) \leftrightarrow ([(M, s)]\phi \wedge [(M, s)]\psi)$
- (ix) $[(M, s)]K_a\phi \leftrightarrow (\text{pre}(s) \rightarrow \bigwedge_{s \sim_a t} K_a[(M, t)]\phi)$
- (x) $[(M, s)][(M', s')]\phi \leftrightarrow [(M, s); (M', s')]\phi$
- (xi) $[(M, s) \cup (M', s')]\phi \leftrightarrow [(M, s)]\phi \wedge [(M', s')]\phi$

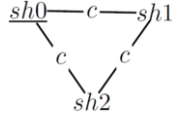
Inference Rules

M.P. $\varphi, \varphi \rightarrow \psi / \psi$ U.G. $\varphi / K_a\varphi$ $\varphi / [\alpha]\varphi$

Every formula in the language of action model logic without common knowledge is equivalent to a formula in the language of epistemic logic [16].

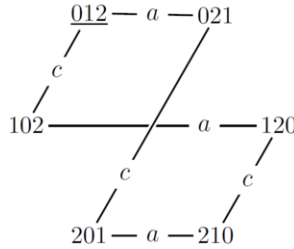
Example 3.6 Continuation of example 2.6

Suppose now agent **a** wants to perform the action of showing her card to agent **b**. In fact, we have three actions, agent **a** showing either card **0**, **1** or **2** to agent **b**. Agents **a** and **b** can distinguish between these three action but agent **c** cannot. This situation can be represented by the action model below.

Fig. 2. Action Model for *show*

- $S = \{sh0, sh1, sh2\}$
- $\sim_a = \{(s, s) \mid s \in S\}$
- $\sim_b = \{(s, s) \mid s \in S\}$
- $\sim_c = S \times S$
- $pre(sh0) = 0_a$
- $pre(sh1) = 1_a$
- $pre(sh2) = 2_a$

If agent **a** performs the action of showing her card to agent **b** on the epistemic model of example 2.6, we obtain:

Fig. 3. *Hexa1* After the Execution of *show*

This new epistemic model, shown in figure 3, is obtained by the product of epistemic model of figure 1 with the action model of figure 2. It is important to notice that the number of states after the product is 18 (6×3), but most of them are thrown out because they do not satisfy the precondition.

4 Epistemic Actions and Concurrent Dynamic Epistemic Logic

This section provides a brief introduction to the works presented in [15] and [16].

Epistemic Actions is an extension of Multi-Agent Epistemic Logic to deal with new information (updates), like Action Models, but it uses a different approach to deal with new information. Concurrent Dynamic Epistemic Logic proposes a way to deal with concurrency in Epistemic Actions.

4.1 Language and Semantics

Definition 4.1 The Epistemic Actions language consists of a countable set Φ of proposition symbols, a finite set \mathcal{A} of agents, the boolean connectives \neg and \wedge , a modality K_a for each agent $a \in \mathcal{A}$ and a modality $[\alpha]$. The formulas and the actions are defined as follows:

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid K_a\varphi \mid [\alpha]\varphi,$$

$$\alpha ::= ?\alpha \mid L_{\mathcal{B}}\beta \mid (\alpha!\alpha) \mid (\alpha!_i\alpha) \mid (\alpha;\beta) \mid (\alpha_1 \cup \alpha_2)$$

where $p \in \Phi$, $a \in \mathcal{A}$, $\mathcal{B} \subseteq \mathcal{A}$, L stands for learning and $L_{\mathcal{B}}\beta$ means ‘group \mathcal{B} learn that β ’, $?\alpha$ is a test, $(\alpha!\alpha)$ is called left local choice, $(\alpha!_i\alpha)$ is called right local choice, $(\alpha;\beta)$ is sequential composition (first α then β), $(\alpha_1 \cup \alpha_2)$ is non-deterministic choice.

