On Representing Actions in Multi-Agent Domains

Chitta Baral and Gregory Gelfond

Department of Computer Science and Engineering
Arizona State University

Abstract. Reasoning about actions forms the foundation of prediction, planning, explanation, and diagnosis in a dynamic environment. Most of the research in this field has focused on domains with a single agent, albeit in a dynamic environment, with considerably less attention being paid to multi-agent domains. In a domain with multiple agents, interesting issues arise when one considers the knowledge of various agents about the world, as well about as each other's knowledge. This aspect of multi-agent domains has been studied in the field of dynamic epistemic logic. In this paper we review work by Baltag and Moss on multi-agent reasoning in the context of dynamic epistemic logic, extrapolate their work to the case where agents in a domain are classified into three types and suggest directions for combining ideas from dynamic epistemic logic and reasoning about actions and change in order to obtain a unified theory of multi-agent actions.

1 Introduction

Actions, when executed, often change the state of the world. Reasoning about actions enables us to predict whether a given sequence of actions is indeed going to achieve some goal; it allows us to plan, or obtain a sequence of actions that would achieve a particular goal; it grants us the ability to explain observations in terms of actions that may have taken place; and it allows us to diagnose faults in a system by determining those actions whose consequences may have lead to them. When actions have non-deterministic effects, higher level reasoning about them is needed to verify and construct policies, enabling us to maintain various properties in addition to achieving certain goals.

As the number of states within a domain is often exponential in terms of the number of fluents (individual properties of the world), a central aspect in reasoning about actions is to develop languages which enable the concise representation of actions and their effects, and whose semantics define transitions between "states" due to their execution. In a single-agent domain, if we assume that an agent has complete knowledge of the values of the fluents, states may be thought of as *states of the world*. In the presence of incompleteness and *sensing actions*, states may be thought of as pairs which combine states of the world, together with an agent's *knowledge state*.

Considerable research has been done on the development of a class of languages, called *action languages*, that allow one to describe the world and the effects of various actions, and in using them for various tasks such as prediction, planning, counterfactual reasoning, as well as in proving the correctness of plans, policies and execution programs. The importance of reasoning about actions in the context of AI was mentioned as early as 1969 by McCarthy [10]. A systematic approach has evolved over the past twenty years which has been guided and influenced by the high level action language approach (beginning with the language $\mathcal{A}$ [7]), the approach of Sandewall [12], and the approach of the Toronto school [9,11].

With few exceptions, [13,8,6], the majority of these approaches have assumed that actions were performed by a single agent. Even in instances where multiple agents have been referred to, their interactions have been kept simple [3] (e.g., two agents simultaneously lifting a large table). In the real world however, interactions between multiple agents are more involved. One key issue that presents itself in multi-agent domains is the potential for discrepancies between the beliefs of various agents due to their different *frames of reference*.

While this issue has not been studied thoroughly within the reasoning about actions community, it has been explored in a somewhat different setting by the dynamic epistemic logic community [4,1]. In this paper we review work by Baltag and Moss [1] and discuss how some of its ideas can be applied to the reasoning about actions setting. We then pose some questions that need to be addressed in order to perform various reasoning tasks in a multi-agent domain.

The rest of the paper is structured as follows. In Sect. 2, we give a bit of background on modal logic and Kripke models. In Sect. 3, we introduce Baltag and Moss's action models and present a few examples (taken from the literature) of their use for modeling multi-agent actions. In Sect. 4 we show how Baltag and Moss's action models can be used to express multi-agent actions involving three classes of agents. In Sect. 5 we discuss some of our concerns with this approach and suggest some ways to overcome them and present an avenue for developing a high-level language to represent and reason about actions in a multi-agent setting. We then conclude with some final thoughts.

2 Background: Modal Logic and Kripke Models

We begin our discussion with an overview of modal logic, which forms the foundation of many of the concepts discussed in this work.

Definition 1 (Multi-Agent Domain). *A* multi-agent domain *is defined as a triple, $(\mathcal{AG}, \mathcal{F}, \mathcal{A})$, where $\mathcal{AG}$ is a finite, non-empty set of agent names, $\mathcal{F}$ is a finite, non-empty set of propositional atoms, and $\mathcal{A}$ is a finite, non-empty set of action names.*

Intuitively, $\mathcal{AG}$ defines the set of agents who are operating in the domain, $\mathcal{F}$ defines the physical properties of the domain, and $\mathcal{A}$ denotes the actions which the agents are capable of performing. Given a multi-agent domain, various properties of the domain are described by modal formulae.

Definition 2 (Modal Formula). *Given a multi-agent domain,* $(\mathcal{AG}, \mathcal{F}, \mathcal{A})$*, a* modal formula *is defined as follows:*

- *if* $\phi \in \mathcal{F}$*, then* ϕ *is modal formula*
- *if* ϕ *is a modal formula, then* $\neg\phi$ *is a modal formula*
- *if* ϕ *is a modal formula, then* $\mathbf{K}_i\phi$*, where* $i \in \mathcal{AG}$*, is a modal formula*
- *if* ϕ *is a modal formula, then* $\mathbf{E}_\alpha\phi$*, where* $\alpha \subseteq \mathcal{AG}$*, is a modal formula*
- *if* ϕ *is a modal formula, then* $\mathbf{C}_\alpha\phi$*, where* $\alpha \subseteq \mathcal{AG}$*, is a modal formula*
- *if* ϕ *and* ψ *are modal formulae, then* $\phi \wedge \psi$*,* $\phi \vee \psi$*,* $\phi \Rightarrow \psi$*, and* $\phi \Leftrightarrow \psi$ *are modal formulae*

Intuitively, a modal formula of the form $\mathbf{K}_i\phi$ means that "agent i knows ϕ". Modal formulae of the form $\mathbf{E}_\alpha\phi$, mean that "every agent in the group α knows ϕ." Lastly, modal formulae of the form $\mathbf{C}_\alpha\phi$, mean that "ϕ is common knowledge amongst the agents in α."[1] Before defining the entailment relation, it is necessary to define the notions of a *Kripke structure*, and a *Kripke world*.

