Cooperative Epistemic Multi-Agent Planning for Implicit Coordination

Thorsten Engesser
Institut für Informatik
Albert-Ludwigs-Universität
Freiburg, Germany
engesset@cs.uni-freiburg.de

Thomas Bolander
DTU Compute
Technical University of Denmark
Copenhagen, Denmark
tobo@dtu.dk

Robert Mattmüller
Institut für Informatik
Albert-Ludwigs-Universität
Freiburg, Germany
mattmuel@cs.uni-freiburg.de

Bernhard Nebel
Institut für Informatik
Albert-Ludwigs-Universität
Freiburg, Germany
nebel@cs.uni-freiburg.de

Epistemic planning can be used for decision making in multi-agent situations with distributed knowledge and capabilities. Recently, Dynamic Epistemic Logic (DEL) has been shown to provide a very natural and expressive framework for epistemic planning. We extend the DEL-based epistemic planning framework to include perspective shifts, allowing us to define new notions of sequential and conditional planning with implicit coordination. With these, it is possible to solve planning tasks with joint goals in a decentralized manner without the agents having to negotiate about and commit to a joint policy at plan time. First we define the central planning notions and sketch the implementation of a planning system built on those notions. Afterwards we provide some case studies in order to evaluate the planner empirically and to show that the concept is useful for multi-agent systems in practice.

1 Introduction

One important task in Multi-Agent Systems is to collaboratively reach a joint goal with multiple autonomous agents. The problem is particularly challenging in situations where the knowledge and capabilities required to reach the goal are distributed among the agents. Most existing approaches therefore apply some centralized coordinating instance from the outside, strictly separating the stages of communication and negotiation from the agents' internal planning and reasoning processes. In contrast, building upon the epistemic planning framework by Bolander and Andersen [8], we propose a decentralized planning notion in which each agent has to individually reason about the entire problem and autonomously decide when and how to (inter-)act. For this, both reasoning about the other agents' possible contributions and reasoning about their capabilities of performing the same reasoning is needed. We achieve our notion of implicitly coordinated plans by requiring all desired communicative abilities to be modeled as epistemic actions which then can be planned alongside their ontic counterparts, thus enabling the agents to perform observations and coordinate at run time. It captures the intuition that communication clearly constitutes an action by itself and, more subtly, that even a purely ontic action can play a communicative role (e.g. indirectly suggesting follow-up actions to another agent). Thus, for many problems our approach appears quite natural. On the practical side, the epistemic planning framework allows a very expressive way of defining both the agents' physical and communicative abilities.

Consider the following example scenario. Bob would like to borrow the apartment of his friend Anne while she is away on vacation. Anne would be very happy to do him this favor. So they now have the

Sujata Ghosh and R. Ramanujam: M4M9
EPTCS 243, 2017, pp. 75–90, doi:10.4204/EPTCS.243.6

joint goal of making sure that Bob can enter the apartment when he arrives. Anne will think about how to achieve the goal, and might come up with the following plan: Anne puts the key under the door mat; when Bob arrives, Bob takes the key from under the door mat; Bob opens the door with the key. Note that the plan does not only contain the actions required by Anne herself, but also the actions of Bob. These are the kind of multi-agent plans that this paper is about.

However, the plan just presented does not count as an *implicitly coordinated* plan. When Bob arrives at the apartment, he will clearly not know that the key is under the door mat, unless Anne has told him, and this announcement was not part of the plan just presented. If Anne has the ability to take Bob's perspective (she has a *Theory of Mind* concerning Bob [29]), Anne should of course be able to foresee this problem, and realize that her plan can not be expected to be successful. An improved plan would then be: Anne puts the key under the door mat; Anne calls Bob to let him know where the key is; when Bob arrives, Bob takes the key from under the door mat; Bob opens the door with the key. This *does* qualify as an implicitly coordinated plan. Anne now knows that Bob will know that he can find the key under the door mat and hence will be able to reach the goal. Anne does not have to request or even coordinate the sub-plan for Bob (which is: take key under door mat; open door with key), as she knows he will himself be able to determine this sub-plan given the information she provides. This is an important aspect of implicit coordination: coordination happens implicitly as a consequence of observing the actions of others (including announcements), never explicitly through agreeing or committing to a specific plan. The essential contributions of this paper are to formally define this notion of implicitly coordinated plans as well as to document and benchmark an implemented epistemic planner that produces such plans.

