The Escapee Domain: A Multi-Agent Planning Domain

Gregory Gelfond

Actions@KR 2018

Research Fellow
Department of Computer Science
University of Nebraska Omaha

Outline

Introduction

Introduction

One of the distinguishing features of dynamic multi-agent domains is the fact that actions may effect both the *physical properties* of the domain, and what the agents *believe about them (and each other)*.

Introduction

One of the distinguishing features of dynamic multi-agent domains is the fact that actions may effect both the *physical properties* of the domain, and what the agents *believe about them (and each other)*.

Recently, a pair of action languages, $m\mathcal{A}+$ and $m\mathcal{AL}$ have been developed, which fuse together concepts from the *dynamic epistemic logic* and *action language* communities, allowing us to represent and reason about such domains.

Introduction

One of the distinguishing features of dynamic multi-agent domains is the fact that actions may effect both the *physical properties* of the domain, and what the agents *believe about them (and each other)*.

Recently, a pair of action languages, $m\mathcal{A}+$ and $m\mathcal{AL}$ have been developed, which fuse together concepts from the *dynamic epistemic logic* and *action language* communities, allowing us to represent and reason about such domains.

This talk covers a multi-agent planning domain, called the “Escapee Domain”, illustrating some of the interesting issues that arise with multi-agent planning, together with its axiomatization in the language $m\mathcal{A}+$.

The Escapee Domain

The Escapee Domain

Let us consider the following domain:

The Escapee Domain

Let us consider the following domain:

- Agent *A* has been bound and captured by a hostile agent *B*.

The Escapee Domain

Let us consider the following domain:

- Agent *A* has been bound and captured by a hostile agent *B*.
- To escape, *A* must be freed from his bonds.

The Escapee Domain

Let us consider the following domain:

- Agent *A* has been bound and captured by a hostile agent *B*.
- To escape, *A* must be freed from his bonds.
- Fortunately, agent *C* is a double agent in *B*'s organization and may release *A*.

The Escapee Domain

Let us consider the following domain:

- Agent *A* has been bound and captured by a hostile agent *B*.
- To escape, *A* must be freed from his bonds.
- Fortunately, agent *C* is a double agent in *B*'s organization and may release *A*.
- *C* must maintain his cover however, and will only release *A* if he can remain undetected.

The Escapee Domain

Let us consider the following domain:

- Agent *A* has been bound and captured by a hostile agent *B*.
- To escape, *A* must be freed from his bonds.
- Fortunately, agent *C* is a double agent in *B*'s organization and may release *A*.
- *C* must maintain his cover however, and will only release *A* if he can remain undetected.
- Once *A* has been released, he must work together with *C* to subdue *B* before making his escape.

The Escapee Domain

Let us consider the following domain:

- Agent *A* has been bound and captured by a hostile agent *B*.
- To escape, *A* must be freed from his bonds.
- Fortunately, agent *C* is a double agent in *B*'s organization and may release *A*.
- *C* must maintain his cover however, and will only release *A* if he can remain undetected.
- Once *A* has been released, he must work together with *C* to subdue *B* before making his escape.
- *A* will only work with *C* if he believes *C* is an ally.

The Escapee Domain

Let us consider the following domain:

- Agent *A* has been bound and captured by a hostile agent *B*.
- To escape, *A* must be freed from his bonds.
- Fortunately, agent *C* is a double agent in *B*'s organization and may release *A*.
- *C* must maintain his cover however, and will only release *A* if he can remain undetected.
- Once *A* has been released, he must work together with *C* to subdue *B* before making his escape.
- *A* will only work with *C* if he believes *C* is an ally.

How can *A* escape?

The Escapee Domain

This domain presents a number of interesting challenges:

The Escapee Domain

This domain presents a number of interesting challenges:

- Agent *C* must preserve his cover. This means that he has to be able to release *A* while keeping *B oblivious of his actions*.

The Escapee Domain

This domain presents a number of interesting challenges:

- Agent *C* must preserve his cover. This means that he has to be able to release *A* while keeping *B oblivious of his actions*.
- In otherwords, not only must he reason about the *physical effects* of his actions, but also about whether or not other agents will *perceive* them.