Definition 3 (Kripke Structure). *A* Kripke structure*,* M*, over a set of agents* $\mathcal{AG}$*, and a set of fluents* $\mathcal{F}$*, is defined as a tuple* $(S, \pi, \{R_i \mid i \in \mathcal{AG}\})$ *where:*

- S *is a non-empty set of state symbols (or possible worlds)*
- π *is a function mapping elements of* S *onto interpretations of* $\mathcal{F}$
- *each* R_i *is a binary relation on* S *(called an accessibility relation)*

Intuitively, a pair $(\sigma, \tau) \in R_i$ is understood to mean that from within possible world σ, agent i cannot distinguish between σ and τ. Depending on the modality of discourse, the accessibility relations have different properties, which correspond to axiom systems associated with that modality. For example, the axiom system S5 is often used with the *knowledge* modality, in which case the accessibility relations must be *reflexive*, *symmetric*, and *transitive*. On the other hand, the axiom system KD45 is often used with the *belief* modality, in which case the accessibility relations are *euclidean*, *serial*, and *transitive*.

Definition 4 (Kripke World). *A* Kripke world *is defined as a pair* (M, σ)*, where* M *is a Kripke structure, and* σ *is a state symbol of* M*.* σ *is said to be the state which designates* the real physical state of the world.

Having defined the notions of a Kripke structure and a Kripke world, we can define the semantics of modal logic.

Definition 5 (Entailment Relation for Modal Formulae). *Let* ϕ *and* ψ *be modal formulae, and let* (M, σ) *be a Kripke world over a multi-agent domain* $D = (\mathcal{AG}, \mathcal{F}, \mathcal{A})$*. The* entailment relation for modal formulae *is defined as follows:*

- *if* $\phi \in \mathcal{F}$*, then* $(M, \sigma) \models \phi$ *if and only if* $\pi(\sigma)(\phi) = \top$
- $(M, \sigma) \models \neg\phi$ *if and only if* $(M, \sigma) \not\models \phi$

[1] The readings of the various modal formulae use the term *knows* in order to remain consistent with prior work done in the field of dynamic epistemic logic.

- $(M,\sigma) \models \phi \wedge \psi$ *if and only if* $(M,\sigma) \models \phi$ *and* $(M,\sigma) \models \psi$
- $(M,\sigma) \models \phi \vee \psi$ *if and only if* $(M,\sigma) \models \phi$ *or* $(M,\sigma) \models \psi$
- $(M,\sigma) \models \phi \Rightarrow \psi$ *if and only if* $(M,\sigma) \models \neg\phi \vee \psi$
- $(M,\sigma) \models \phi \Leftrightarrow \psi$ *if and only if* $(M,\sigma) \models (\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$
- $(M,\sigma) \models \mathbf{K}_i\phi$ *if and only if* $(M,\tau) \models \phi$ *for all* τ *such that* $(\sigma,\tau) \in R_i$
- $(M,\sigma) \models \mathbf{E}_\alpha\phi$ *if and only if* $(M,\sigma) \models \mathbf{K}_i\phi$ *for every agent* $i \in \alpha$

Let $\mathbf{E}^0_\alpha\phi$ *be an abbreviation for* ϕ*, and let* $\mathbf{E}^{k+1}_\alpha\phi = \mathbf{E}^k_\alpha\mathbf{E}_\alpha\phi$*.*

- $(M,\sigma) \models \mathbf{C}_\alpha\phi$ *if and only if* $(M,\sigma) \models \mathbf{E}^k_\alpha\phi$ *for* $k = 0,1,2,\ldots$

Example 1 (The Strongbox Domain of [1]). Consider a domain in which two agents, A and B, are together in a room containing a strongbox in which there is a coin. This fact is common knowledge amongst the agents, as is the fact that none of them knows which face of the coin is showing. Suppose that the coin is actually lying heads up. The Kripke world shown in Fig. 1 represents the initial configuration of the world and the agents' knowledge about the world. In the figure, circles represent states and a double circle represents the real physical state of the world. The Kripke world in Fig. 1 entails the following modal formulae (among others):

- $(M,\sigma) \models H$
- $(M,\sigma) \models \neg\mathbf{K}_A H \wedge \neg\mathbf{K}_A\neg H$
- $(M,\sigma) \models \neg\mathbf{K}_B H \wedge \neg\mathbf{K}_B\neg H$
- $(M,\sigma) \models \mathbf{C}_{\{A,B\}}(\neg\mathbf{K}_A H \wedge \neg\mathbf{K}_A\neg H \wedge \neg\mathbf{K}_B H \wedge \neg\mathbf{K}_B\neg H)$

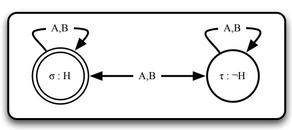

Fig. 1. Kripke world for the Strongbox Domain. Neither A nor B know the real world; the external observer knows that H is true in the real world.

3 Baltag-Moss Action Models

In a multi-agent setting, the agents in the domain may have differing frames of reference with respect to an action. Continuing with the Strongbox Domain, agent A may peek into the box to find that the coin is showing heads with agent B watching this take place. Baltag and Moss [1] describe a construct called an *action model* which they use to represent these differences of perspective. Moreover, they define a means by which an action model may be used to *update* a Kripke world in order to obtain a successor world modeling the effects of the execution of the underlying action.

Intuitively, an action model may be thought of as a directed graph, whose nodes are referred to as simple actions, and are labeled by modal formulae representing their preconditions. The edges of an action model are labeled by the names of the agents in the domain, and are meant to convey the frames of reference of the agents with respect to the complex action described by the model.

