

Dynamic Epistemic Logic with Assignments, Concurrency and Communication Actions

Mario R. F. Benevides^{1,2}, Isaque M. S. Lima²

¹Instituto de Computação
Universidade Federal Fluminense (UFF)
Niterói - RJ

²PESC/COPPE
Universidade Federal Rio de Janeiro (UFRJ)
Rio de Janeiro - RJ

mario@ic.uff.br, isaque@cos.ufrj.br

Abstract. *This work proposes an extension of the Dynamic Epistemic Logic with Communication Actions by adding the notion of postconditions from the Dynamic Epistemic Logic with Assignments to deal with boolean assignments to action models. Other concurrent logics, like the Concurrent Epistemic Action Logic introduced by Ditmarsch, Hoek and Kooi, do not deal with boolean assignments. We present an axiomatization and show that the proof of soundness, completeness and decidability can be done using a reduction method.*

Resumo. *Este trabalho propõe uma extensão da Lógica Epistêmica Dinâmica com Ações de Comunicação adicionando a noção de pós-condições da Lógica Epistêmica Dinâmica com Atribuições para lidar com atribuições booleanas para modelos de ação. Outras lógicas concorrentes, como a Lógica Concorrente de Ação, introduzida por Ditmarsch, Hoek e Kooi, não tratam de atribuições booleanas. Apresentamos uma axiomatização e mostramos que a prova de correção, completude e decidibilidade podem ser feitas usando um método de redução.*

1. Introduction

Multi-Agent Epistemic Logic has been investigated in Computer Science [Fagin et al. 1995] to represent and reason about agents (or groups of agents') knowledge and beliefs. Dynamic Logic aims to reason about actions (programs) and their effects [Harel 1984]. Dynamic Epistemic Logic [van Ditmarsch et al. 2008] is conceived to reason about actions that change agents (or groups of agents') epistemic state, i.e., actions which change agents knowledge and beliefs.

The first Dynamic Epistemic Logic was proposed independently by [Plaza 1989] and [Gerbrandy and Groeneveld 1997] 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 [Baltag and Moss 2004, Baltag et al. 1998]. We choose this approach because both [Benevides and Lima 2019] and [Benevides and Lima 2017] are extensions of the Action Model Logic.

Concurrent Dynamic Epistemic Logic was introduced in [van Ditmarsch et al. 2003] and it was intended to extend Epistemic Action Logic proposed by Van Ditmarsch

in [van Ditmarsch 2001] 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 [Peleg 1987]. An interesting work, entitled Logics of Communication and Knowledge, presented in [Sietsma 2012], proposes a framework for modelling message passing situations that combine properties of dynamic epistemic semantics and history-based approaches, which consists of Kripke models with records of sent messages in their valuations. The Dynamic Epistemic Logic with Communication Actions [Benevides and Lima 2019] proposes another approach to deal with concurrent actions, where the communication actions are represented as private action models and can be executed concurrently, they were inspired by the work of [Gerbrandy and Groeneveld 1997] to represent communication actions as private actions.

Action models with postconditions were proposed in [Benevides and Lima 2017] as an extension of Action Models [van Ditmarsch et al. 2003] to deal with boolean assignments. There are other works that deal with assignments, like [van Ditmarsch et al. 2005], but they are not based on the Action Model, in fact, they create a new mechanism to make the assignments. Since [Benevides and Lima 2019] is based on the Action Models, it is more straightforward for us to use [Benevides and Lima 2017] to deal with assignments.

Example: Consider that there are two students waiting for a message from a teacher to tell them which exercise each one should do, that the students must respond to the teacher with ACK 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: the 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 cannot be performed before the send action, the response action cannot be performed before the receive action and the students’ actions can be performed concurrently. We can model this using [Benevides and Lima 2019]. Now imagine that the teacher message was wrong and that he wants to inform this to his students. The [Benevides and Lima 2019] logic does not have assignments, so we cannot change the value of the prepositions. The [Benevides and Lima 2017] logic has assignment but does not have concurrent/communication actions so we cannot model the teacher sending the message to both students at the same time. So it’s needed a new logic that can deal at the same time with assignments and concurrent/communication actions.

This work proposes a way to combine the concepts of two Dynamic Epistemic Logics, one that deals with assignments [Benevides and Lima 2017] and another that deals with concurrency and communication [Benevides and Lima 2019], resulting in a new Dynamic Epistemic Logic to deal with assignments, concurrency and communication. This approach allows us to prove soundness, completeness and decidability. The proofs of soundness, completeness and decidability can be done using a reduction method.