Our work is situated in the area of distributed problem solving and planning [15] and directly builds upon the framework introduced by Bolander and Andersen [8] and Löwe, Pacuit, and Witzel [22], who formulated the planning problem in the context of Dynamic Epistemic Logic (DEL) [13]. Andersen, Bolander, and Jensen [4] extended the approach to allow strong and weak conditional planning in the single-agent case. Algorithmically, (multi-agent) epistemic planning can be approached either by compilation to classical planning [2, 21, 23] or by search in the space of "nested" [8] or "shallow" knowledge states [26, 27, 28]. Since compilation approaches to classical planning can only deal with bounded nesting of knowledge (or belief), similar to Bolander and Andersen [8], we use search in the space of epistemic states to find a solution. One of the important features that distinguishes our work from more traditional multi-agent planning [9] is the explicit notion of perspective shifts needed for agents to reason about the possible plan contributions of other agents—and hence needed to achieve implicit coordination.

Our concepts can be considered related to recent work in temporal epistemic logics [10, 19, 20], which addresses a question similar to ours, namely what groups of agents can jointly achieve under imperfect information. These approaches are based on concurrent epistemic game structures. Our approach is different in a number of ways, including: 1) As in classical planning, our actions and their effects are explicitly and compactly represented in an action description language (using the event models of DEL); 2) Instead of joint actions we have sequential action execution, where the order in which the agents act is not predefined; 3) None of the existing solution concepts considered in temporal epistemic logics capture the stepwise shifting perspective underlying our notion of implicitly coordinated plans.

The present paper is an extended and revised version of a paper presented at the Workshop on Distributed and Multi-Agent Planning (DMAP) 2015 (without archival proceedings). The present paper primarily offers an improved presentation: extended and improved introduction, improved motivation, better examples, improved formulation of definitions and theorems, simplified notation, and more discussions of related work. The technical content is essentially as in the original DMAP paper, except we now compare implicitly coordinated plans to standard sequential plans, we now formally derive the correct notion of an implicitly coordinated plan from the natural conditions it should satisfy, and we have added a proposition that gives a semantic characterisation of implicitly coordinated policies.

2 Theoretical Background

To represent planning tasks as the 'apartment borrowing' example of the introduction, we need a formal framework where: (1) agents can reason about the knowledge and ignorance of other agents; (2) both fully and partially observable actions can be described in a compact way (Bob doesn't see Anne placing the key under the mat). Dynamic Epistemic Logic (DEL) satisfies these conditions. We first briefly recapitulate the foundations of DEL, following the conventions of Bolander and Andersen [8]. What is new in this exposition is mainly the parts on perspective shifts in Section 2.2.

We now define epistemic languages, epistemic states and epistemic actions. All of these are defined relative to a given finite set of *agents* $\mathcal{A}$ and a given finite set of *atomic propositions* P. To keep the exposition simple, we will not mention the dependency on $\mathcal{A}$ and P in the following.

2.1 Epistemic Language and Epistemic States

Definition 1. The *epistemic language* $\mathcal{L}_{\text{KC}}$ is $\varphi ::= \top \mid \bot \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid C\varphi$ where $p \in P$ and $i \in \mathcal{A}$.

We read $K_i\varphi$ as "agent i knows φ" and $C\varphi$ as "it is common knowledge that φ".

Definition 2. An *epistemic model* is $\mathcal{M} = \langle W, (\sim_i)_{i\in\mathcal{A}}, V\rangle$ where

- The *domain* W is a non-empty finite set of *worlds*.
- $\sim_i \subseteq W \times W$ is an equivalence relation called the *indistinguishability relation* for agent i.
- $V : P \to \mathcal{P}(W)$ assigns a *valuation* to each atomic proposition.