Definition 6 (Action Model). *An* action model*,* $\mathbf{\Sigma}$*, over a set of agents* $\mathcal{AG}$*, is defined as a tuple* $(\Sigma, s, \{R_i \mid i \in \mathcal{AG}\}, pre)$

- Σ *is a set whose elements are called simple actions*
- $s \subseteq \Sigma$ *is a set of designated simple actions*
- *each* R_i *is a binary relation on* Σ
- *pre is a function mapping each simple action* $\alpha \in \Sigma$ *onto a modal formula representing its preconditions*

Action models correspond to individual occurrences of a complex, knowledge producing action[2], and induce an *update operation* on the Kripke worlds describing the *state of the world* prior to its execution. Within this paper they are represented as directed graphs, whose nodes correspond to the set of simple actions (and are labeled by the modal formulae describing their preconditions), and whose arcs are defined by the binary relations over Σ. Prior to defining the update operation, we introduce the notion of a *knowledge state.*

Definition 7 (Knowledge State). *A* knowledge state *in a multi-agent domain is defined as a set of Kripke worlds that have the same underlying Kripke structure and differ only in the state representing the physical state of the world. The different physical state of the world are from the frame of reference of an external observer.*

Example 2 (Knowledge State in the Strongbox Domain). Consider a domain in which two agents, A and B, are together in a room containing a strongbox in which there is a fair coin. This fact is common knowledge amongst the agents, as is the fact that none of them knows which face of the coin is showing. The graph shown in Fig. 2 shows the knowledge state of the agents A and B. Notice

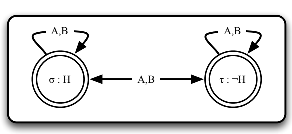

Fig. 2. A knowledge state in the Strongbox Domain; neither A, nor B and nor the external observer know the real world.

[2] Baltag and Moss limit their discussion in [1] to actions which affect the knowledge of the agents, as opposed to the physical state of the domain, hence the term *knowledge producing action.*

that from our frame of reference as external observers, we ourselves do not know which state corresponds to the real physical state of the world, and consequently both σ and τ are marked as possibilities.

For notational convenience, given a knowledge state $\mathbf{S}$, let $\mathbf{S}[S]$ denote the set of state symbols in $\mathbf{S}$; $\mathbf{S}[s]$ denote the set of state symbols in $\mathbf{S}$ describing the possible real physical states of the world; $\mathbf{S}[\pi]$ denote the interpretation function in $\mathbf{S}$; and $\mathbf{S}[R_i]$ denote the accessibility relation for agent i in $\mathbf{S}$. Similarly, given an action model $\mathbf{\Sigma}$, let $\mathbf{\Sigma}[S]$ denote the set of action symbols in $\mathbf{\Sigma}$; $\mathbf{\Sigma}[s]$ denote the set of action symbols in $\mathbf{\Sigma}$ describing the designated simple actions; and $\mathbf{\Sigma}[R_i]$ denote the relation for agent i in $\mathbf{\Sigma}$.

Definition 8 (Update Operation Induced by an Action Model). *Let* $\mathbf{S}$ *denote a knowledge state within a multi-agent domain, and let* $\mathbf{\Sigma}$ *be an action model for a knowledge producing action. The* update operation*, induced by* $\mathbf{\Sigma}$*, is defined as a new knowledge state,* $O(\mathbf{S}) = \mathbf{S} \otimes \mathbf{\Sigma}$ *where:*

- $O(\mathbf{S})[S] = \{(\sigma, \tau) : \sigma \in \mathbf{S}[S], \tau \in \mathbf{\Sigma}[S]$ *and* $(M, \sigma) \models pre(\tau)\}$*, where* M *is the underlying Kripke structure of* $\mathbf{S}$
- $((\sigma, \tau), (\sigma', \tau')) \in O(\mathbf{S})[R_i]$ *if and only if* $(\sigma, \sigma') \in \mathbf{S}[R_i]$ *and* $(\tau, \tau') \in \mathbf{\Sigma}[R_i]$
- $(\sigma, \tau) \in O(\mathbf{S})[s]$ *if and only if* $(\sigma, \tau) \in O(\mathbf{S})[S]$ *and* $\sigma \in \mathbf{S}[s]$*,* $\tau \in \mathbf{\Sigma}[s]$
- $O[\pi]((\sigma, \tau)) = \mathbf{S}(\pi)(\sigma)$

We now present several action models from [1] and their impact on the knowledge state of the Strongbox Domain shown in Fig. 1. Throughout this paper the examples of successor knowledge states obtained as a consequence of the update operation have been simplified (for example by removing states from $O(\mathbf{S})[S]$, along with their incoming and outgoing arcs, which are not reachable from those in $O(\mathbf{S})[s]$). Due to space considerations a full description of these simplification actions has been omitted.

3.1 Global Announcement Actions

The first class of an action scenario that we examine is that of *global announcements*, actions in which a new piece of information is made common knowledge amongst the agents of the domain[3].

Example 3 (Global Announcement Actions). Consider the variant of the Strongbox Domain mentioned in Ex. 2. Suppose that it is announced to both A and B that the coin is showing heads. The corresponding action model, $\mathbf{\Sigma_1}$, is shown in Fig. 3. Note that the intuitive reading of the model is that the action causes H to be common knowledge to both A and B. Furthermore, the real state of the world is the one in which H is true. Application of the update operation yields the transition shown in Fig. 4.

The action model from Fig. 3 may be generalized to the global announcement of an arbitrary modal formula as shown in Fig. 5.

[3] It is assumed that global announcements only convey factually correct information.

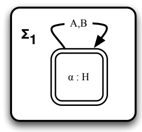

Fig. 3. Baltag-Moss action model for the global announcement of H to A and B.

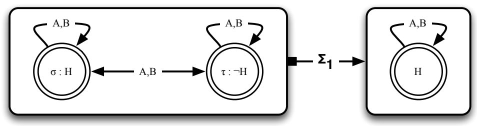

Fig. 4. Transition defined by $\mathbf{\Sigma_1}$, the global announcement that H is true.

3.2 Private Announcement Actions

In the previous subsection we considered announcement actions that were made globally. In private announcements the announcement is only made to a selected group of agents. The rest of the agents are oblivious about the announcement.