In order to facilitate the proofs of soundness, completeness, and decidability we restricted our concurrency approach, like [Benevides and Lima 2019]. We do not deal with “true concurrency” like in [van Ditmarsch et al. 2003]. Instead, like [Benevides and Lima 2019], 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 the 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 more complicated, we will leave this as future work.

In Sections 2 and 3 we give a brief introduction to Action Model with Assignments 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. Action Model with Assignments

This section is based on [Benevides and Lima 2017].

In Action Model Logic we cannot change the value of the propositions. Action Model with Assignments is an extension of Action Models to deal with boolean proposition assignments.

2.1. Syntax and Semantics

Definition 2.1 *The Action Model language (\mathcal{L}) 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:*

$$\begin{aligned}\varphi &::= p \mid \top \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid K_a\varphi \mid [\alpha]\varphi, \\ \alpha &::= (\mathbf{M}, \mathbf{s}) \mid \alpha_1; \alpha_2 \mid \alpha_1 \cup \alpha_2\end{aligned}$$

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

Definition 2.2 *A action model with assignments \mathbf{M} consist of a structure $\langle S, \sim, \text{pre}, \text{pos} \rangle$, where:*

- S is a finite domain of action points or events,
- \sim_a is the equivalence relation in S for the agent a ,
- $\text{pre} : S \mapsto \mathcal{L}$ is the precondition function that assigns a precondition for each $s \in S$,
- $\text{pos}(s) = \{(p, x) \mid \forall p \in \Phi \text{ and } x = \text{true or false}\}$.

The language of the action model with assignments is the same language of the action model without assignment. The premise that if an agent can differentiate the two actions consequently he can differentiate the states resulting from these actions, still holds.

Definition 2.3 *Given an epistemic state (\mathcal{M}, s) with $\mathcal{M} = \langle S, \sim_a, V \rangle$ and a rooted action model (\mathbf{M}, \mathbf{s}) with $\mathbf{M} = \langle S, \sim, \text{pre}, \text{pos} \rangle$, the result of a execution of (\mathbf{M}, \mathbf{s}) in (\mathcal{M}, s) is $(\mathcal{M} \otimes \mathbf{M}, (s, \mathbf{s}))$ where $\mathcal{M} \otimes \mathbf{M} = \langle S', \sim', V' \rangle$ such that:*

1. $S' = \{(s, \mathbf{s}) \mid s \in S, \mathbf{s} \in S, \text{ and } \mathcal{M}, s \models \text{pre}(\mathbf{s})\}$,
2. $(s, \mathbf{s}) \sim'_a (t, \mathbf{t})$ iff $(s \sim_a t \text{ and } \mathbf{s} \sim_a \mathbf{t})$,
3. $V'(p) = \{(s, \mathbf{s}) \mid (p, V) \in \text{pos}(\mathbf{s})\}$.

Definition 2.4 *Given a rooted action model (\mathbf{M}, \mathbf{s}) with $\mathbf{M} = \langle S, \sim, \text{pre}, \text{pos} \rangle$, the definition of $fpos(\mathbf{s})$ is:*

1. $L(\mathbf{s}) = \{p \mid (p, V) \in \text{pos}(\mathbf{s})\}$, set of true propositions in \mathbf{s} .
2. $p_1, \dots, p_h \in L(\mathbf{s})$.
3. $q_1, \dots, q_m \notin L(\mathbf{s})$.
4. $fpos(\mathbf{s}) = p_1 \wedge \dots \wedge p_h \wedge \neg q_1 \wedge \dots \wedge \neg q_m$.

Definition 2.5 *Given the rooted action models (\mathbf{M}, \mathbf{s}) with $\mathbf{M} = \langle S, \sim, \text{pre}, \text{pos} \rangle$ and $(\mathbf{M}', \mathbf{s}')$ with $\mathbf{M}' = \langle S', \sim', \text{pre}', \text{pos}' \rangle$, their composition is the model of action $(\mathbf{M}; \mathbf{M}', (s, \mathbf{s}'))$ with $\mathbf{M}; \mathbf{M}' = \langle S'', \sim'', \text{pre}'', \text{pos}'' \rangle$:*

- $S'' = \{(s, s') \mid s \in S, s' \in S' \wedge \text{pos}(s) \rightarrow \text{pre}(s')\}$
- $(s, s') \sim_a'' (t, t') \text{ iff } (s \sim_a t \wedge s' \sim_a t')$
- $\text{pre}''(s, s') = \text{pre}(s)$
- $\text{pos}''(s, s') = \text{pos}'(s')$

Incompatible states are eliminated by the precondition. In Action Model with Assignments, the order of the composition is important, since an action can change the value of a proposition.