Definition 4.2 Given the epistemic model $\mathcal{M} = \langle S, \sim_a, V \rangle$ and the state $s \in S$. The notion of satisfaction $\mathcal{M}, s \models \varphi$ extends from 2.4 and is defined as follows

1,2,3, 4 as in definition 2.4

5. $\mathcal{M}, s \models [\alpha]\phi$ iff for all $(\mathcal{M}', s') : (\mathcal{M}, s)[\alpha](\mathcal{M}', s')$ implies $(\mathcal{M}', s') \models \phi$

6. $(\mathcal{M}, s)[?\phi](\mathcal{M}', s')$ iff $\mathcal{M}' = \langle [\phi]_M, \emptyset, V \cap [\phi]_M \rangle$ and $s' = s$

7. $(\mathcal{M}, s)[L_G\phi](\mathcal{M}', s')$ iff $\mathcal{M}' = \langle S', \sim', V' \rangle$ and $(\mathcal{M}, s)[\phi]s'$

8. $\llbracket \alpha; \alpha' \rrbracket = \llbracket \alpha \rrbracket \circ \llbracket \alpha' \rrbracket$

9. $\llbracket \alpha \cup \alpha' \rrbracket = \llbracket \alpha \rrbracket \cup \llbracket \alpha' \rrbracket$

10. $\llbracket \alpha!\alpha' \rrbracket = \llbracket \alpha \rrbracket$

The Concurrent Dynamic Epistemic Logic language adds the concurrent execution operator to the actions of Epistemic Actions language. The actions are defined as follows:

$$\alpha ::= ?\alpha \mid L_{\mathcal{B}}\beta \mid (\alpha!\alpha) \mid (\alpha!_i\alpha) \mid (\alpha;\beta) \mid (\alpha_1 \cup \alpha_2) \mid (\alpha_1 \cap \alpha_2)$$

where $(\alpha_1 \cap \alpha_2)$ represents a concurrent execution.

Example 4.3 In order to illustrate the use of the language of Epistemic Actions, we consider the card game presented in section 3.

The Epistemic Model is the same shown in figure 1.

The action of “agent **a** showing her card to agent **b**” can be model as:

$$(L_{(b)}?0_a \cup L_{(b)}?1_a \cup L_{(b)}?2_a); (L_{(a,b,c)}?(K_b0_a \vee K_b1_a \vee K_b2_a))$$

This means that agent **a** tells agent **b** her card and after that all agents know that agent **b** knows the card that agent **a** holds. After performing this action, the resulting epistemic model is the same as in figure 3.

5 Dynamic Epistemic Logic with Communication Actions

5.1 Process Calculus

In this section, we propose a very small process (program) calculus for the programs of Dynamic Epistemic Logic with Communication Actions (DELWCA). It is inspired by [17].

Let $\mathcal{A} = \{1, \dots, n\}$, denoted by i, j, \dots , be a finite set of agents, $AMS = \{a_1, a_2, a_3, \dots\}$ be a finite set of action models and $\mathcal{N} = \{c_1, c_2, c_3, \dots, \overline{c_1}, \overline{c_2}, \overline{c_3}, \dots\}$ be a finite set of communication actions. As a convention, communication actions with one overline represent output and with no overlines represent an input. Communication actions can be combined to form a private action model, by joining an output communication action with its respective input ($[c_1, \overline{c_1}] = a_1$). The action model resultant of the join of two communication actions is known as silent action, denoted by $\tau_{i,j}^s(\cdot)$, that can be interpreted as the result of a communication between agents i and j ⁵.

Definition 5.1 *The language can be defined as follows.*

$$\eta ::= \alpha \mid \alpha.\eta \mid \eta_1;\eta_2 \mid \eta_1 + \eta_2, \text{ where } \alpha \in AMS \cup \mathcal{N}$$

$$\pi ::= \eta \mid \beta.\pi \mid \pi_1;\pi_2 \mid \pi_1 + \pi_2 \mid \eta_1 \parallel \eta_2 \cdots \parallel \eta_n$$

where $n = |\mathcal{A}|$ and η_i denotes the program performed by agent i .

We use π and η to denote processes (programs) and α and β to denote action models and communication actions.

