An MDE Approach for Modelling and Reasoning About Multi-agent Systems

Fazle Rabbi[1,2](✉), Yngve Lamo[1], and Lars Michael Kristensen[1]

[1] Western Norway University of Applied Sciences, Bergen, Norway
{Fazle.Rabbi,Yngve.Lamo,Lars.Michael.Kristensen}@hvl.no
[2] University of Oslo, Oslo, Norway

Abstract. Epistemic logic plays an important role in artificial intelligence for reasoning about multi-agent systems. Current approaches for modelling multi-agent systems with epistemic logic use Kripke semantics where the knowledge base of an agent is represented as atomic propositions, but intelligent agents need to be equipped with formulas to derive implicit information. In this paper, we propose a metamodelling approach where agents' state of affairs are separated in different scopes, and the knowledge base of an agent is represented by a propositional logic language restricted to *Horn clauses*. We propose to use a model driven approach for the diagrammatic representation of multi-agent systems knowledge (and nested knowledge). We use a message passing for updating the state of affairs of agents and use belief revision to update the knowledge base of agents.

Keywords: Model-driven engineering · Epistemic logic · Modal logic · Knowledge base

1 Introduction

Our approach for modelling multi-agent system is based on combining metamodelling and epistemic logic with diagrammatic specifications and logic statements. We use the *basic modal system* **K** [5] for epistemic logic and enhance the modelling with the use of model-driven engineering techniques. Traditionally, Kripke structures have been used to give the semantics of epistemic logic by representing the information state of several agents [5]. Although Kripke structures can be used to model the cognitive states of other agents but the knowledge representation of different agents in a Kripke model are not structured. In our approach, agents' information states are modularized into scopes. Scopes include states where the knowledge base of a state is represented by horn clauses making agents capable of deducing new information. We propose to use metamodels for specifying the information state of agents. Using metamodels for defining domain specific modelling languages has potential as languages can be easily customized. However, model-driven engineering using metamodels has not been explored for modelling multi-agent systems.

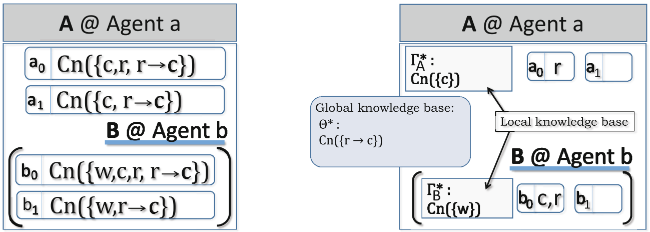
N. Criado Pacheco et al. (Eds.): EUMAS 2016/AT 2016, LNAI 10207, pp. 49–57, 2017.
DOI: 10.1007/978-3-319-59294-7_5

Our work is closely related to the deductive model of belief proposed by Konolige [8] where agents' beliefs are described as a set of sentences in a formal language together with a deductive process for deriving consequences of those beliefs. Deductive model of belief provides a model for agents' problem solving ability based on the reasoning about other agents' problem solving ability. Konolige introduced the concept of *belief subsystem* which can model fairly complicated and confusing situations where agents believe that other agents have belief subsystems of varying capabilities. Some of these scenarios would be useful in representing situations where agents have different beliefs. In real-life, different sources will expose agents to different information, which can naturally lead to disagreement. In our approach, we use several distinct elements to represent and to reason about knowledge in a MAS setting. We use Diagram Predicate Framework [11] for the diagrammatic representation of agents' state of affairs where the knowledge base is represented using Horn clauses. We apply category theory for structuring the knowledge of different agents and combine our approach with Delgrande's inconsistency-based contraction [3] for knowledge base revision. Our proposed approach of modular information states can be used to represent different knowledge of agents, and leads to a simple mechanism to update the knowledge base of agents. We introduce the application of category theoretical operations for extracting the local and global knowledge base of agents which opens up a new formal way of modelling multi-agent systems.

The paper is organized as follows. In Sect. 2, we present a language for representing the knowledge base of agents. Section 3 presents the modelling artifacts for specifying multi-agent systems. Section 4 provides details of how agents communicate information and update their knowledge base, Sect. 5 concludes the paper with a direction for future work.