2.2. Axiomatization

The following is an axiomatization for the Action Model with Assignment

$$fpos(s) = p_1 \wedge \dots \wedge p_n \wedge \neg p_{n+1} \wedge \dots \wedge \neg p_m.$$

Axioms

1. $[M, s]p \leftrightarrow (\text{pre}(s) \rightarrow (fpos(s) \rightarrow p)), \quad \bigwedge_{s \sim_a t} K_a[M, t]\phi,$
2. $[M, s]\neg\phi \leftrightarrow (\text{pre}(s) \rightarrow \neg[M, s]\phi),$
3. $[M, s](\phi \wedge \psi) \leftrightarrow ([M, s]\phi \wedge [M, s]\psi),$
4. $[M, s]K_a\phi \leftrightarrow (\text{pre}(s) \rightarrow$
5. $[[M, s] \cup [M', s']]\phi \leftrightarrow [M, s]\phi \wedge [M', s']\phi,$
6. $[M, s][M', s']\phi \leftrightarrow [(M, s); (M', s')]\phi.$

Inference Rules

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

where σ is a uniform substitution of formulas by propositional variables.

3. Dynamic Epistemic Logic with Communication Actions

This section is based on [Benevides and Lima 2019].

3.1. Process Calculus

In this section, we present a very small process (program) calculus for the programs of Dynamic Epistemic Logic with Communication Actions (DELWCA) proposed in [Benevides and Lima 2019].

Let $\mathcal{A} = \{1, \dots, n\}$, denoted by i, j, \dots , be a finite set of agents, $\text{AMS} = \{a_1, a_2, a_3 \dots a_m\}$ be a finite set of action models and $\mathcal{N} = \{c_1, c_2, c_3, \dots, c_k, \overline{c_1}, \overline{c_2}, \overline{c_3}, \dots, \overline{c_k}\}$ 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 from 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 ¹.

Definition 3.1 *The Dynamic Epistemic Logic with Communication Actions language can be defined as follows.*

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

¹As silent actions $\tau_{i,j}^s(\cdot)$ are interpreted as private action models, the index s denotes the root of the action model $\tau_{i,j}^s(\cdot)$.

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

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.

3.2. Language

In this section we present the DELWCA language.

Definition 3.2 *The DELWCA language consists of a set Φ of countably many proposition symbols, a set Π of programs as defined in 3.1, 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 3.1) 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.

3.3. Semantics

For communication actions (actions in \mathcal{N}) they [Benevides and Lima 2019] relax the fact that relations in action models are equivalence relations.

Definition 3.3 *Let \mathcal{A} be the set of all agents and $i, j \in \mathcal{A}$. The action model $\tau_{i,j}^s(\varphi) = (M, s)$, with $M = \langle S, \sim, \text{pre} \rangle$, is defined as follows:*

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

3.3.1. Axiomatization

- | | |
|--|--|
| <ol style="list-style-type: none"> 1. <i>All propositional tautologies,</i> 2. $K_a(\varphi \rightarrow \psi) \rightarrow (K_a\varphi \rightarrow K_a\psi),$ 3. $K_a\varphi \rightarrow \varphi,$ 4. $K_a\varphi \rightarrow K_aK_a\varphi,$ 5. $\neg K_a\varphi \rightarrow K_a\neg K_a\varphi,$ <p style="text-align: center;">Action Model Axioms</p> | <ol style="list-style-type: none"> 6. $[(M, s)]p \leftrightarrow (\text{pre}(s) \rightarrow p),$ 7. <i>Axioms (2) to (4) of section 2.2,</i> <p style="text-align: center;">PDL Axioms</p> <ol style="list-style-type: none"> 8. $[\pi](\phi \rightarrow \psi) \rightarrow ([\pi]\phi \rightarrow [\pi]\psi)$ 9. $[\pi_1][\pi_2]\phi \leftrightarrow [\pi_1; \pi_2]\phi$ 10. $[\pi_1 + \pi_2]\phi \leftrightarrow [\pi_1]\phi \wedge [\pi_2]\phi$ 11. $[\alpha.\pi]\phi \leftrightarrow [\alpha][\pi]\phi^2$ |
|--|--|