The Escapee Domain

This domain presents a number of interesting challenges:

- Agent *C* must preserve his cover. This means that he has to be able to release *A* while keeping *B oblivious of his actions*.
- In otherwords, not only must he reason about the *physical effects* of his actions, but also about whether or not other agents will *perceive* them.
- Furthermore, to effect his escape, *A* must act in concert with *C*, and this collaborative action is predicated on *A*'s beliefs about *C*.

The Escapee Domain

These challenges have been met by the action languages $m\mathcal{A}+$ and $m\mathcal{AL}$ through:

The Escapee Domain

These challenges have been met by the action languages $m\mathcal{A}+$ and $m\mathcal{AL}$ through:

- The recognition that both the *direct* and *indirect effects* of actions may alter the physical and epistemic properties of a domain, and the addition of linguistic constructs to represent them.

The Escapee Domain

These challenges have been met by the action languages $m\mathcal{A}+$ and $m\mathcal{AL}$ through:

- The recognition that both the *direct* and *indirect effects* of actions may alter the physical and epistemic properties of a domain, and the addition of linguistic constructs to represent them.
- The discovery of a certain class of fluents (called *perspective fluents*), which govern how the agents perceive the action occurrences in a given state, and the incorporation of this discovery into the semantics of the language.

These challenges have been met by the action languages $m\mathcal{A}+$ and $m\mathcal{AL}$ through:

- The recognition that both the *direct* and *indirect effects* of actions may alter the physical and epistemic properties of a domain, and the addition of linguistic constructs to represent them.
- The discovery of a certain class of fluents (called *perspective fluents*), which govern how the agents perceive the action occurrences in a given state, and the incorporation of this discovery into the semantics of the language.

In the slides that follow, we will flesh out these ideas some more in the context of an axiomatization of this domain in $m\mathcal{A}+$.

The Action Language $m\mathcal{A}+$

Before diving in to the axiomatization, it's important to note that in a multi-agent context, states of the domain must reflect both *physical* and *epistemic* properties.

Before diving in to the axiomatization, it's important to note that in a multi-agent context, states of the domain must reflect both *physical* and *epistemic* properties.

Physical properties of the domain are represented as in classical action languages by *fluents*, but epistemic properties are represented by *modal formulae* containing operators C_γ and $\mathcal{B}_\alpha$.

Before diving in to the axiomatization, it's important to note that in a multi-agent context, states of the domain must reflect both *physical* and *epistemic* properties.

Physical properties of the domain are represented as in classical action languages by *fluents*, but epistemic properties are represented by *modal formulae* containing operators C_γ and $\mathcal{B}_\alpha$.

Consequently, a state of the domain is now understood as a *pointed Kripke structure* defined by the language's semantics.

Before diving in to the axiomatization, it's important to note that in a multi-agent context, states of the domain must reflect both *physical* and *epistemic* properties.

Physical properties of the domain are represented as in classical action languages by *fluents*, but epistemic properties are represented by *modal formulae* containing operators C_γ and $\mathcal{B}_\alpha$.

Consequently, a state of the domain is now understood as a *pointed Kripke structure* defined by the language's semantics.

Edges as before are labeled by individual actions.

As is the case with its single-agent counterparts, we begin by establishing the domain's signature. For a multi-agent domain, the signature is defined as a triple of finite, disjoint sets:

As is the case with its single-agent counterparts, we begin by establishing the domain's signature. For a multi-agent domain, the signature is defined as a triple of finite, disjoint sets:

- $\mathcal{AG}$ — is a set of agent names;

As is the case with its single-agent counterparts, we begin by establishing the domain's signature. For a multi-agent domain, the signature is defined as a triple of finite, disjoint sets:

- $\mathcal{AG}$ — is a set of agent names;
- $\mathcal{F}$ — is a set of *fluents*;

As is the case with its single-agent counterparts, we begin by establishing the domain's signature. For a multi-agent domain, the signature is defined as a triple of finite, disjoint sets:

- $\mathcal{AG}$ — is a set of agent names;
- $\mathcal{F}$ — is a set of *fluents*;
- $\mathcal{A}$ — is a set of *elementary actions*

In the case of the Escapee Domain, a partial domain signature could be:

$$\mathcal{AG} = \{A, B, C\}$$
$$\mathcal{F} = \{bound(\alpha), attentive(\alpha), free(\alpha), allies(\alpha_1, \alpha_2), united(\alpha_1, \alpha_2)\}$$
$$\mathcal{A} = \{release(\alpha_1, \alpha_2), unite(\alpha_1, \alpha_2), distract(\alpha_1, \alpha_2)\}$$

where α, α_1, and α_2 are variables over $\mathcal{AG}$.

With the signature in place, the initial state is defined by a collection of *initial state axioms*:

With the signature in place, the initial state is defined by a collection of *initial state axioms*:

$$\textbf{\textit{initially}}\ C_{\{A,B,C\}}(\mathit{attentive}(\alpha) \wedge \mathit{bound}(A))$$

$$\textbf{\textit{initially}}\ C_{\{A,B,C\}}(\mathcal{B}_B \neg \mathit{allies}(A, C))$$

$$\textbf{\textit{initially}}\ (\mathcal{B}_C \mathit{allies}(A, C) \wedge \mathcal{B}_A \mathit{allies}(A, C))$$

With the signature in place, the initial state is defined by a collection of *initial state axioms*:

$$\textbf{\textit{initially}}\ C_{\{A,B,C\}}(\mathit{attentive}(\alpha) \wedge \mathit{bound}(A))$$
$$\textbf{\textit{initially}}\ C_{\{A,B,C\}}(\mathcal{B}_B \neg \mathit{allies}(A, C))$$
$$\textbf{\textit{initially}}\ (\mathcal{B}_C \mathit{allies}(A, C) \wedge \mathcal{B}_A \mathit{allies}(A, C))$$

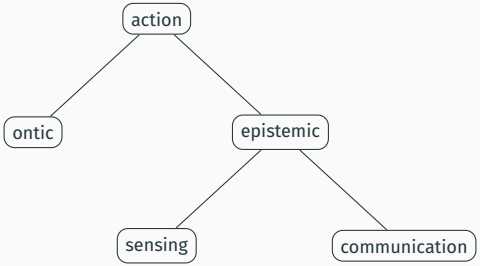
Generally speaking, such axioms are statements of the form:

$$\textbf{\textit{initially}}\ \varphi$$

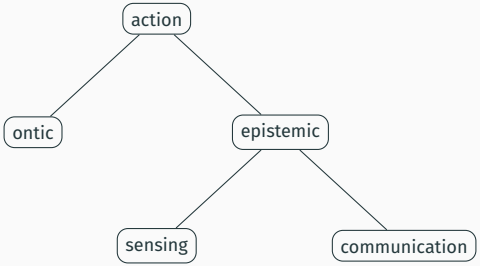
where φ is a restricted kind of *modal formula*, and have the informal reading of: "φ is initially true."

In a multi-agent context, actions fall into three distinct categories:

In a multi-agent context, actions fall into three distinct categories:



In a multi-agent context, actions fall into three distinct categories:



In $m\mathcal{A}+$, each of these distinct classes of actions is paired with a construct of the language.

The causal relationships between an *ontic action* and its *direct effects* are represented by *dynamic causal laws* which are statements of the form:

$$a \textbf{ causes } f \textbf{ if } \phi$$

where a is an action, f is a fluent literal, and ϕ is a modal formula[1].

[1] As with initial state axioms, these are from a syntactically restricted class.