Example 4 (Private Announcement Actions). As before, we begin by considering the variant of the Strongbox Domain mentioned in Ex. 2. Suppose that agent A peeks into the strongbox, learning that the coin is showing heads, and that agent B is unaware of this. The corresponding action model, $\mathbf{\Sigma_3}$, is shown in Fig. 6. The intuitive reading of this model is that agent A knows H to be true in the real physical state of the world, and also correctly believes that the knowledge of agent B is unchanged (represented by the state labeled $\top$). The transition defined by application of the update operation is shown in Fig. 7.

The action model in Fig. 6 may be viewed as a special case of an action in which differing pieces of information are declared to the agents. Specifically, the model may be interpreted as the composition of two simple actions, the first of which announces a formula, $\varphi = H$ to agent A, while the second essentially models an announcement by agent A that nothing has changed to agent B (represented by the formula $\psi = \top$). With this reading in mind, the action model may be generalized in a fashion as shown in Fig. 8.

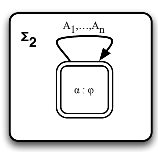

Fig. 5. Global announcement of a modal formula φ to agents $\{A_1, \ldots, A_n\}$

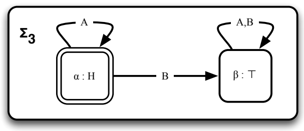

Fig. 6. Action model for agent A observing H unbeknownst to agent B.

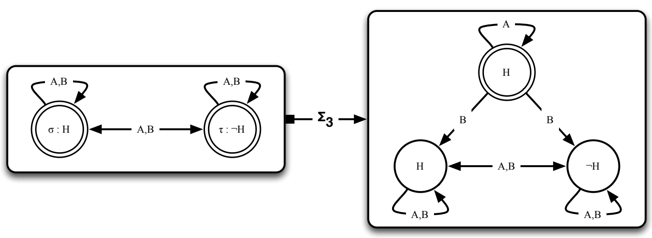

Fig. 7. Transition defined by $\mathbf{\Sigma_3}$.

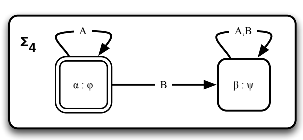

Fig. 8. Action model, $\mathbf{\Sigma_4}$: Agent A learns φ, but agent B believes that ψ was announced to A instead and A is aware of B's mistaken belief.

Notice that in both Fig. 6 and Fig. 8, the *external observer* knows the actual state of the world (in the former, we are directly told that agent A observes H, in the later this is extended to the formula φ).

3.3 Sensing Actions

In addition to acquiring new information about the domain through announcements, agents may perform *sensing actions* in order to learn more about their surroundings. The effect of sensing actions is not known in advance, consequently, the external observers do not a-priori know the result.

Example 5 (Sensing Actions). As before, consider the variant of the Strongbox Domain in which we have n agents, $A_1, \ldots, A_n$. Now suppose that agents $A_1, \ldots, A_m$ peek into the box together, thereby learning which side of the coin is facing up while agents $A_{m+1}, \ldots, A_n$ remain unaware of what has transpired. The corresponding action model $\mathbf{\Sigma_5}$ is shown in Fig. 9.

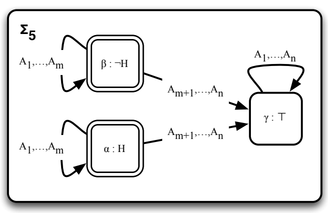

Fig. 9. Agents $A_1, \ldots, A_m$ secretly sense value of H; agents $A_{m+1}, \ldots, A_n$ are oblivious.

All of the previous examples of sensing actions have assumed that agents within a domain could be partitioned into two categories: *actors*, and *oblivious agents*. Baltag and Moss also consider a different class of non-actors, who are not directly sensing but are (partially) observing the occurrence of the sensing actions. We call such agents *partial-observers* as they do not know the result of such an action, but do know that it occurred.

Example 6 (Sensing Actions with Partial Observers). Consider the variant of the Strongbox Domain from Ex. 5. Suppose however that agents $A_1, \ldots, A_m$ peek into the box together with the knowledge of agents $A_{m+1}, \ldots, A_n$ (e.g., agents $A_{m+1}, \ldots, A_n$ are watching this take place). The corresponding action model $\mathbf{\Sigma_6}$ is shown in Fig. 10.

In addition to the deterministic sensing actions discussed before, there are also *nondeterministic sensing actions*. Such actions represent the notion of an agent possibly performing a sensing action.

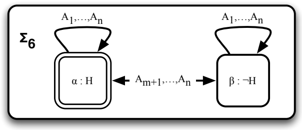

Fig. 10. Agents $A_1, \ldots, A_m$ sense H to be true; agents $A_{m+1}, \ldots, A_n$ are aware of H being sensed but not the value that was sensed.

Example 7 (Nondeterministic Sensing Actions). Consider the Strongbox Domain from Ex. 5. Suppose that agents $A_1, \ldots, A_m$ may have peeked into the box, *possibly determining* the face of the coin, and that the remaining agents $A_{m+1}, \ldots, A_n$, are aware of this possibility. The corresponding action model $\mathbf{\Sigma_7}$ is shown in Fig. 11.

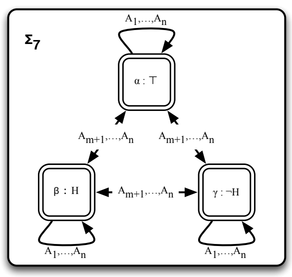

Fig. 11. Agents $A_1, \ldots, A_m$ may have peeked into the box; agents $A_{m+1}, \ldots, A_n$, are aware of this possibility

A further variant of sensing deals with the case where one agent believes that some other agent may have sensed the value of a fluent. The actions models representing two instances of this scenario are given in Fig. 12 and Fig. 13. Fig. 12 deals with the case where the mistaken belief of one agent is later proven to be true, while Fig. 13 deals with the case in which one agent has lied to another. Both scenarios are modeled as the sequential composition of two separate actions, the first of which (given by $\mathbf{\Sigma_8}$ and $\mathbf{\Sigma_{10}}$ respectively) is common to both of them. This action in essence *sets the stage* for its successor, by allowing the possibility for one agent to have potentially erroneous information about the domain.