²It is important to notice that Prefix is a special case of Composition

12. $[\alpha.\pi]\phi \leftrightarrow pre(\alpha) \rightarrow [\pi]\phi$ 13. $[\eta_1 \parallel \dots \parallel \eta_n]\phi \leftrightarrow [Exp(\eta_1 \parallel \dots \parallel \eta_n)]\phi$
Concurrent Action Axiom

Inference Rules

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

Proposition 3.1 $\vdash [\alpha; \pi_2]\phi \leftrightarrow [\alpha][\pi_2]\phi \leftrightarrow [\alpha.\pi_2]\phi \leftrightarrow pre(\alpha) \rightarrow [\pi_2]\phi$

4. Dynamic Epistemic Logic with Assignments, Concurrency and Communication Actions

To deal with assignments, concurrency and communication actions we need to combine the concepts of [Benevides and Lima 2017] and [Benevides and Lima 2019]. Our approach is to add the assignment mechanism from [Benevides and Lima 2017] into [Benevides and Lima 2019], creating a new dynamic epistemic logic (DELWACCA).

4.1. Syntax and Semantics

Definition 4.1 *The DELWACCA language is that same of DELWCA and consists of a set Φ of countably many proposition symbols, a set Π of programs as defined in 3.1, 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 3.1) and a modality K_a for each agent a . The formulas are defined as follows:*

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

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.

Definition 4.2 *A action model with assignments, concurrency and communication M consist of a structure $\langle S, \sim, pre, pos \rangle$, where:*

- S is a finite domain of action points or events,
- \sim_a is the equivalence relation in S for the agent a ,
- $pre : S \mapsto \mathcal{L}$ is the precondition function that assigns a precondition for each $s \in S$,
- $pos(s) = \{(p, x) \mid \forall p \in \Phi \text{ and } x = V \text{ or } F\}$.

The definition of the action model with assignments, concurrency and communication is the same as the definition of action model with assignments.

Definition 4.3 *Let \mathcal{A} be the set of all agents and $i, j \in \mathcal{A}$. The action model $\tau_{i,j}^s(\varphi) = (M, s)$, with $M = \langle S, \sim, pre, pos \rangle$, is defined as follows:*

- | | | |
|----------------------|-----------------------------------|--|
| • $S = \{s, t\}$ | • $pos(s) = \{(\varphi, true)\}$ | • $\sim_j = \{(s, s), (t, t)\}$ |
| • $pre(s) = \varphi$ | • $pos(t) = \{(\varphi, false)\}$ | • $\sim_k = \{(s, t), (t, t)\}$, for |
| • $pre(t) = \top$ | • $\sim_i = \{(s, s), (t, t)\}$ | all $k \in \mathcal{A} \setminus \{i, j\}$ |

In DELWACCA, like [Benevides and Lima 2017], the order of the actions matter, because a postcondition of action can change the execution of the other action. In [Benevides and Lima 2019] executing $(\alpha; \beta)$ is the same to executing $(\beta; \alpha)$, but this does not hold in DELWACCA.

In order to obtain the definition of satisfaction for DELWACCA we add the following condition (from 3.3) to definition 2.4:

$$\llbracket (\eta_1 \parallel \dots \parallel \eta_n) \rrbracket = \{ \llbracket \tau_{i,j}(\cdot) \rrbracket; \llbracket (\eta_1 \parallel \dots \parallel \eta'_i \parallel \dots \parallel \eta'_j \parallel \dots \parallel \eta_n) \rrbracket, \text{ for all } (\eta_i \xrightarrow{c_1} \eta'_i) \ \& \ (\eta_j \xrightarrow{c_2} \eta'_j) \} \cup \{ \llbracket \alpha \rrbracket; \llbracket (\eta_1 \parallel \dots \parallel \eta'_i \parallel \dots \parallel \eta_n) \rrbracket, \text{ for all } (\eta_i \xrightarrow{\alpha} \eta'_i) \}$$

4.1.1. Axiomatization

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. <i>All propositional tautologies,</i> 2. <i>Axioms (2) to (5) of section 3.3.1,</i> 3. <i>Axioms (1) to (4) of section 2.2,</i> | <p style="text-align: center;">PDL Axioms</p> <ol style="list-style-type: none"> 4. <i>Axioms (8) to (11) of section 3.3.1,</i> 5. $[\alpha.\pi]\phi \leftrightarrow pre(\alpha) \rightarrow (fpos(s) \rightarrow [\pi]\phi)$ 6. <i>Axiom (13) of section 3.3.1,</i> <p style="text-align: center;">Concurrent Action Axiom</p> |
|--|---|