2 Knowledge Representation of Multi-agent Systems

Unlike Kripke models, states in our multi-agent model are associated with knowledge bases (`KB`). The knowledge base is given by a restricted form of the propositional logic language $\mathcal{L}_{\mathcal{HC}}$ based on a finite set of atoms (atomic propositions) $\mathbf{P} = \{\perp, p, q, r, \ldots\}$, where $\mathbf{P}$ includes the distinguished atom $\perp$ (false). The language $\mathcal{L}_{\mathcal{HC}}$ over $\mathbf{P}$ is given by a set of horn clauses as defined in [3]. A horn clause can be written as a rule in the form $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n \rightarrow \alpha$, where $n \geq 0$, and α, α_i $(1 \leq i \leq n)$ are distinct atoms and $\rightarrow$ represents implication. Let φ be a horn clause in $\mathcal{L}_{\mathcal{HC}}$, if $n = 0$ then φ represents a ground atom and $\rightarrow \alpha$ is written as α. We will use $body(\varphi)$ to refer to the set of atomic propositions to the left of $\rightarrow$ and $head(\varphi)$ to refer to the consequence of φ. We use $\perp$ in the consequence of a clause to represent an *integrity constraint* [3]. In other words, a clause with $\perp$ in the consequence represents an impossible situation. A horn clause φ can be derived from a set of horn clauses Φ, written $\Phi \vdash \varphi$ if φ can be obtained by the inference relation given in [3]. A set of horn clauses Φ is inconsistent if $\Phi \vdash \perp$. We use the notation $\Phi^* = \mathtt{Cn}(\Phi) = \{\varphi : \Phi \vdash \varphi\}$ to represent the set of all logical consequences of Φ. A *scope* of an agent consists of one or



Flat representation of agent knowledge

Hierarchical representation of agent knowledge

Fig. 1. An example model of an agent a and its scopes

more states representing epistemic alternatives. To express what an agent know in its states or about other agents, we need a language of epistemic logic $\mathcal{L}_k$ which is defined as: $\varphi ::= p|\neg\varphi|(\varphi \wedge \varphi)|(\varphi \vee \varphi)|(\varphi \rightarrow \varphi)|K_a\varphi$. Here p is an atomic proposition and K_a is a knowledge operator. For an agent a, $K_a\varphi$ is interpreted as "agent a knows φ".

Figure 1(left) shows an example where the knowledge of an agent a is represented diagrammatically. The example illustrates a scope **A** of agent a consisting of two states a_0 and a_1. These two states are representing two epistemic alternatives of agent a. We use $\texttt{KB}(a_0)$ and $\texttt{KB}(a_1)$ to refer to the knowledge bases of states a_0 and a_1, respectively. In scope **A**, agent a knows that it is cloudy (represented by the ground atom c) and rain implies cloudy (represented by horn clause: $r \rightarrow c$), but agent a does not know if it is raining or not. Therefore, he cannot distinguish between states a_0 and a_1 where $r \in \texttt{KB}(a_0)$ and $r \notin \texttt{KB}(a_1)$ (i.e., uncertain about r). What agent a knows about agent b's epistemic alternatives are represented in the internal scope **B**. The internal scope **B** represents the information state of agent b where agent b knows that it is windy (represented by the ground atom w) but he cannot distinguish between b_0 and b_1 where $c \in \texttt{KB}(b_0), r \in \texttt{KB}(b_0)$ and $c \notin \texttt{KB}(b_1), r \notin \texttt{KB}(b_1)$. Figure 1(right) illustrates this model with hierarchically structured information. Distinguished clauses that are true only in a particular state are represented inside the rectangular box representing the state. Horn clauses that are commonly known among all the agents are represented in a global knowledge base Θ. Horn clauses that are commonly known among all the states in a scope are visualized in a local knowledge base (e.g., Γ_A, Γ_B in Fig. 1). For brevity we will not display all the logical consequences inside the boxes. Note that local knowledge bases implicitly include the global knowledge base. Therefore in Fig. 1(right), $\Gamma_A^* = \texttt{Cn}(\{r \rightarrow c, c\})$ and $\Gamma_B^* = \texttt{Cn}(\{r \rightarrow c, w\})$. We apply a pullback operation for extracting the local and global knowledge bases of agents which is a very common construction in category theory [1]. Below we formalize the notion of local and global knowledge base of agents. In order to apply category theory, we need to constitute a category for *knowledge base* where the objects are sets of horn clauses and morphisms are given by the inclusion mapping of horn clauses.