4 Using Baltag-Moss Action Models to Express Three Classes of Agents

The various action models in the previous section are from [1] and other works. They propose a notion of action signatures generalizing those action models.

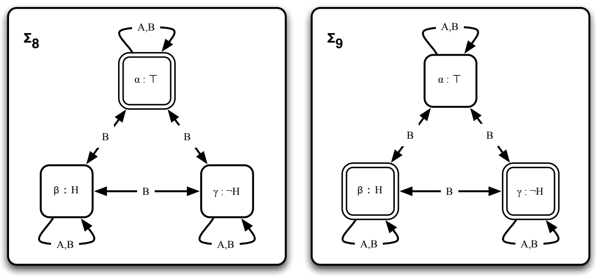

Fig. 12. $\mathbf{\Sigma_8}$: Agent B mistakenly believes that agent A has peeked into the box. $\mathbf{\Sigma_9}$: Agent A has indeed peeked and agent B correctly believes this to be the case.

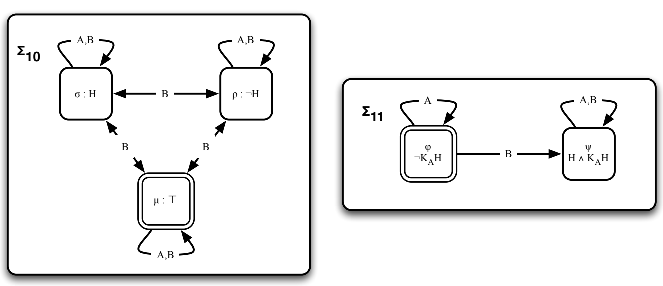

Fig. 13. $\mathbf{\Sigma_{10}}$: Agent B mistakenly believes that agent A may have peeked. $\mathbf{\Sigma_{11}}$: Agent A does not know the value of H but successfully lies to B that it knows and the value is H.

They also propose a notion of program models built using action models and action signatures. In this section we do a generalization of their actions in a different dimension.

Broadly speaking, the action models described by Baltag and Moss separate the agents into two groups. Each action defines a set of *actors*, representing the set of agents who are "recipients" of the action's direct effects. In addition, an action either defines a set of agents who are *observers*, i.e., agents who are aware that some action has occurred, and as a consequence receive the action's indirect effects; or who are *oblivious*, i.e., are completely unaware that the action has transpired, and whose knowledge is therefore unchanged.

It is often the case however that two classes of agents are not enough to fully express the consequences of a knowledge producing action. Consider for example a variant of the Strongbox Domain consisting of three agents, A, B, and C. In this case it is quite natural to envision a scenario in which agent A performs some action, with agent B observing, and agent C being completely oblivious as to what has transpired.

4.1 Extending Announcement Actions

The global announcements described by Baltag and Moss may be thought of as a special case of an *announcement*, where all of the agents within the domain are actors. Suppose however, that only a specific subset of the agents receives the announcement. In this case the action is more complex, with differing effects depending upon how the agents are grouped with respect to their *awareness of the action occurrence.*

Example 8 (Announcements with Partial Observers). Consider a variant of the Strongbox Domain comprised of three agents, A, B, and C, and in which the initial knowledge state is unchanged. Suppose that it is announced to agent A that the coin is showing heads. Furthermore, let us suppose that agent B is aware that some piece of information concerning the coin has been announced to agent A, and that agent C is oblivious to what has transpired. The action model, $\mathbf{\Sigma_{12}}$, is shown in Fig. 14. Applying the update yields the transition shown in Fig. 15. Note that in the resulting Kripke world, agent A knows that H is true; agent B knows that agent A knows the value of H but does know what the value is; however, as far as agent C is concerned, he believes that none of the agents know the value of H.

4.2 Sensing Actions With Three Classes of Agents

As with communication actions, sensing actions also need to be extended to the case where all three classes of agents are present.

Example 9 (Extended Sensing Actions). Consider the the Strongbox Domain in which we have three agents, A, B, and C. Suppose that agent A performs a *sensing action* (as originally introduced in Ex. 5), with agent B observing the

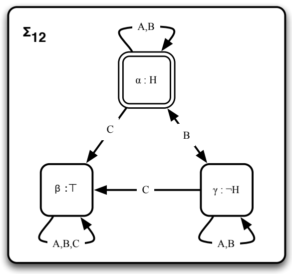

Fig. 14. Action model for the announcement of H to A with agent B as an observer from far, and an oblivious agent C.

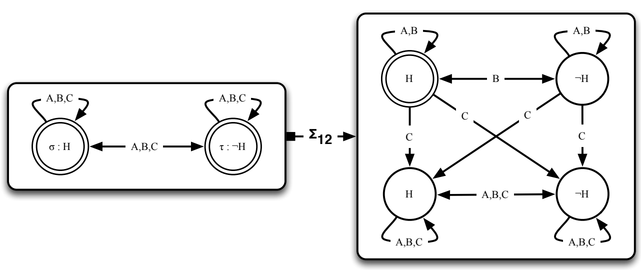

Fig. 15. Transition defined by $\mathbf{\Sigma_{12}}$.

occurrence but not the result, and agent C is oblivious to what has transpired. Fig. 16 shows the corresponding action model, $\mathbf{\Sigma_{13}}$, for this action occurrence. Applying the update yields the transition shown in Fig. 17.

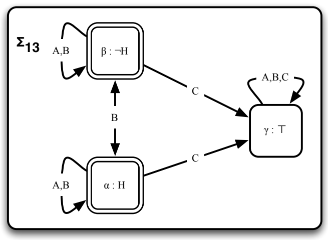

Fig. 16. Action model for a sensing action performed by agent A, with agent B as a partial observer, and an oblivious agent C.

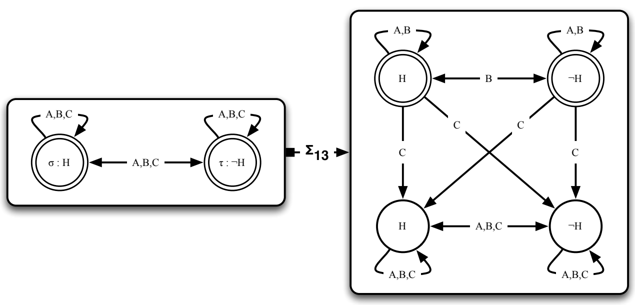

Fig. 17. Transition defined by $\mathbf{\Sigma_{13}}$.