Inference Rules

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

Proposition 4.1 $\vdash [\alpha; \pi_2]\phi \leftrightarrow [\alpha][\pi_2]\phi \leftrightarrow [\alpha.\pi_2]\phi \leftrightarrow pre(\alpha) \rightarrow (fpos(s) \rightarrow [\pi_2]\phi)$

4.2. Soundness, Completeness and Decidability

4.2.1. Soundness

We need to prove that all axioms are valid. Almost all axioms are from other dynamic epistemic logic, like [van Ditmarsch et al. 2008], [Benevides and Lima 2017] and [Benevides and Lima 2019]. The only axiom that was introduced by us (5) can be seen as a particular case of axiom (6) from section 2.2. So all axioms are valid.

4.2.2. Completeness and Decidability

The proof of completeness is similar to the proof for Public Announcement and Action Models Logics introduced in [van Benthem et al. 2006] Dynamic Epistemic Logic. We prove completeness showing that every formula in DELWACCA is equivalent to a formula in Epistemic Logic. In order to achieve that we only have to provide a translation function that translates every DELWACCA formula to a formula without assignments, concurrency and communication actions. The only difference to the proof in [Benevides and Lima 2019] is that now we have given a translation to postconditions, which was already done in [Benevides and Lima 2017]. The proof is straightforward, since we have all the translations from [Benevides and Lima 2017] and [Benevides and Lima 2019], we only need to change a constant value in the cost of $c(M, s)$, this constant value is bounded by the length of $fpos(s)$. Decidability follows directly from the decidability of $S5_a$.

5. Conclusions

In this work we present a Dynamic Epistemic Logic with Assignments, Concurrency and Communication Actions. In order to achieve that we combine process calculus from [Benevides and Lima 2019] with the assignments from [Benevides and Lima 2017].

We show that is possible to combine these two logics into one and highlight the limitations. The main feature of it is the addition of postconditions to the Action Models from [Benevides and Lima 2019]. We present an axiomatization and show how to prove soundness, completeness and decidability.

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, change DEMO to deal with this dynamic epistemic logic, or create a new Model Checker, to deal with assignments, concurrency and communication.

Referências

- Baltag, A. and Moss, L. (2004). Logics for epistemic programs. *Synthese*, (139):165–224.
- Baltag, A., Moss, L., and Solecki, S. (1998). The logic of common knowledge, public announcements and private suspicion. In Gilboa, I., editor, *The 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 98)*, pages 43–56.
- Benevides, M. and Lima, I. (2017). Action models with postconditions. *Computacion y Sistemas*, 21:401–406.
- Benevides, M. and Lima, I. (2019). Dynamic epistemic logic with communication actions. *Electronic Notes in Theoretical Computer Science*, pages 67–82.
- Fagin, R., Halpern, J., Moses, Y., and Vardi, M. (1995). *Reasoning about Knowledge*. MIT Press, USA.
- Gerbrandy, J. and Groeneveld, W. (1997). Reasoning about information change. *Journal of Logic, Language, and Information*, (6):147–169.
- Harel, D. (1984). *Dynamic Logic*. Handbook of Philosophical Logic. Dordrecht:Reidel, Vol.2. ed. D. Gabbay and F. Guenther.
- Peleg, D. (1987). Communication in concurrent dynamic logic. *Journal of Computer and System Sciences*, 35:23–58.
- Plaza, J. A. (1989). *Logics of Public Communications*. Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems. North-Holland: 201–216.
- Sietsma, F. (2012). *Logics of Communication and Knowledge*. PhD thesis, University of Amsterdam (CWI-ILLC).
- van Benthem, J., van Eijck, J., and Kooi, B. P. (2006). Logics of communication and change. *Inf. Comput.*, 204(11):1620–1662.
- van Ditmarsch, H. (2001). *The Semantics of Concurrent Knowledge Actions*. Workshop on Logic and Games, Ed. M. Pauly and G. Sandu, ESSLLI.
- van Ditmarsch, H., van der Hoek, W., and Kooi, B. (2003). *Concurrent Dynamic Epistemic Logic*. Kluwer, Ed. V.F. Hendricks et al., , vol. 322.
- van Ditmarsch, H., van der Hoek, W., and Kooi, B. (2005). Dynamic epistemic logic with assignment. *AAMAS*, 4(1):141–148.
- van Ditmarsch, H., van der Hoek, W., and Kooi, B. (2008). *Dynamic Epistemic Logic*. Synthese Library Series, volume 337. Springer, The Netherland.