The prefix operator $.$ denotes that the process will first perform the action α and then behave as π . The summation (or nondeterministic choice) operator $+$ denotes that the process will make a nondeterministic choice to behave as either π_1 or π_2 . The parallel composition operator \parallel denotes that the processes η_1, \dots, η_n , performed by agents $1, \dots, n$ respectively, may proceed independently or may communicate through a common channel.

We write $\pi \xrightarrow{\alpha} \pi'$ to express that the process π can perform the action α and after that behave as π' . We write $\pi \xrightarrow{\alpha} \sqrt{}$ to express that the process π successfully finishes after performing the action α . A process finishes when there is no possible action left for it to perform. For example, $\beta \xrightarrow{\beta} \sqrt{}$. When a process finishes inside a parallel composition, sequential composition or non-deterministic choice we write π instead of $\pi|\sqrt{}$, $\pi;\sqrt{}$ and $\pi + \sqrt{}$. We also write $\sqrt{}$ instead of $\sqrt{}|\sqrt{}$.

Like [9] we need to restrict the agents to perform some actions. In our case we don't want to perform communication actions, but we can perform τ action which results from the combination of communication actions $(\overline{a}, \overline{a})$.

⁵ As silent actions $\tau_{i,j}^s(\cdot)$ are interpreted as private action models, the index s denotes the root of the action model $\tau_{i,j}^s(\cdot)$.

The semantics of our process calculus can be given by the transition rules presented in table 1, where π and η are process specifications, while π' and η' are process specifications or $\sqrt{}$. The $\tau_{i,j}^s(\cdot)$ action represents an internal communication action from agent i to agent j .

| |
|--|
| $\alpha \xrightarrow{\alpha} \sqrt{}$ |
| $\alpha.\pi \xrightarrow{\alpha} \pi$ |
| $\frac{\pi_1 \xrightarrow{\alpha} \pi'_1}{\pi_1; \pi_2 \xrightarrow{\alpha} \pi'_1; \pi_2}$ |
| $\frac{\pi_1 \xrightarrow{\alpha} \pi'_1}{\pi_1 + \pi_2 \xrightarrow{\alpha} \pi'_1}$ |
| $\frac{\pi_2 \xrightarrow{\beta} \pi'_2}{\pi_1 + \pi_2 \xrightarrow{\beta} \pi'_2}$ |
| $\frac{\eta_i \xrightarrow{\alpha} \eta'_i}{(\eta_1 \parallel \dots \parallel \eta_i \parallel \dots \parallel \eta_n) \xrightarrow{\alpha} (\eta_1 \parallel \dots \parallel \eta'_i \parallel \dots \parallel \eta_n)}, \text{ for all } i, j \in \mathcal{A}$ |
| $\frac{\eta_i \xrightarrow{\alpha} \eta'_i, \eta_j \xrightarrow{\beta} \eta'_j}{(\eta_1 \parallel \dots \parallel \eta_i \parallel \dots \parallel \eta_j \parallel \dots \parallel \eta_n) \xrightarrow{\tau_{i,j}^s(\cdot)} (\eta_1 \parallel \dots \parallel \eta'_i \parallel \dots \parallel \eta'_j \parallel \dots \parallel \eta_n)}, \text{ for all } i, j \in \mathcal{A}$ |

Table 1
Transition Relation

Example 5.2 Continuation of the card game example.

Now suppose that the game is online and player **a** sends a message p to players **b** and **c**. So after the message p players **b** and **c** know all the cards. This problem can be modelled as follows:

- $\pi_1 = \overline{c_{ab}}(p); \overline{c_{ac}}(p) + \overline{c_{ac}}(p); \overline{c_{ab}}(p)$
- $\pi_2 = c_{ab}(\cdot).\beta$
- $\pi_3 = c_{ac}(\cdot).\gamma$
- $\pi_1 \parallel \pi_2 \parallel \pi_3$

Given the programs π_1, π_2 and π_3 , initially we have two possible actions: communication between **a** and **b** or communication between **a** and **c**. Suppose that the communication between **a** and **b** occurs first, then we will have two possible actions: communication between **a** and **c** or action β and so on ...