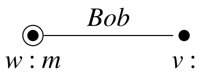
For $W_d \subseteq W$, the pair $(\mathcal{M}, W_d)$ is called an *epistemic state* (or simply a *state*), and the worlds of W_d are called the *designated worlds*. A state is called *global* if $W_d = \{w\}$ for some world w (called the *actual world*), and we then often write $(\mathcal{M}, w)$ instead of $(\mathcal{M}, \{w\})$. We use S^{gl} to denote the set of global states. For any state $s = (\mathcal{M}, W_d)$, we let $Globals(s) = \{(\mathcal{M}, w) \mid w \in W_d\}$. A state $(\mathcal{M}, W_d)$ is called a *local state* for agent i if W_d is closed under $\sim_i$. A local state for i is *minimal* if W_d is a minimal set closed under $\sim_i$. We use $S_i^{\min}$ to denote the set of minimal local states of i. Given a state $s = (\mathcal{M}, W_d)$, the *associated local state* of agent i, denoted s^i, is $(\mathcal{M}, \{v \mid v \sim_i w \text{ and } w \in W_d\})$.

Definition 3. Let $(\mathcal{M}, W_d)$ be a state with $\mathcal{M} = \langle W, (\sim_i)_{i\in\mathcal{A}}, V\rangle$. For $i \in \mathcal{A}$, $p \in P$ and $\varphi, \psi \in \mathcal{L}_{\text{KC}}$, we define truth as follows (with the propositional cases being standard and hence left out):

$$\begin{aligned}
(\mathcal{M}, W_d) &\models \varphi && \text{iff } (\mathcal{M}, w) \models \varphi \text{ for all } w \in W_d \\
(\mathcal{M}, w) &\models K_i\varphi && \text{iff } (\mathcal{M}, w') \models \varphi \text{ for all } w' \sim_i w \\
(\mathcal{M}, w) &\models C\varphi && \text{iff } \mathcal{M}, w' \models \varphi \text{ for all } w' \sim^* w
\end{aligned}$$

where $\sim^*$ is the transitive closure of $\bigcup_{i\in\mathcal{A}} \sim_i$.

Example 1. Let $\mathcal{A} = \{Anne, Bob\}$ and $P = \{m\}$, where m is intended to express that the key is under the door mat. Consider the following global state $s = (\mathcal{M}, w)$, where the nodes represent worlds, the edges represent the indistinguishability relations (reflexive edges left out), and ◉ is used for designated worlds:

$s = (\mathcal{M}, w) =$ 

Each node is labeled by the name of the world, and the list of atomic propositions true at the world. The state represents a situation where the key is under the door mat (m is true at the actual world w), but Bob considers it possible that it isn't (m is not true at the world v indistinguishable from w by Bob). We can verify that in this state Anne knows that the key is under the mat, Bob doesn't, and Anne knows that he doesn't: $s \models K_{Anne} m \wedge \neg K_{Bob} m \wedge K_{Anne} \neg K_{Bob} m$. The fact that Bob does not know the key to be under the mat can also be expressed in terms of local states. Bob's local perspective on the state s is his associated local state $s^{Bob} = (\mathcal{M}, \{w, v\})$. We have $s^{Bob} \not\models m$, signifying that from Bob's perspective, m can not be verified.

2.2 Perspective Shifts

In general, given a global state s, the associated local state s^i will represent agent i's internal perspective on that state. Going from s^i to $(s^i)^j$ amounts to a *perspective shift*, where agent i's perspective on the perspective of agent j is taken. In Example 1, Anne's perspective on the state s is s^{Anne}, which is s itself. Bob's perspective is $s^{Bob} = (\mathcal{M}, \{w, v\})$. When Anne wants to reason about the available actions to Bob, e.g. whether he will be able to take the key from under the door mat or not, she will have to shift to his perspective, i.e. reason about what holds true in $(s^{Anne})^{Bob}$, which is the same as s^{Bob} in this case. This type of perspective shift is going to be central in our notion of implicitly coordinated plans, since it is essential to the ability of an agent to reason about other agents' possible contributions to a plan *from their own perspective*. As the introductory example shows, this ability is essential: If Anne can not reason about Bob's contribution to the overall plan from his own perspective, she will not realize that she needs to call him to let him know where the key is.