Definition 1 (Inclusion mapping of horn clauses). *Let Φ and Ψ be two sets of horn clauses. An inclusion mapping $f : \Phi \rightarrow \Psi$ exists if $\forall\varphi \in \Phi$, $\Psi \vdash \varphi$.*

Definition 2 (Local knowledge base). *Let s_0, s_1, $\dots s_n$ be states of agent a in scope A_1 and $\Phi_0 = KB(s_0)$, $\Phi_1 = KB(s_1), \dots$ knowledge bases comprised of horn clauses. The local knowledge base Γ_{A_1} of scope A_1 is the information commonly known by agent a in scope A_1 and is obtained by the limit of the inclusion mappings $\Phi_0 \rightarrow \Phi_{\mathcal{C}}, \Phi_1 \rightarrow \Phi_{\mathcal{C}}, \dots \Phi_m \rightarrow \Phi_{\mathcal{C}}$ where $\Phi_{\mathcal{C}}$ is the combined knowledge base of $\Phi_0, \Phi_1, \dots \Phi_m$.*

Definition 3 (Global knowledge base). *Let $\Gamma_{A_1}, \Gamma_{A_2}, \dots \Gamma_{A_n}$ be the local knowledge bases of scopes $A_1, A_2, \dots A_n$. The global knowledge base Θ is the information commonly known in scopes A_1, A_2, $\dots A_n$ and is obtained by the limit of the inclusion mappings $\Gamma_{A_1} \rightarrow \Gamma_{\mathcal{C}}, \Gamma_{A_2} \rightarrow \Gamma_{\mathcal{C}}, \dots \Gamma_{A_n} \rightarrow \Gamma_{\mathcal{C}}$ where $\Gamma_{\mathcal{C}}$ is the combined knowledge base of all the local knowledge bases.*

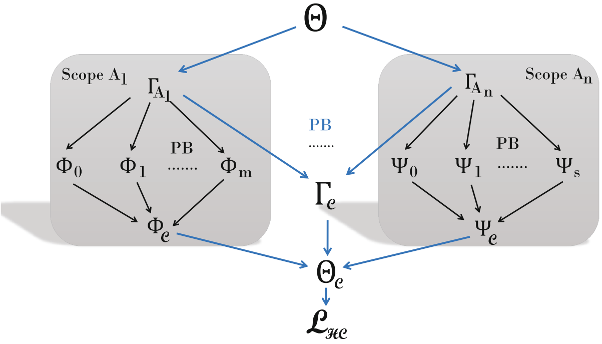


Fig. 2. Local and global knowledge base

Figure 2 shows the knowledge bases of different agents and their local and global knowledge bases. All the arrows in the figure represent inclusion mappings of horn clauses.

Theorem 1. *The limit of a set of inclusion mappings of horn clauses $\{\Phi_0 \rightarrow \Psi, \Phi_1 \rightarrow \Psi, \dots \Phi_m \rightarrow \Psi\}$ is a consistent knowledge base if at least one of $\Phi_0, \Phi_1, \dots \Phi_m$ are consistent.*

Proof: Let Γ be the limit of the inclusion mappings and $\Phi_i (0 \leq i \leq m)$ be a consistent knowledge base. Since $\forall \gamma \in \Gamma$, $\Phi_i \vdash \gamma$, it is not possible that Γ is an inconsistent knowledge base while Φ_i is a consistent knowledge base. □

In Fig. 2 the colimit $\Theta_{\mathcal{C}}$ represents the combined knowledge base of all the agents and the colimit $\Gamma_{\mathcal{C}}$ represents the distributed knowledge of a set of agents. There exists an inclusion mapping from $\Theta_{\mathcal{C}}$ to the language $\mathcal{L}_{\mathcal{HC}}$. It is possible that the colimits might be inconsistent. However, a consistent $\Gamma_{\mathcal{C}}$ supports collaboration of the agents' knowledge.

3 Metamodelling with DPF

We use Diagrammatic Logic [4] and the Diagram Predicate Framework (DPF) [11] for the formal development of metamodel specifications. A metamodel specifies the abstract syntax of a modelling language that often includes a set of modelling concepts, their attributes and their relationships, as well as the rules for combining these concepts to specify valid models. In the context of this paper we develop a metamodel for hierarchical representation of agents' knowledge. In DPF, a (meta)model is represented by a diagrammatic specification $\mathfrak{S} = (S, C^{\mathfrak{S}} : \Sigma)$ consisting of an underlying graph S together with a set of *atomic constraints* $C^{\mathfrak{S}}$ specified by a *predicate signature* Σ. A predicate is used to specify constraints in a model by means of graph homomorphisms. DPF provides a formalization of multi-level metamodelling by defining the conformance relation between models at adjacent levels of a metamodelling hierarchy [10].