We can represent this using parallel composition:

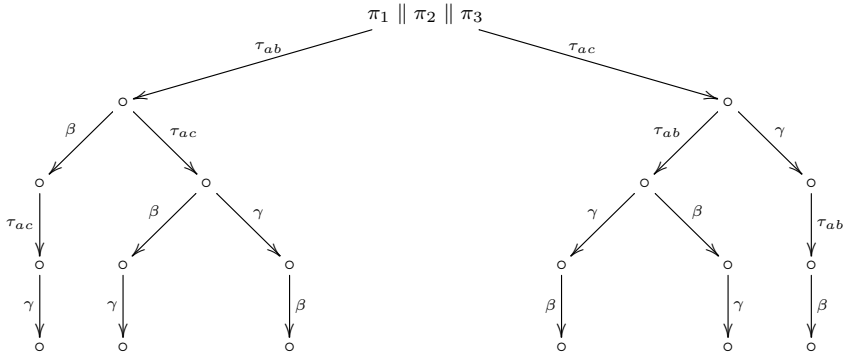


Fig. 4. Possible Runs of Process $\pi_1 \parallel \pi_2 \parallel \pi_3$

So :

- $\langle \pi_1 \parallel_2 \parallel_3 \rangle . (K_2p \wedge K_3p)$ is true
- $\langle \pi_1 \parallel_2 \parallel_3 \rangle . (K_2p \vee K_3p)$ is true
- $\langle \pi_1 \parallel_2 \parallel_3 \rangle . \neg (K_2p \vee K_3p)$ is false

5.2 Bisimulation

The concept of bisimulation is a key notion in any process algebra. It is an equivalence relation between processes which have mutually similar behavior. The intuition is that two bisimilar processes cannot be distinguished by an external observer. Using the notion of bisimulation allows us to transform any process in an equivalent one that is a summation of all their possible actions, that is what the Expansion Law (theorem 5.5) states.

There are two possible semantics for the τ action in CCS: it can be regarded as being observable, in the same way as the communication actions, or it can be regarded as being invisible. We adopt the first one, since it is more generic and fits better in our formalism. Whenever the τ action is observable the bisimulation relation is called *strong*.

Definition 5.3 ([9]) *Let Π be the set of all processes. A set $Z \subseteq \Pi \times \Pi$ is a strong bisimulation if $(\pi_1, \pi_2) \in Z$ implies the following for all $\alpha \in \text{AMS}$:*

- If $\pi_1 \xrightarrow{\alpha} \pi'_1$, then there is $\pi'_2 \in \Pi$ such that $\pi_2 \xrightarrow{\alpha} \pi'_2$ and $(\pi'_1, \pi'_2) \in Z$;
- If $\pi_2 \xrightarrow{\alpha} \pi'_2$, then there is $\pi'_1 \in \Pi$ such that $\pi_1 \xrightarrow{\alpha} \pi'_1$ and $(\pi'_1, \pi'_2) \in Z$;
- $\pi_1 \xrightarrow{\alpha} \surd$ if and only if $\pi_2 \xrightarrow{\alpha} \surd$.

Definition 5.4 ([9]) *Two process π and π' are strongly bisimilar (or simply bisimilar), denoted by $\pi \simeq \pi'$, if there is a strong bisimulation Z such that $(\pi, \pi') \in Z$.*

Now, we introduce the Expansion Law, which is very important in the definition of the semantic and in the axiomatisation of our logic. We present a particular case of the Expansion Law, which is suited to our needs. The most general case of the Expansion Law is presented in [9].