Note that any local state s induces a unique set of global states, $Globals(s)$, and that we can hence choose to think of s as a compact representation of the "belief state" $Globals(s)$ over global states.

We have the following basic properties concerning perspective shifts (associated local states), where the third follows directly from the two first:

Proposition 1. *Let s be a state, $i \in \mathcal{A}$ and $\varphi \in \mathcal{L}_{\mathrm{KC}}$.*

1. *$s^i \models \varphi$ iff $s \models K_i \varphi$.*
2. *If s is local for agent i then $s^i = s$.*
3. *If s is local for agent i then $s \models \varphi$ iff $s \models K_i \varphi$.* □

2.3 Dynamic Language and Epistemic Actions

To model actions, like announcing locations of keys and picking them up, we use the *event models* of DEL.

Definition 4. An *event model* is $\mathcal{E} = \langle E, (\sim_i)_{i \in \mathcal{A}}, pre, post \rangle$ where

- The *domain* E is a non-empty finite set of *events*.
- $\sim_i \subseteq E \times E$ is an equivalence relation called the *indistinguishability relation* for agent i.
- $pre : E \to \mathcal{L}_{\mathrm{KC}}$ assigns a *precondition* to each event.
- $post : E \to \mathcal{L}_{\mathrm{KC}}$ assigns a *postcondition* to each event. For all $e \in E$, $post(e)$ is a conjunction of literals (atomic propositions and their negations, including $\top$).

For $E_d \subseteq E$, the pair $(\mathcal{E}, E_d)$ is called an *epistemic action* (or simply an *action*), and the events in E_d are called the *designated events*. An action is called *global* if $E_d = \{e\}$ for some event e, and we then often write $(\mathcal{E}, e)$ instead of $(\mathcal{E}, \{e\})$. Similar to states, $(\mathcal{E}, E_d)$ is called a *local action* for agent i when E_d is closed under $\sim_i$.

Each event of an epistemic action represents a different possible outcome. By using multiple events $e, e' \in E$ that are indistinguishable (i.e. $e \sim_i e'$), it is possible to obfuscate the outcomes for some agent $i \in \mathcal{A}$, i.e. modeling partially observable actions. Using event models with $|E_d| > 1$, it is also possible to model sensing actions (where a priori, multiple outcomes are considered possible), and nondeterministic actions [8].

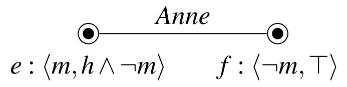
The *product update* is used to specify the successor state resulting from the application of an action in a state.

Definition 5. Let a state $s = (\mathcal{M}, W_d)$ and an action $a = (\mathcal{E}, E_d)$ be given with $\mathcal{M} = \langle W, (\sim_i)_{i \in \mathcal{A}}, V\rangle$ and $\mathcal{E} = \langle E, (\sim_i)_{i \in \mathcal{A}}, pre, post\rangle$. Then the *product update* of s with a is $s \otimes a = (\langle W', (\sim_i')_{i \in \mathcal{A}}, V'\rangle, W_d')$ where

- $W' = \{(w,e) \in W \times E \mid \mathcal{M}, w \models pre(e)\}$
- $\sim_i' = \{((w,e),(w',e')) \in W' \times W' \mid w \sim_i w' \text{ and } e \sim_i e'\}$
- $V'(p) = \{(w,e) \in W' \mid post(e) \models p \text{ or } (\mathcal{M}, w \models p \text{ and } post(e) \not\models \neg p)\}$
- $W_d' = \{(w,e) \in W' \mid w \in W_d \text{ and } e \in E_d\}$.