As with sensing actions, we also extend nondeterministic sensing actions to the case where all three classes of agents are represented.

Example 10 (Nondeterministic Sensing Actions). As with Ex. 9, we begin with the Strongbox Domain comprised of three agents, A, B, and C. Suppose that agent A attempts to use a sensor (which may or may not work properly) to discern which face of the coin is showing. As before, let us suppose that agent B watches this occur (but does not know the outcome) and that agent C is oblivious. The action model, $\mathbf{\Sigma_{14}}$, for this scenario is shown in Fig. 18, and yields a considerably more complex successor state shown in Fig. 19.

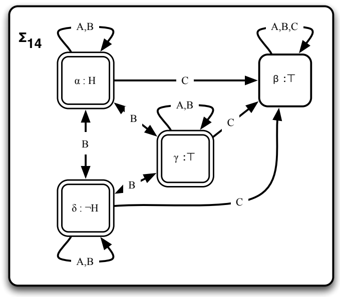

Fig. 18. Action model for agent A attempting to sense the value of H while partially observed by agent B with C unaware of what has happened.

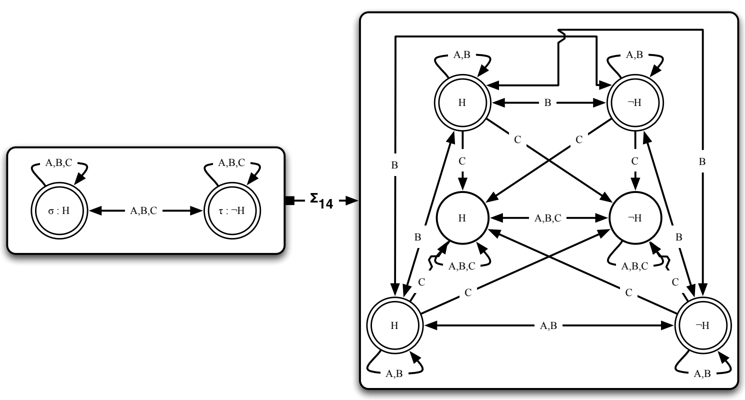

Fig. 19. Transition defined by $\mathbf{\Sigma_{14}}$.

4.3 Actions of Misdirection With Three Classes of Agents

Lastly we consider the class of actions dealing with lying and misdirection in which all three categories of agents are present.

Example 11 (Lying About the World). Consider the Strongbox Domain comprised of three agents, A, B, and C, and suppose that as before, it is common knowledge that none of them know which face of the coin is showing. Suppose that agent A lies to B, convincing that he knows the coin is truly showing heads, while agent C is oblivious to what has occurred. As with the case in which we have two classes of agents, such an action is modeled by the *sequential composition* of two simpler actions: the first in which agent B develops a suspicion that agent A may know which face of the coin is showing; and the second in which A actually lies. The action models for both of these actions, $\mathbf{\Sigma_{15}}$ and $\mathbf{\Sigma_{16}}$ respectively, are shown in Fig. 20 below. Applying the update operation induced

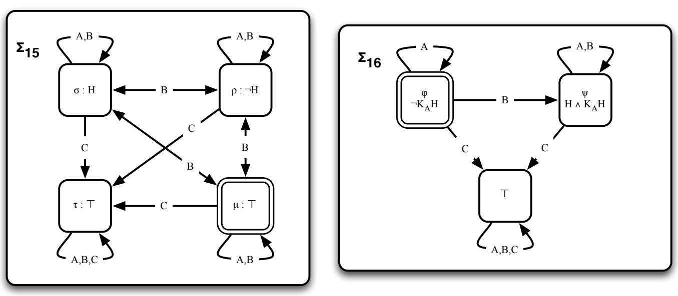

Fig. 20. Action models for lying about knowledge with three class of agents. $\mathbf{\Sigma_{15}}$: Agent B believes that agent A may have peeked. Agent C is oblivious. $\mathbf{\Sigma_{16}}$: Agent A does not know the value of H but successfully lies to B that it knows and the value is H. Agent C is oblivious.

by $\mathbf{\Sigma_{15}}$ to our initial state defines the intermediate transition shown Fig. 21. Once this intermediate state has been obtained, we apply the update operation induced by $\mathbf{\Sigma_{16}}$ in order to obtain our final successor state shown in Fig. 22.

In addition to lying about the value of a particular fluent, "The coin is facing heads.", agents may mislead each other about their knowledge with respect to a particular fluent, "I know which face of the coin is showing". The latter is captured in Ex. 12.

Example 12 (Lying About Knowledge). As with Ex. 11, we consider the Strongbox Domain consisting of three agents A, B, and C. Suppose that agent A lies to agent B by telling him that he knows which face of the coin is showing, while agent C remains oblivious. As before, this scenario is modeled by the sequential composition of two simpler actions: the first being identical to the one discussed

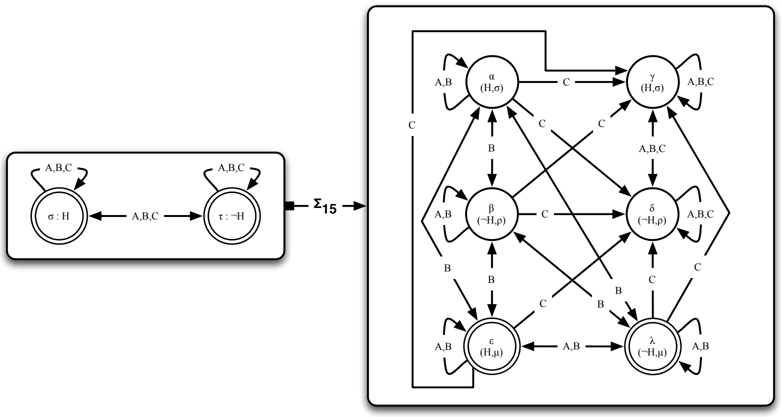

Fig. 21. Intermediate transition corresponding to agent B suspecting that A knows which face of the coin is showing, while agent C is oblivious.