For instance: A single agent may *release* another agent causing him to no longer be *bound*. This may be readily represented by the following dynamic causal law:

For instance: A single agent may *release* another agent causing him to no longer be *bound*. This may be readily represented by the following dynamic causal law:

$$release(\alpha_1, \alpha_2) \textbf{ causes } \neg bound(\alpha_2)$$

For instance: A single agent may *release* another agent causing him to no longer be *bound*. This may be readily represented by the following dynamic causal law:

$$release(\alpha_1, \alpha_2) \textbf{ causes } \neg bound(\alpha_2)$$

What's new is that in this particular domain, such an action may have additional *epistemic effects*. For example:

For instance: A single agent may *release* another agent causing him to no longer be *bound*. This may be readily represented by the following dynamic causal law:

$$release(\alpha_1, \alpha_2) \textbf{ causes } \neg bound(\alpha_2)$$

What's new is that in this particular domain, such an action may have additional *epistemic effects*. For example:

- If *C* releases *A* but *B* is mindful of his surroundings, *B* will become aware of this.

For instance: A single agent may *release* another agent causing him to no longer be *bound*. This may be readily represented by the following dynamic causal law:

$$release(\alpha_1, \alpha_2) \textbf{ causes } \neg bound(\alpha_2)$$

What's new is that in this particular domain, such an action may have additional *epistemic effects*. For example:

- If *C* releases *A* but *B* is mindful of his surroundings, *B* will become aware of this.
- If *C* releases *A* while *B* is distracted, *B* will not be aware of what has transpired.

For instance: A single agent may *release* another agent causing him to no longer be *bound*. This may be readily represented by the following dynamic causal law:

$$release(\alpha_1, \alpha_2) \textbf{ causes } \neg bound(\alpha_2)$$

What's new is that in this particular domain, such an action may have additional *epistemic effects*. For example:

- If *C* releases *A* but *B* is mindful of his surroundings, *B* will become aware of this.
- If *C* releases *A* while *B* is distracted, *B* will not be aware of what has transpired.

Perspective axioms allow us to represent the dependance of such indirect effects on the values of *perspective fluents*.

With respect to the action *release*, the relevant *perspective axioms* are as follows:

With respect to the action *release*, the relevant *perspective axioms* are as follows:

$$\{\alpha_1, \alpha_2\} \textbf{\textit{ observes }} \mathit{release}(\alpha_1, \alpha_2)$$

$$\{\alpha\} \textbf{\textit{ observes }} \mathit{release}(\alpha_1, \alpha_2) \textbf{\textit{ if }} \mathit{attentive}(\alpha)$$

With respect to the action *release*, the relevant *perspective axioms* are as follows:

$$\{\alpha_1, \alpha_2\} \textbf{\textit{ observes }} \mathit{release}(\alpha_1, \alpha_2)$$
$$\{\alpha\} \textbf{\textit{ observes }} \mathit{release}(\alpha_1, \alpha_2) \textbf{\textit{ if }} \mathit{attentive}(\alpha)$$

Generally speaking, such statements have the form:

$$X \textbf{\textit{ observes }} a \textbf{\textit{ if }} \phi$$

where X is a set of agents, a is an action, and ϕ is a modal formula.

With respect to the action *release*, the relevant *perspective axioms* are as follows:

$$\{\alpha_1, \alpha_2\} \textbf{\textit{ observes }} \mathit{release}(\alpha_1, \alpha_2)$$
$$\{\alpha\} \textbf{\textit{ observes }} \mathit{release}(\alpha_1, \alpha_2) \textbf{\textit{ if }} \mathit{attentive}(\alpha)$$

Generally speaking, such statements have the form:

$$X \textbf{\textit{ observes }} a \textbf{\textit{ if }} \phi$$

where X is a set of agents, a is an action, and ϕ is a modal formula.

Note that the second axiom encodes the idea that attentive agents are aware of occurrences of the action *release* (and hence its consequences).

To effect his escape, *A* must be able to work together with *C*. In order to do so, they must *unite*.

To effect his escape, *A* must be able to work together with *C*. In order to do so, they must *unite*.

However, we know that *A* will only work with *C* if he believes *C* is an ally.

To effect his escape, *A* must be able to work together with *C*. In order to do so, they must *unite*.

However, we know that *A* will only work with *C* if he believes *C* is an ally.

Note that this second point requires us to be able to *constrain the executability* of this action to agents who *believe themselves to be allied*.

This kind of constraint may be readily expressed by an *executability condition*, which is a statement of the form:

$$\textbf{\textit{executable}}\ a\ \textbf{\textit{if}}\ \phi$$

where a is an action, ϕ is a modal formula.