Example 2. Consider the following epistemic action *try-take* $= (\mathcal{E}, \{e, f\})$, using the same conventions as for epistemic models, except each event e is labeled by $\langle pre(e), post(e)\rangle$:

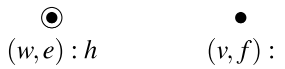
try-take $= (\mathcal{E}, \{e, f\}) =$


Anne
$e : \langle m, h \wedge \neg m\rangle$
$f : \langle \neg m, \top\rangle$

It represents the action of Bob attempting to take the key from under the mat. The event $e = \langle m, h \wedge \neg m\rangle$ represents that if the key is indeed under the mat (the precondition m is true), then the result will be that Bob holds the key and it is no longer under the mat (the postcondition $h \wedge \neg m$ becomes true). The event $f = \langle \neg m, \top\rangle$ represents that if the key is not under the mat, nothing will happen (the postcondition is the trivial one, $\top$). Note the indistinguishability edge for Anne: She is not there to see whether the action is successful or not. Note however that she is still is aware that either e or f happens, so the action represents the situation where she knows that he is attempting to take the key, but not necessarily whether he is successful (except if she already either knows m or knows $\neg m$).

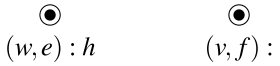
Letting s denote the state from Example 1, we can calculate the result of executing *try-take* in s:

$s \otimes$ *try-take* $=$



Note that the result is for Bob to have the key and for this to be common knowledge among Anne and Bob ($s \otimes$ *try-take* $\models Ch$). So it seems that if we assume initially to be in the state s, and want to find a plan to achieve h, then the simple plan consisting only of the action *try-take* should work. It is, however, a bit more complicated than that. Let us assume that Bob does strong planning, that is, only looks for plans that are guaranteed to reach the goal. The problem is then that, from Bob's perspective, *try-take* can not be guaranteed to reach the goal. This is formally because we have:

$s^{Bob} \otimes$ *try-take* $=$



Both worlds being designated, but distinguishable, means that Bob at *plan time* (before executing the action) considers them both as possible outcomes of *try-take*, but is aware that he will at *run time* (after having executed the action) know which one it is (see [8] for a more full discussion of the plan time/run time distinction). Since the world (v, f) is designated, we have $s^{Bob} \otimes \mathit{try\text{-}take} \not\models h$. So from Bob's perspective, *try-take* might fail to produce h and is hence not a strong plan to achieve h. Intuitively, it is of course simply because he does not, at plan time, know whether the key is under the mat or not.

Since $s^{Anne} = s$ and $s \otimes \mathit{try\text{-}take} \models h$, it might seem that *try-take* is still a strong plan to achieve h from the perspective of Anne. But in fact, it is not, at least not of the implicitly coordinated type we will define below. The issue is, *try-take* is still an action that Bob has to execute, but Anne knows that Bob does not know it to be succesful, and she can therefore not expect him to execute it. The idea is that when Anne comes up with a plan that involves actions of Bob, she should change her perspective to his, in order to find out what he can be expected to do. Technically speaking, it is because $(s^{Anne})^{Bob} \otimes \mathit{try\text{-}take} \not\models h$ that the plan is not implicitly coordinated from the perspective of Anne.

We extend the language $\mathcal{L}_{\mathrm{KC}}$ into the *dynamic language* $\mathcal{L}_{\mathrm{DEL}}$ by adding a modality $[(\mathcal{E}, e)]$ for each global action $(\mathcal{E}, e)$. The truth conditions are extended with the following standard clause from DEL: $(\mathcal{M}, w) \models [(\mathcal{E}, e)]\varphi$ iff $(\mathcal{M}, w) \models pre(e)$ implies $(\mathcal{M}, w) \otimes (\mathcal{E}, e) \models \varphi$.

We define the following abbreviations:

$$[(\mathcal{E}, E_d)]\varphi := \bigwedge_{e \in E_d} [(\mathcal{E}, e)]\varphi \quad \text{and} \quad \langle(\mathcal{E}, E_d)\rangle\varphi := \neg[(\mathcal{E}, E_d)]\neg\varphi.$$

We say that an action $(\mathcal{E}, E_d)$ is *applicable* in a state $(\mathcal{M}, W_d)$ if for all $w \in W_d$ there is an event $e \in E_d$ s.t. $(\mathcal{M}, w) \models pre(e)$. Intuitively, an action is applicable in a state if for each possible situation (designated world), at least one possible outcome (designated event) is specified.