The graph in Fig. 3(b) represents the specification of a multi-agent model $\mathfrak{S}_1 = (S_1, C^{\mathfrak{S}_1} : \Sigma)$. Constraints are added into the structure by predicates. Figure 3(a) shows the predicates used for constraining the model $\mathfrak{S}_1$. Each predicate has a name p, a shape graph (arity) α(p), a visualization, and a semantic interpretation. For instance, the intended semantics of *<mult(n,m)>* is that for each instance of X, f must have at least n and at most m instances. The predicates are constraining the model $\mathfrak{S}_1$ by means of a graph homomorphism $\delta : \alpha(p) \rightarrow S_1$ from the arity of the predicate p to the graph of the model $\mathfrak{S}_1$. The model $\mathfrak{S}_1$ specifies that an agent may have a scope consisting of a number of states. An agent's scope may have internal scopes of other agents. An instance (I, ι) of $\mathfrak{S}_1$ is shown (represented in abstract syntax) in Fig. 3(c). The instance (I, ι) of $\mathfrak{S}_1$ is given by a graph I together with a typing graph homomorphism $\iota : I \rightarrow S_1$ that satisfies the constraints $C^{\mathfrak{S}_1}$. The diagram shown earlier in Fig. 1 is the concrete syntax of this instance. The semantics of the predicates are provided in a fibred manner [4]. That is, the semantics of a predicate p is given by the set of its instances. The multiplicity predicate *<mult(n,m)>* is used to add an atomic constraint on edge '`contains`' in Fig. 3. This atomic constraint specifies that every `Scope` instance must contain at least one `State` instance. The irreflexive predicate *<irreflexive>* is used to add an atomic constraint on edge '`internal`'. The atomic constraint specifies that a `Scope` instance cannot have reflexive reference of type '`internal`'.

We use the concept of a single 'State' in the multi-agent model to represent the condition of an agent and use the word 'information state' to represent an instance of a multi-agent model. An instance of an agent's state is valid if the associated knowledge base is consistent. Let (M, ι) be a multi-agent model instance consisting of a set of agents $\mathcal{A}$ (instances of `Agent`) and $\Phi = \texttt{KB}(s_a)$ be a set of horn clauses representing the knowledge base of a state s_a in scope $\mathbf{A}$ of an agent $a \in \mathcal{A}$. The state s_a is consistent or satisfiable if $\Phi \nvdash \bot$. We define that a propositional logic formula φ is true in s_a, written as $(M, \iota), s_a \models \varphi$, as follows:

$(M, \iota), s_a \models p$ iff $p \in \Phi^*$ (Φ^* is the set of all logical consequences of Φ)
$(M, \iota), s_a \models \neg p$ iff $p \notin \Phi^*$

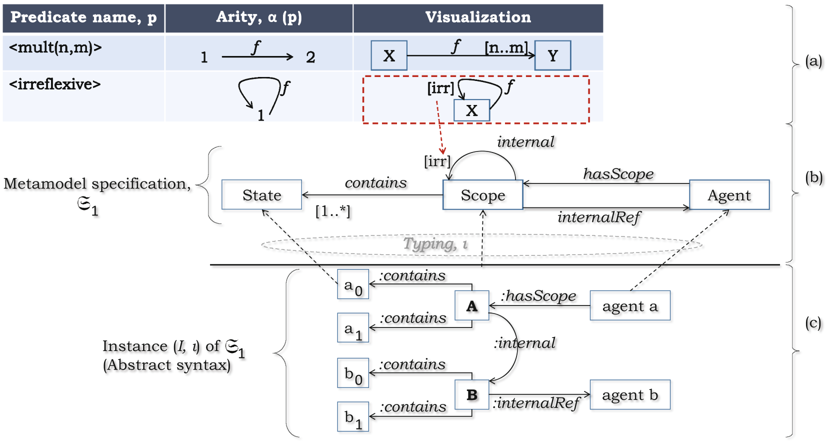


Fig. 3. A DPF specification $\mathfrak{S}_1$ for multi-agent model

$(M, \iota), s_a \models (\varphi \wedge \psi)$ iff $(M, \iota), s \models \varphi$ and $(M, \iota), s \models \psi$
$(M, \iota), s_a \models (\varphi \vee \psi)$ iff $(M, \iota), s \models \varphi$ or $(M, \iota), s \models \psi$