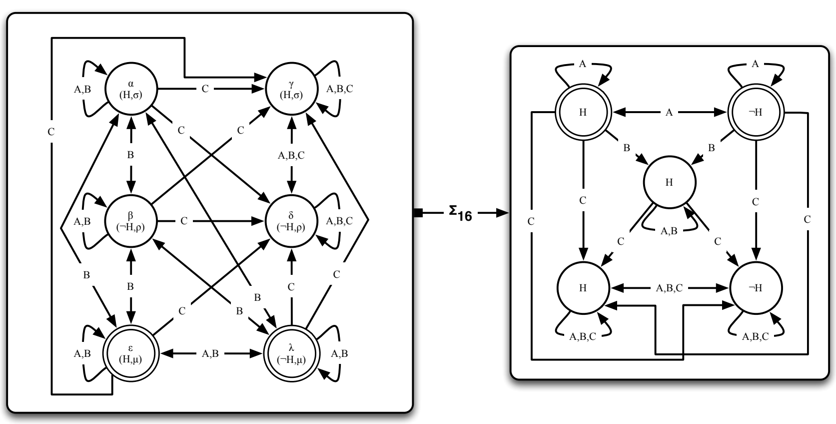

Fig. 22. Final transition for lying about the world with three classes of agents.

in Ex. 11, as is the intermediate transition; and the second being almost identical to the one discussed there as well, with the only change being in the modal formulae associated as preconditions to the component simple actions. The action model for this second action, $\mathbf{\Sigma_{17}}$, is shown in Fig. 23, and the final transition is shown in Fig. 24

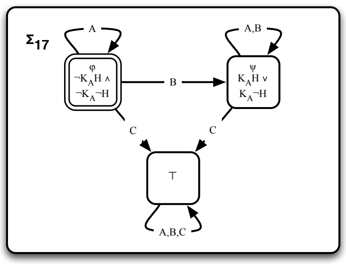

Fig. 23. Agent A does not know the value of H but successfully lies to B that it may know. Agent C is oblivious.

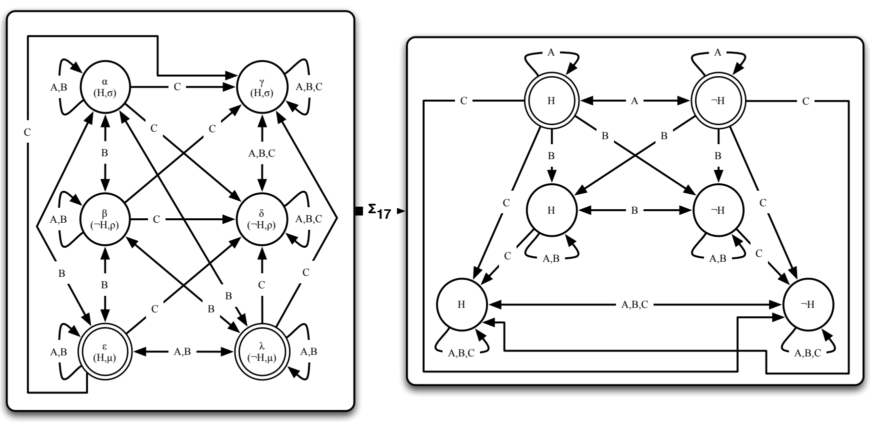

Fig. 24. Final transition when lying about knowledge with three classes of agents.

5 From Action Models to High Level Action Languages

In Sect. 3 we presented the Baltag-Moss action models and several examples of their use. We also presented several action signatures as defined by Baltag and Moss. Action models and signatures seem to be good technical devices for expressing certain kinds of multi-agent actions. The update operations induced by action models are simple to understand and seem to be a natural extension of the set product operator to graphs with nodes labeled by propositional formulae. Baltag and Moss also give a language to construct programs, which they call program models, from action models.

We believe that although action models, action signatures, and program models are a good start, further work needs to be done to express multi-agent action scenarios.

5.1 Future Directions on Action Models

Currently, even though one could give meaning to actions models that are constructed from the given action signatures, we could find no work in the literature that gives meaning to individual nodes and edges of action models in such a way that they can be used to build up the overall meaning of an action model. Discovering or developing such a semantics would be very helpful.

Alternatively, developing a general algorithm to construct action models for certain kinds of action scenarios would also be helpful. This seems to be a promising direction as we were able to generalize the examples in Sect. 3 to come with action models for various interesting multi-agent action scenarios when there are three classes of agents. We presented those examples in Sect. 4. Our action model examples of Sect. 4 are simple extensions of the known action models; yet they are novel in the sense that they do not follow any of the known action signatures and lead to new action signatures.

5.2 A High Level Action Language That Uses Action Models as a Semantic Tool

Another representational aspect of action models that needs to be enhanced is that there is often a dependency between an agent's role with respect to an action scenario and the state of the world. For example, consider an enhancement of the Strongbox Domain, where agent A has various actions at her disposal such as: (a) an action to distract another agent who is watching the strongbox and (b) an action to signal an agent that is not watching the strongbox to pay attention. Such actions when executed change subsequent action scenarios. For example, suppose that agent C is initially watching the strongbox and agent A distracts him. If agent A then peeks into the strongbox, C will be oblivious.

Such knowledge can be expressed in the style of high-level action language $\mathcal{A}$ [7] by *agent-role statements* of the following kinds:

- Y **observes** $peek(X)$ **if** $lookingAt(Y,X), near(Y,X)$
- Y **partially observes** $peek(X)$ **if** $lookingAt(Y,X), \neg near(Y,X)$

Using statements of the above kind one can determine who are the observers, partial observers, and oblivious agents with respect to an action like *peek*. Once that is determined one can then construct the appropriate action model and compute the transition due to that action model.