Example 3. Consider again the state s from Example 1 and the action *try-take* from Example 2. The action *try-take* is trivially seen to be applicable in the state s, since the designated event e has its precondition satisfied in the designated world w. The action *try-take* is also applicable in s^{Bob}, since $(\mathcal{M}, w) \models pre(e)$ and $(\mathcal{M}, v) \models pre(f)$. This shows that *try-take* is applicable from the perspective of Bob. Intuitively, it is so because it is only an action for *attempting* to take the key. Even if the key is not under the mat, the action will specify an outcome (having the trivial postcondition $\top$).

Let $s = (\mathcal{M}, W_d)$ denote an epistemic state and $a = (\mathcal{E}, E_d)$ an action. Andersen [3] shows that a is applicable in s iff $s \models \langle a\rangle\top$, and that $s \models [a]\varphi$ iff $s \otimes a \models \varphi$. We now define a further abbreviation: $((a))\varphi := \langle a\rangle\top \wedge [a]\varphi$. Hence:

$$s \models ((a))\varphi \ \text{ iff } \ a \text{ is applicable in } s \text{ and } s \otimes a \models \varphi \tag{1}$$

Thus $((a))\varphi$ means that the application of a is possible and will (necessarily) lead to a state fulfilling φ.

3 Cooperative Planning

As mentioned in the introduction, in this paper we assume each action to be executable by a single agent, that is, we are not considering joint actions. In our 'apartment borrowing' example there are two agents, Anne and Bob. They are supposed to execute different parts of the plan to reach the goal of Bob getting access to the apartment. For instance, Anne is the one to put the key under the mat, and Bob is the one to take it from under the mat. To represent who performs which actions of a plan, we will introduce what we call an *owner function* (inspired by the approach of Löwe, Pacuit, and Witzel [22]). An *owner function* is defined to be a mapping $\omega : A \rightarrow \mathcal{A}$, mapping each action to the agent who can execute it. For any action $a \in A$, we call $\omega(a)$ the *owner* of a. Note that by defining the owner function this way, every action has a *unique* owner. This can be done without loss of generality, since we can always include any number of semantically equivalent actions in A, if we wish some action to be executable by several agents (e.g., if we want both Anne and Bob to be able to open and close the door). We can now define epistemic planning tasks.

Definition 6. A *cooperative planning task* (or simply a *planning task*) $\Pi = \langle s_0, A, \omega, \gamma \rangle$ consists of an initial epistemic state s_0, a finite set of epistemic actions A, an owner function $\omega : A \to \mathcal{A}$, and a goal formula γ of $\mathcal{L}_{\mathrm{KC}}$. Each $a \in A$ has to be local for $\omega(a)$. When s_0 is a global state, we call it a *global planning task*. When s_0 is local for agent i, we call it a *planning task for agent i*. Given a planning task $\Pi = \langle s_0, A, \omega, \gamma \rangle$ and a state s, we define $\Pi(s) := \langle s, A, \omega, \gamma \rangle$. Given a planning task $\Pi = \langle s_0, A, \omega, \gamma \rangle$, the *associated planning task for agent i* is $\Pi^i = \Pi(s^i)$.

Given a multi-agent system facing a global planning task Π, each individual agent i is facing the planning task Π^i (agent i cannot observe the global initial state s_0 directly, only the associated local state ${s_0}^i$).

In the following, we will investigate various possible solution concepts for cooperative planning tasks. The solution to a planning task is called a *plan*. A plan can either be sequential (a sequence of actions) or conditional (a policy). We will first, in Section 3.1, consider the simplest possible type of solution, a *standard sequential plan*. Then, in Section 3.2, we are going to introduce the more complex notion of a *sequential implicitly coordinated plan*, and in Section 3.3 this will be generalized to *implicitly coordinated policies*.