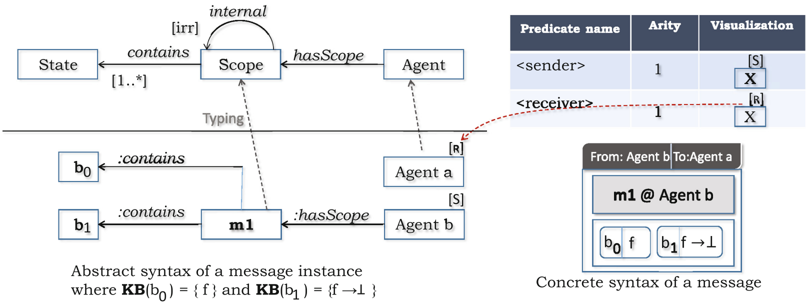
Let us consider that a scope **A** contains a set of states, **S**. Any formula φ generated by the language of epistemic logic $\mathcal{L}_k$ is true in scope **A** (written as $(M, \iota), \mathbf{A} \models \varphi$) or in a state $s_a \in \mathbf{S}$ (also written as $(M, \iota), s_a \models \varphi$) as defined below:

$(M, \iota),\ \mathbf{A} \models p$ iff $\forall s \in \mathbf{S}\ (M, \iota), s \models p$
$(M, \iota),\ \mathbf{A} \models \neg p$ iff $\exists s \in \mathbf{S}\ (M, \iota), s \not\models p$
$(M, \iota),\ \mathbf{A} \models (\varphi \wedge \psi)$ iff $\forall s \in \mathbf{S}\ (M, \iota), s \models \varphi$ and $(M, \iota), s \models \psi$
$(M, \iota),\ \mathbf{A} \models (\varphi \vee \psi)$ iff $\forall s \in \mathbf{S}\ (M, \iota), s \models \varphi$ or $(M, \iota), s \models \psi$
$(M, \iota),\ \mathbf{A} \models K_a\varphi$ iff $\forall s \in \mathbf{S}\ (M, \iota), s \models \varphi$
$(M, \iota),\ \mathbf{A} \models K_b\varphi\ (b \in \mathcal{A}\ \wedge\ b \neq a)$ iff $(M, \iota),\ \mathbf{B} \models \varphi$ where agent b's scope **B** is an internal scope of **A**
$(M, \iota), s_a \models K_a\varphi$ iff $(M, \iota),\ \mathbf{A} \models K_a\varphi$
$(M, \iota), s_a \models K_b\varphi\ (b \in \mathcal{A}\ \wedge\ b \neq a)$ iff $(M, \iota),\ \mathbf{A} \models K_b\varphi$

An instance of a multi-agent model (i.e., an information state of an agent) is valid if it satisfies all the domain constraints specified in the DPF specification and contains only valid states for agents.

4 Message Passing Communication

We envision a system where agents collaborate with each other by exchanging messages which include epistemic information. These messages are used to update the knowledge bases of agents and their information states. The agents have their own knowledge base, and in addition they are aware of other agents' knowledge bases.



Predicate name	Arity	Visualization
<sender>	1	[S] X
<receiver>	1	[R] X

Fig. 4. Predicates for annotating message instance

A message is an instance of the multi-agent model where the agents are annotated with $[S]$ and $[R]$ ($<$*sender*$>$ and $<$*receiver*$>$ predicates). Figure 4 shows the predicates and the abstract and concrete syntax of a message. A message contains a scope of an agent with a set of states. The states have an associated knowledge base which contains a set of propositional horn clauses. An incoming message from the sender agent is used to update the internal scope of the receiver agent. Two kinds of update operations are performed in order to update the internal scope of the receiver agent: (i) product of states (in the category of sets), and (ii) knowledge base revision. The product operation deals with the higher order information and the revision operation updates the knowledge base of states. Figure 5 illustrates an agent b sending a message to agent a. Agent b informs agent a that he does not know if it is foggy (represented by f) or not. The figure shows the effect of an update operation where a product is formed to update the information states of the internal scope **B**. After performing the product operation, the knowledge base of the states are revised based on the knowledge base from the message.