Theorem 5.5 ([9]) *[Expansion Law (EL)] Let $\pi = (\eta_1 \parallel \dots \parallel \eta_n)$. Then*

$$\pi \sim \sum_{\eta_i \xrightarrow{\alpha} \eta'_i} \alpha . (\eta_1 \parallel \dots \parallel \eta'_i \parallel \dots \parallel \eta_n) + \sum_{(\eta_i \xrightarrow{c} \eta'_i) \& (\eta_j \xrightarrow{\bar{c}} \eta'_j)} \tau_{i,j} . (\eta_1 \parallel \dots \parallel \eta'_i \parallel \dots \parallel \eta'_j \parallel \dots \parallel \eta_n)$$

where α is a action model and $\tau_{i,j}$ is a private action model resulted by the combination of two communication actions.

We denote the right side of this bisimilarity by $\text{Exp}(\pi)$. We also denote by $\mathbf{0}$ the processes whose expansion is empty, i.e., there is no $(\eta_i \xrightarrow{c} \eta'_i)$, $(\eta_j \xrightarrow{\bar{c}} \eta'_j)$ and $(\eta_k \xrightarrow{\alpha} \eta'_k)$ for any $i, j, k \in \{1, \dots, n\}$.

Proof. This follows from table 1 and definitions 5.3 and 5.4. A detailed proof for the most general case of this theorem can be found in [9]. \square

The Expansion Law is a very useful property of CCS processes. Its intuition is that processes can be rewritten as a summation of all their possible actions. Suppose we have a processes $A \stackrel{def}{=} c.A' + \alpha.A''$ and $B \stackrel{def}{=} \bar{c}.B' + \beta.B''$, then the process $(A \parallel B)$ is equivalent, using the Expansion Law, to

$$(A \parallel B) \simeq \alpha.(A'' \parallel B) + \beta.(A \parallel B'') + \tau_{AB}.(A' \parallel B')$$

5.3 Language

In this section we present the DELWCA language.

Definition 5.6 *The DELWCA language consists of a set Φ of countably many proposition symbols, a set Π of programs as defined in 5.1, a finite set \mathcal{A} of agents, the boolean connectives \neg and \wedge , a modality $\langle \pi \rangle$ for every program $\pi \in \Pi$ (as defined in section 5.1) and a modality K_a for each agent a . The formulas are defined as follows:*

$$\varphi ::= p \mid \top \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle \pi \rangle \varphi \mid K_i \varphi$$

where $p \in \Phi$, $\pi \in \Pi$, $i \in \mathcal{A}$ and $\langle \pi \rangle \varphi$ means that exists a execution of π that leads to a state where φ is true.

5.4 Semantics

For communication actions (actions in \mathcal{N}) we need to relax the fact that relations in action models are equivalence relations, we just need them to be relations. For this case all the definitions of action models (def. 3.1), execution (product) of action models (def. 3.3), composition of action models (def. 3.4) can be easily adapted.

Definition 5.7 *Let \mathcal{A} be the set of all agents and $i, j \in \mathcal{A}$. The action model $\tau_{i,j}^s(\varphi) = (\mathbf{M}, \mathbf{s})$, with $\mathbf{M} = \langle \mathbf{S}, \sim, \text{pre} \rangle$, is defined as follows:*

- $\mathbf{S} = \{\mathbf{s}, \mathbf{t}\}$
- $\sim_i = \{(\mathbf{s}, \mathbf{s}), (\mathbf{t}, \mathbf{t})\}$
- $\sim_j = \{(\mathbf{s}, \mathbf{s}), (\mathbf{t}, \mathbf{t})\}$
- $\sim_k = \{(\mathbf{s}, \mathbf{t}), (\mathbf{t}, \mathbf{t})\}$, for all $k \in \mathcal{A} \setminus \{i, j\}$
- $\text{pre}(\mathbf{s}) = \varphi$
- $\text{pre}(\mathbf{t}) = \top$