3.1 Standard Sequential Plans

The standard notion of a sequential plan in automated planning is as follows (see, e.g., [17]). An action sequence $(a_1, \ldots, a_n)$ is called a (standard sequential) plan if for every $i \leq n$, a_i is applicable in the result of executing the action sequence $(a_1, \ldots, a_{i-1})$ in the initial state, and when executing the entire sequence $(a_1, \ldots, a_n)$ in the initial state, a state satisfying the goal formula is reached. Let us transfer this notion into the DEL-based setting of this paper. In our setting, the result of executing an action a in a state s is given as $s \otimes a$. Hence, the above conditions for $(a_1, \ldots, a_n)$ being a plan can be expressed in the following way in our setting, where s_0 denotes the initial state, and γ the goal formula: for every $i \leq n$, a_i is applicable in $s_0 \otimes a_1 \otimes \cdots \otimes a_{i-1}$, and $s_0 \otimes a_1 \otimes \cdots \otimes a_n \models \gamma$. Note that by equation (1) above, these conditions are equivalent to simply requiring $s_0 \models ((a_1))((a_2)) \cdots ((a_n))\gamma$. Hence we get the following definition.

Definition 7. Let $\Pi = \langle s_0, A, \omega, \gamma \rangle$ be a planning task. A *standard sequential plan* for Π is a sequence $(a_1, \ldots, a_n)$ of actions from A satisfying $s_0 \models ((a_1))((a_2)) \cdots ((a_n))\gamma$.

This solution concept is equivalent to the one considered in [8]. As the following example shows, it is not sufficiently strong to capture the notion of an ‘implicitly coordinated plan’ that we are after.

Example 4. Consider again the scenario of Example 2 where the key is initially under the mat, Bob does not know this, and the goal is for Bob to have the key. The only action available in the scenario is for Bob to attempt to take the key from under the mat. Using the states and actions defined in Examples 1 and 2, we can express this scenario as a cooperative planning task $\Pi = \langle s_0, A, \omega, \gamma \rangle$ where $s_0 = s$, $A = \{\textit{try-take}\}$, $\omega(\textit{try-take}) = \textit{Bob}$, $\gamma = h$. In Example 2, we informally concluded that the plan only consisting of *try-take* is a strong plan, since it is guaranteed to reach the goal, but that it is not a strong plan from the perspective of Bob. Given our formal definitions, we can now make this precise as follows:

1. The sequence $(\textit{try-take})$ is a standard sequential plan for Π.

2. The sequence $(\textit{try-take})$ is *not* a standard sequential plan for Π^{Bob}.

The first item follows from the fact that *try-take* is applicable in s (Example 3), and that $s \otimes \textit{try-take} \models h$ (Example 2). The second item follows from $s^{Bob} \otimes \textit{try-take} \not\models h$ (Example 2).

We also have that $(\textit{try-take})$ is a standard sequential plan for Π^{Anne}, since $s^{Anne} = s$. This proves that the notion of a standard sequential plan is insufficient for our purposes. If Anne is faced with the planning task Π^{Anne}, she should not be allowed to consider $(\textit{try-take})$ as a sufficient solution to the problem. She should be aware that the action *try-take* is to be executed by Bob, and from Bob's perspective, $(\textit{try-take})$ is *not* a (strong) solution to the planning task (Item 2 above). So we need a way of explicitly integrating perspective-shifts into our notion of a solution to a planning task, and this is what we will do next.

3.2 Implicitly Coordinated Sequential Plans

It follows from Definition 7 that $(a_1,\dots,a_n)$ is a standard sequential plan for some planning task $\Pi = \langle s_0, A, \omega, \gamma\rangle$ iff $a_1,\dots,a_n \in A$ and the formula $((a_1))\cdots((a_n))\gamma$ is true in s_0. More generally, we can think of a planning notion as being defined via a mapping X that takes a plan π and a planning task Π as parameters, and returns a logical formula $X(\pi,\Pi)$ such that, for all states s, $s \models X(\pi,\Pi)$ iff π is a plan for $\Pi(s)$. In the case of standard sequential plans, it follows directly from Definition 7 that X would be defined like this:

$$X((a_1,\dots,a_n),\langle s_0,A,\omega,\gamma\rangle) = ((a_1))\cdots((a_n))\gamma \tag{2}$$

for all planning tasks $\langle s_0,A,\omega,\gamma\rangle$ and all $a_1,\dots,a_n \in A$.