One can then borrow the other constructs of high level languages to express world changing actions as well as relationship between properties of the world. Some of those constructs are the following:

– Dynamic Causal Laws: a **causes** ϕ **if** ψ
– Executability Conditions: **executable** a **if** ϕ
– Sensing Actions: a **determines** f
– Non-deterministic Sensing Actions: a **may determine** ϕ
– Static Causal Laws: ψ **if** ϕ
– Initial State Axioms: **initially** ψ

Intuitive readings of the aforementioned statements are as follows. Dynamic causal laws state that when an action a is performed in a state where ϕ is true, ψ becomes true in the resulting state. Executability conditions state that an action a is only executable in states where ϕ is true. Sensing actions state that after the performing the action a, the agent knows the value of the fluent f. Nondeterministic sensing actions are taken to mean that after performing the action a, the agent may know the value of the fluent f. Static causal laws state that all states which satisfy ϕ must also satisfy ψ. Initial state axioms state that ψ is true in the initial state.

The above readings are appropriate for domains with one active agent. In the presence of multiple agents however, they may change slightly. For example, in a multi-agent domain the meaning of a sensing action is that after an occurrence of the action a, the value of the fluent f becomes known to those agents who are observing that action, the fact that the action occurred is known to the agents who are partially observing the action, while all other agents are oblivious.

Thus a high level action language for multi-agent domains can be defined by combining existing constructs from action languages with new ones such as *agent-role statements*. The semantics of such a language would then be defined by generalizing the transition semantics of existing action languages with the addition that in the presence of multiple agents, an action is enhanced to an action model by the application of agent-role statements, and then the action model is used to compute the corresponding transition.

The above is a good first step in the development of a high level language for multi-agent domains. However, several other concerns need to be addressed.

Various reasoning about action tasks, such as planning, require the notion of an initial state. Since in a multi-agent domain the initial state axioms may contain modal formulae, and the Kripke worlds satisfying them may have an infinite number of states, there is a need to identify a subset of modal logic for use in these axioms such that: the Kripke worlds satisfying them are finite; can be constructed easily; and are able to express interesting and important domains from the multi-agent literature (for example, the muddy-children domain [5,2]). One such sub-language is where ψ is a formula without modal operators, or has the form $\mathbf{C}\varphi$ or $\mathbf{C}(\mathbf{K}_i\varphi \vee \mathbf{K}_i\neg\varphi)$, where φ is a formula without modal operators.

5.3 Knowledge and Belief

A major concern that needs to be addressed in the future is that even though we may begin with the knowledge modality, after a number of actions occur we begin to have a mixture of belief and knowledge. For example, consider $\mathbf{\Sigma_3}$ and its associated transition as shown in Fig. 7. Prior to the execution of that action, it is common knowledge among agents A and B that neither knows the value of H. After the execution of $\mathbf{\Sigma_3}$, A knows that H is true. But what about agent B's knowledge? Per Baltag and Moss, the given reading of the successor state shown in Fig. 7 is that B knows that A does not know the value of H. However, it is often argued that one may only *know* truths, yet *believe* falsehoods. In this case, a more appropriate reading would be that B believes that A does not know the value of H. To be able to do this, one needs to capture both the knowledge and belief modalities. One possible approach would be to treat the Kripke models in [1] and this paper as a short hand for a more complex Kripke model that has accessibility relations covering both knowledge and belief. We are currently working on formalizing this structure.

6 Final Thoughts

As was mentioned in the introduction, considerable bodies of research with respect to both single, and multi-agent reasoning have been developed. With a few exceptions however ([13,8,6]), there has not been much crossover between these two areas. With regards to single-agent domains, a myriad of techniques has arisen, involving the use of action languages for reasoning about actions with application in planning, diagnosis, and other reasoning tasks. From the multi-agent standpoint, the use of modal logics and other methods has produced a body of work for reasoning about the knowledge and beliefs of the agents present in a domain.

The work done by Baltag and Moss begins to provide a framework for describing the effects of knowledge producing actions within a multi-agent setting. While somewhat limited from the standpoint of knowledge representation, we believe that a higher level action language can be developed by adding *agent-role statements* to existing action languages and characterizing the new language by translating the higher level language of causal laws to their corresponding action model representations. Successor states could then be defined in terms of the update operation induced by these action models. Once this has been accomplished, one can extend the work done with respect to various reasoning about action tasks such as planning and diagnosis from single, to multi-agent domains.

References

1. Baltag, A., Moss, L.: Logics for epistemic programs. Synthese (2004)
2. Baral, C., Gelfond, G., Son, T., Pontelli, E.: Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge. In: AAMAS 2010 (2010)

3. Baral, C., Son, T., Pontelli, E.: Modeling multi-agent domains in an action language: An empirical study using C. In: LPNMR. pp. 409–415 (2009)
4. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer (2007)
5. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. MIT press (1995)
6. Gelfond, G.: A declarative framework for modeling multi-agent systems. Masters Thesis, Texas Tech University (2007)
7. Gelfond, M., Lifschitz, V.: Representing actions in extended logic programs. In: Apt, K. (ed.) Joint International Conference and Symposium on Logic Programming. pp. 559–573. MIT Press (1992)
8. Ghaderi, H., Levesque, H., Lespérance, Y.: A logical theory of coordination and joint ability. In: AAAI 2007. pp. 421–426 (2007)
9. Levesque, H., Reiter, R., Lespérance, Y., Lin, F., Scherl, R.: GOLOG: A logic programming language for dynamic domains. In: Journal of Logic Programming 31(1-3), 59–84 (April-June 1997)
10. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. In: Meltzer, B., Michie, D. (eds.) Machine Intelligence, vol. 4, pp. 463–502. Edinburgh University Press (1969)
11. Reiter, R.: Knowledge in action: logical foundation for describing and implementing dynamical systems. MIT press (2001)
12. Sandewall, E.: The range of applicability of some non-monotonic logics for strict inertia. In: Journal of Logic and Computation 4(5), 581–616 (October 1994)
13. Shapiro, S., Lespérance, Y., Levesque, H.: The cognitive agents specification language and verification environment for multiagent systems. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (2002)