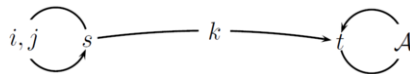


Fig. 5. Action Model for $\tau_{i,j}^s$

In order to obtain the definition of satisfaction for DELWCA we must add the following condition to definition 3.5:

$$\llbracket (\eta_1 \parallel \dots \parallel \eta_n) \rrbracket = \{ \llbracket \tau_{i,j}(\cdot) \rrbracket; \llbracket (\eta_1 \parallel \dots \parallel \eta'_i \parallel \dots \parallel \eta'_j \parallel \dots \parallel \eta_n) \rrbracket, \text{ for all } (\eta_i \xrightarrow{c_i} \eta'_i) \text{ \& } (\eta_j \xrightarrow{c_j} \eta'_j) \} \cup \{ \llbracket \alpha \rrbracket; \llbracket (\eta_1 \parallel \dots \parallel \eta'_i \parallel \dots \parallel \eta_n) \rrbracket, \text{ for all } (\eta_i \xrightarrow{\alpha} \eta'_i) \}$$

5.4.1 Axiomatisation

(i) All instantiations of propositional tautologies,

Epistemic Logic Axioms

Axioms (i), (ii), (iii), (iv) and (v) of section 2.2,

Action Model Axioms

Axioms (vi), (vii), (viii) and (ix) of section 3.2,

PDL Axioms

- (x) $[\pi](\phi \rightarrow \psi) \rightarrow ([\pi]\phi \rightarrow [\pi]\psi)$ (K axiom)
- (xi) $[\pi_1][\pi_2]\phi \leftrightarrow [\pi_1; \pi_2]\phi$ (Composition)
- (xii) $[\pi_1 + \pi_2]\phi \leftrightarrow [\pi_1]\phi \wedge [\pi_2]\phi$ (Non-deterministic Choice)
- (xiii) $[\alpha.\pi]\phi \leftrightarrow [\alpha][\pi]\phi$ (Prefix)⁶
- (xiv) $[\alpha.\pi]\phi \leftrightarrow pre(\alpha) \rightarrow [\pi]\phi$

Concurrent Action Axiom

- (xv) $[\eta_1 \parallel \dots \parallel \eta_n]\phi \leftrightarrow [Exp(\eta_1 \parallel \dots \parallel \eta_n)]\phi$

Inference Rules

$$\text{M.P. } \varphi, \varphi \rightarrow \psi / \psi \quad \text{U.G. } \varphi / [\pi]\varphi \quad \varphi / K_a\varphi$$

Proposition 5.8 $\vdash [\alpha; \pi_2]\phi \leftrightarrow [\alpha][\pi_2]\phi \leftrightarrow [\alpha.\pi_2]\phi \leftrightarrow pre(\alpha) \rightarrow [\pi_2]\phi$

Example 5.9 A supervisor Ane (1) and her two students Bob(2) and Cathy(3) are working in their computer located at their own house. The supervisor wants to book a meeting "tomorrow at 16:00". She sends a message asynchronously to Bob and Cathy. We are supposing that the supervisor uses channels c_{12} and c_{13} to communicate with Bob and Cathy respectively. We represent Anne, Bob and Cathy by processes π_1 , π_2 and π_3 respectively, and their parallel composition by $\pi_1 \parallel \pi_2 \parallel \pi_3$.

- $\pi_1 = \bar{c}_{12}(p); \bar{c}_{13}(p) + \bar{c}_{12}(p)$
- $\pi_2 = c_{12}(\cdot)$

⁶ It is important to notice that Prefix is a special case of Composition

- $\pi_3 = c_{13}(\cdot)$
- $\pi_{1\parallel 2\parallel 3} = (\pi_1 \parallel \pi_2 \parallel \pi_3)$

We have two possible runs process $\pi_{1\parallel 2\parallel 3}$ as shown in the tree in figure 6.

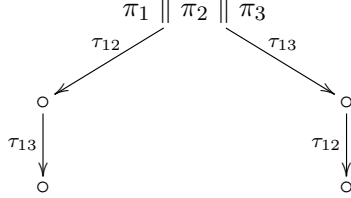


Fig. 6. Possible Runs of Process $\pi_1 \parallel \pi_2 \parallel \pi_3$

Let propositional symbol p represent "tomorrow at 16:00". The epistemic model \mathcal{M}_i at the begging is as shown in figure 7.