This leads us to the following axiomatization of the action *unite*:

[2] Omitted for the sake of brevity.

This leads us to the following axiomatization of the action *unite*:

$$unite(\alpha_1, \alpha_2) \textbf{ causes } united(\alpha_1, \alpha_2)$$
$$\textbf{executable } unite(\alpha_1, \alpha_2) \textbf{ if } \mathcal{B}_{\alpha_1} allies(\alpha_1, \alpha_2)$$

[2] Omitted for the sake of brevity.

This leads us to the following axiomatization of the action *unite*:

$$\mathit{unite}(\alpha_1, \alpha_2) \textbf{ \textit{causes} } \mathit{united}(\alpha_1, \alpha_2)$$

$$\textbf{\textit{executable}}\ \mathit{unite}(\alpha_1, \alpha_2)\ \textbf{\textit{if}}\ \mathcal{B}_{\alpha_1}\mathit{allies}(\alpha_1, \alpha_2)$$

along with the requisite perspective axioms[2].

[2] Omitted for the sake of brevity.

With all of these elements in place, a *collaborative action* such as *subdue* may be expressed in a straightforward manner[3]:

[3] As before, the relevant perspective axioms have been omitted.

With all of these elements in place, a *collaborative action* such as *subdue* may be expressed in a straightforward manner[3]:

$$subdue(\alpha_1, \alpha_2, \alpha_3) \textbf{ causes } bound(\alpha_3)$$

$$\textbf{executable } subdue(\alpha_1, \alpha_2, \alpha_3) \textbf{ if } united(\alpha_1, \alpha_2)$$

[3] As before, the relevant perspective axioms have been omitted.

Concluding Thoughts

Concluding Thoughts

Once the domain has been axiomatized, the semantics of the language enables us to answer questions involving both *temporal projection* and *planning* in such a setting. For example:

Concluding Thoughts

Once the domain has been axiomatized, the semantics of the language enables us to answer questions involving both *temporal projection* and *planning* in such a setting. For example:

- If *C* immediately releases *A* while *B* is attentive, he will be discovered.

Concluding Thoughts

Once the domain has been axiomatized, the semantics of the language enables us to answer questions involving both *temporal projection* and *planning* in such a setting. For example:

- If *C* immediately releases *A* while *B* is attentive, he will be discovered.
- If *C* however chooses to *distract* agent *B* prior to releasing *A*, his cover will be intact.

Concluding Thoughts

Once the domain has been axiomatized, the semantics of the language enables us to answer questions involving both *temporal projection* and *planning* in such a setting. For example:

- If *C* immediately releases *A* while *B* is attentive, he will be discovered.
- If *C* however chooses to *distract* agent *B* prior to releasing *A*, his cover will be intact.
- In order for *A* escape, *B* must first be distracted, then *A* must be released. Once this is done, *A* and *C* unite to subdue *B*, after which *A* makes his way to freedom.

Concluding Thoughts

In addition, the domain may be extended by the inclusion of communication actions to allow *A* and *C* to communicate their allied status rather than making it explicit in the initial state.

Concluding Thoughts

In addition, the domain may be extended by the inclusion of communication actions to allow *A* and *C* to communicate their allied status rather than making it explicit in the initial state.

It should be noted that the semantics of $m\mathcal{A}+$ has a natural translation into a logic program under the answer-set semantics. This allows us to reduce *temporal projection* and *planning* to finding the answer sets of programs based on this translation.

In addition, the domain may be extended by the inclusion of communication actions to allow *A* and *C* to communicate their allied status rather than making it explicit in the initial state.

It should be noted that the semantics of $m\mathcal{A}+$ has a natural translation into a logic program under the answer-set semantics. This allows us to reduce *temporal projection* and *planning* to finding the answer sets of programs based on this translation.

Lastly, the successor to $m\mathcal{A}+$, called $m\mathcal{AL}$ further extends the language by the inclusion of *state constraints* in the manner of $\mathcal{AL}$.

Thank You