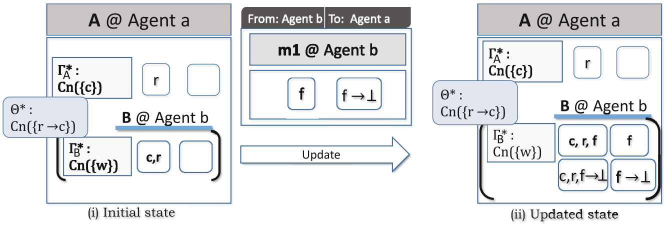


Fig. 5. Example effect of an update

We consider Delgrande's inconsistency based contraction [3] for knowledge base revision. The purpose of this type of revision is to modify the knowledge base in such a way that adding new horn clauses from the message does not result in an inconsistent knowledge base. While modifying the knowledge base we want to retain as much as possible from the old knowledge base. We use Delgrande's definition of i-reminder set for the horn language:

Definition 4 (Horn i-Reminder Sets). *Given a knowledge base Φ in $\mathcal{L}_{\mathcal{HC}}$ and a set of new horn clauses Ψ, Horn i-reminder sets of Φ w.r.t. Ψ, written $\Phi \downarrow_i \Psi$ is the set such that $K \in \Phi \downarrow_i \Psi$ iff (i) $K \subseteq \Phi$, (ii) $K \cup \Psi \nvdash \bot$, (iii) $\nexists K'$ such that $K \subset K' \subseteq \Phi$, $K' \cup \Psi \nvdash \bot$.*

While updating the knowledge base Φ of a state s of an agent a due to the new information Ψ of a message, we propose to use Horn i-reminder sets. If there are more than one element in $\Phi \downarrow_i \Psi$, multiple states are produced by replacing s containing different possibilities of revised knowledge base. However other strategies may be followed to revise Φ such as Horn i-Contraction [2].

5 Conclusion and Future Work

Epistemic logic was first introduced by Hintikka in [7] and later on used by numerous researchers for modelling multi-agent systems where the information state of multi-agent systems are given by Kripke semantics [5,6]. One issue with this approach is that models become very big in size as the number of epistemic alternatives increases. In this paper, we presented a model driven approach where the states are modularized in scopes which clearly represents agents dimension of epistemic alternatives. To extract the local and global knowledge base of agents, we use pullback, limit and colimit operations.

In future, we will investigate reasoning algorithms to rule out uncertainty. Reasoning about uncertainty may play an important role in optimizing resources via strategies. In [9], we introduced a categorical approach for metamodelling epistemic game theory. As part of the future work, we will investigate how game theoretic concepts can be applied in a multi–agent system environment using model driven engineering approaches.

References

1. Barr, M., Wells, C. (eds.): Category Theory for Computing Science, 2nd edn. Prentice Hall International (UK) Ltd., Upper Saddle River (1995)
2. Booth, R., Meyer, T., Varzinczak, I.J.: Next steps in propositional horn contraction. In: Boutilier, C. (ed.) Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009), pp. 702–707 (2009)
3. Delgrande, J.P.: Horn clause belief change: contraction functions. In: Brewka, G., Lang, J. (eds.) Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning, KR 2008, pp. 156–165. AAAI Press (2008)
4. Diskin, Z., Wolter, U.: A diagrammatic logic for object-oriented visual modeling. Electron. Notes Theoret. Comput. Sci. **203**(6), 19–41 (2008). Proceedings of the Second Workshop on Applied and Computational Category Theory (ACCAT 2007)
5. Van Ditmarsch, H., van Der Hoek, W., Kooi, B.: Dynamic Epistemic Logic, 1st edn. Springer, Heidelberg (2007)

6. Gerbrandy, J.: Dynamic epistemic logic. In: Logic, Language and Computation, Stanford, CA, USA, vol. 2, pp. 67–84. Center for the Study of Language and Information (1999)
7. Hintikka, J.: Knowledge and Belief: An Introduction to the Logic of the Two Notions. Texts in Philosophy, King's College Publications (2005)
8. Konolige, K.: A deductive model of belief. In: Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 1, IJCAI 1983, San Francisco, CA, USA, pp. 377–381. Morgan Kaufmann Publishers Inc. (1983)
9. Rabbi, F., Lamo, Y., Yu, I.C.: Towards a categorical approach for meta-modelling epistemic game theory. In: Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, MODELS 2016, pp. 57–64. ACM. New York (2016)
10. Rabbi, F., Lamo, Y., Yu, I.C., Kristensen, L.M.: WebDPF: a web-based metamodelling and model transformation environment. In: MODELSWARD 2016 - Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development, Rome, Italy, 19–21 February 2016, pp. 87–98. SciTePress (2016)
11. Rutle, A.,: Diagram predicate framework: a formal approach to MDE. Ph.D. thesis, Department of Informatics, University of Bergen, Norway (2010)