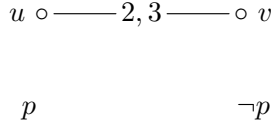


Fig. 7. Initial Epistemic Model \mathcal{M}_0

The action models for τ_{12} and τ_{13} are presented in figures 8 and 9.

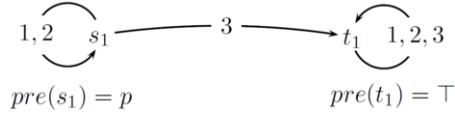


Fig. 8. Action Model for τ_{12}

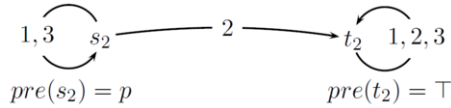


Fig. 9. Action Model for τ_{13}

Suppose τ_{12} is performed before τ_{13} . After the execution of τ_{12} we obtain the epistemic model picture in figure 10.

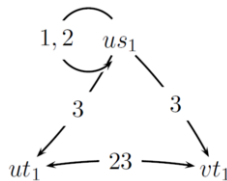


Fig. 10. Epistemic Model $\mathcal{M}_1 = \mathcal{M}_0 \otimes \tau_{12}$

It is important to notice that at state us_1 Ane and Bob knows p $\mathcal{M}_1, us_1 \vdash K_1p \wedge K_2p$ but Cath doesn't $\mathcal{M}_1, us_1 \vdash \neg K_3p$. After the second communication τ_{13} we have the epistemic model of figure 11.

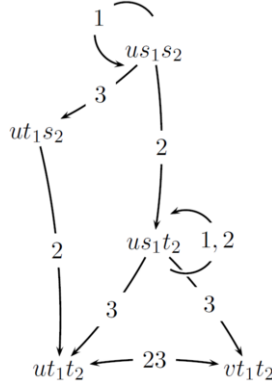


Fig. 11. Epistemic Model $\mathcal{M}_2 = \mathcal{M}_1 \otimes \tau_{13} = \mathcal{M}_0 \otimes \tau_{12} \otimes \tau_{13}$

We can notice, from figure 11 that at state us_1s_2 Ane, Bob and Cath knows p $\mathcal{M}_2, us_1s_2 \vdash K_1p \wedge K_2p \wedge K_3p$ as expected. If we execute run $\tau_{13}; \tau_{12}$ we obtain the model \mathcal{M}_3 as shown in figure 12.

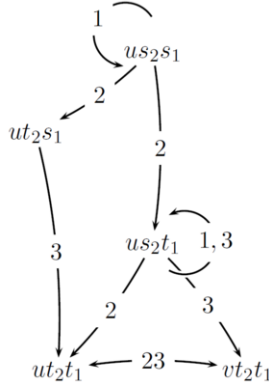


Fig. 12. Epistemic Model $\mathcal{M}_3 = \mathcal{M}_0 \otimes \tau_{13} \otimes \tau_{12}$

We can show, from figure 12, that Ane, Bob and Cath know p $\mathcal{M}_3, us_1s_2 \vdash K_1p \wedge K_2p \wedge K_3p$ as expected.

5.5 Soundness, Completeness and Decidability

5.5.1 Soundness

We need to prove that all axioms are valid. Axioms *i* to *xiii* are standard from Dynamic Epistemic Logic literature and can be found in [16]. We prove validity only for axiom 6.

Lemma 5.10 $[\eta_1 \parallel \dots \parallel \eta_n]\phi \leftrightarrow [Exp(\eta_1 \parallel \dots \parallel \eta_n)]\phi$ is valid.

Proof. This proof follows straightforward from the rules of table 1 and theorem 5.5. \square

A detailed proof of soundness can be found in [3].

5.5.2 Completeness

The proof of completeness is similar to the proof for Public Announcement and Action Models Logics introduced in [13] Dynamic Epistemic Logic. We prove completeness showing that every formula in DELWCA is equivalent to formula in Epistemic Logic. In order to achieve that we only have to provide a translation function that translate every DELWCA formula to a formula without communication actions and concurrency.

A detailed proof of completeness can be found in [3].

5.5.3 Decidability

Decidability follows directly from the decidability of $\mathbf{S5}_a$.

6 Conclusions

In this work we present a Dynamic Epistemic Logic with Communication Actions that can be performed concurrently. In order to achieve that we propose a PDL like language for actions and develop a small process calculus. We show that it's easy to model problems of communication and concurrency with the proposed dynamic epistemic logic. The main feature of it is the Expansion rule which allows for representing the parallel composition operator. This approach is similar to the one introduced in [4, 5].

We represent communication actions as private Action Models where the relations are not equivalence relations. We present an axiomatisation and prove completeness using reduction technique.

As future work we would like to investigate the extension with common knowledge and/or iteration operators, study other types of communications where agents are not reliable or not trustful, extend this to Dynamic Epistemic Logic With Post-Conditions and change DEMO, or create a new Model Checker, to deal with concurrency and communication.

References

- [1] Baltag, A. and L. Moss, *Logics for epistemic programs*, Synthese (2004), pp. 165–224.
- [2] Baltag, A., L. Moss and Solecki, *The logic of common knowledge, public announcements and private suspension*, in: I. Gilboa, editor, *The 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 98)*, 1998, pp. 43–56.
- [3] Benevides, M. and I. Lima, *Dynamic epistemic logic with communication actions*, Technical Report 1902.01164, ArXiv (2019).
URL <http://arxiv.org/abs/1902.01164>

- [4] Benevides, M. and L. Schechter, *A propositional dynamic logic for CCS programs*, in: *Proceedings of the XV Workshop on Logic, Language, Information and Computation*, LNAI **5110** (2008), pp. 83–97.
- [5] Benevides, M. and L. Schechter, *A propositional dynamic logic for concurrent programs based on the π -calculus*, in: *Proceedings of Methods for Modalities*, Electronic Notes in Theoretical Computer Science ENTCS **262** (2010), pp. 49–64.
- [6] Fagin, R., J. Halpern, Y. Moses and M. Vardi, “Reasoning about Knowledge,” MIT Press, USA, 1995.
- [7] Gerbrandy, J. and W. Groeneveld, *Reasoning about information change*, Journal of Logic, Language, and Information (1997), pp. 147–169.
- [8] Harel, D., “Dynamic Logic,” Handbook of Philosophical Logic, Dordrecht:Reidel, Vol.2, 1984, ed. D. Gabbay and F. Guenther.
- [9] Milner, R., “Communication and Concurrency,” Prentice Hall, 1989.
- [10] Peleg, D., *Communication in concurrent dynamic logic*, Journal of Computer and System Sciences **35** (1987).
- [11] Plazza, J. A., “Logics of Public Communications,” Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems, North-Holland: 201–216, 1989.
- [12] Sietsma, F., “Logics of Communication and Knowledge,” Ph.D. thesis, University of Amsterdam (CWI-ILLC) (2012).
- [13] van Benthem, J., J. van Eijck and B. P. Kooi, *Logics of communication and change*, Inf. Comput. **204** (2006), pp. 1620–1662.
- [14] van Ditmarsch, H., “The Semantics of Concurrent Knowledge Actions,” Workshop on Logic and Games, Ed. M. Pauly and G. Sandu, ESSLII, 2001.
- [15] van Ditmarsch, H., W. van der Hoek and B. Kooi, “Concurrent Dynamic Epistemic Logic,” Kluwer, Ed. V.F. Hendricks et al., , vol. 322, 2003.
- [16] van Ditmarsch, H., W. van der Hoek and B. Kooi, “Dynamic Epistemic Logic,” Synthese Library Series, volume 337, Springer, The Netherland, 2008.
- [17] van Glabbeek, R. J., *The linear time - branching time spectrum i: The semantics of concrete, sequential processes*, in: A. P. J.A. Bergstra and S. Smolka, editors, *Handbook of Process Algebra, chapter 1*, Elsevier, 2001 pp. 3–99.