We now wish to define a similar mapping Y, so that $s \models Y(\pi,\Pi)$ iff π is an *implicitly coordinated plan* for $\Pi(s)$. Our strategy is to list the natural conditions that Y should satisfy, and then derive the exact definition of Y (and hence implicitly coordinated plans) directly from those. First of all, the empty action sequence, denoted by ε, should be an implicitly coordinated plan iff it satisfies the goal formula, which is expressed by the following simple condition on Y:

$$Y(\varepsilon,\langle s_0,A,\omega,\gamma\rangle) = \gamma. \tag{3}$$

For non-empty action sequences, the 'apartment borrowing' example studied above gives us the following insights. If Anne is trying to come up with a plan where one of the steps is to be executed by Bob, then Anne has to make sure that Bob can himself verify his action to be applicable, and that he can himself verify that executing the action will lead to a state where the rest of the plan will succeed. More generally, for an action sequence $(a_1,\dots,a_n)$ to be considered implicitly coordinated, the owner of the first action a_1 has to *know* that a_1 is applicable and will lead to a situation where $(a_2,\dots,a_n)$ is again an implicitly coordinated plan. This leads us directly to the following condition on Y, for all planning tasks $\langle s_0,A,\omega,\gamma\rangle$ and all $a_1,\dots,a_n \in A$ with $n \geq 1$:

$$Y((a_1,\dots,a_n),\langle s_0,A,\omega,\gamma\rangle) = K_{\omega(a_1)}((a_1))Y((a_2,\dots,a_n),\langle s_0,A,\omega,\gamma\rangle) \tag{4}$$

It is now easy to see that any mapping Y satisfying (3) and (4) must necessarily be defined as follows, for all planning tasks $\langle s_0,A,\omega,\gamma\rangle$ and all action sequences $(a_1,\dots,a_n) \in A$:

$$Y((a_1,\dots,a_n),\langle s_0,A,\omega,\gamma\rangle) = K_{\omega(a_1)}((a_1))K_{\omega(a_2)}((a_2))\cdots K_{\omega(a_n)}((a_n))\gamma$$

This leads us directly to the following definition.

Definition 8. Let $\Pi = \langle s_0, A, \omega, \gamma \rangle$ be a cooperative planning task. An *implicitly coordinated plan* for Π is a sequence $(a_1, \ldots, a_n)$ of actions from A such that:

$$s_0 \models K_{\omega(a_1)}((a_1))K_{\omega(a_2)}((a_2)) \cdots K_{\omega(a_n)}((a_n))\gamma \tag{5}$$

If $(a_1, \ldots, a_n)$ is an implicitly coordinated plan for Π^i, then it is said to be an *implicitly coordinated plan for agent i* to the planning task Π.

Note that the formula used to define implicitly coordinated plans above is *uniquely* determined by the natural conditions (3) and (4).

The following proposition gives a more structural, and semantic, characterization of implicitly coordinated plans. It becomes clear that such plans can be found by performing a breadth-first search over the set of successively applicable actions, shifting the perspective for each state transition to the owner of the respective action.

Proposition 2. *For a cooperative planning task* $\Pi = \langle s_0, A, \omega, \gamma \rangle$*, a non-empty sequence* $(a_1, \ldots, a_n)$ *of actions from A is an implicitly coordinated plan for* Π *iff* a_1 *is applicable in* $s_0^{\omega(a_1)}$ *and* $(a_2, \ldots, a_n)$ *is an implicitly coordinated plan for* $\Pi(s_0^{\omega(a_1)} \otimes a_1)$.

The proposition can be seen as a semantic counterpart of (4), and is easily proven using (1), Proposition 1 and (5).

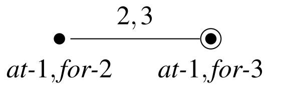
Example 5. Consider again the planning task $\Pi = \langle s_0, A, \omega, \gamma \rangle$ of Example 4 with $s_0 = s$, $A = \{\textit{try-take}\}$, $\omega(\textit{try-take}) = \textit{Bob}$, $\gamma = h$. In Example 4 we concluded that $(\textit{try-take})$ is a standard sequential plan for Π. From Example 2, we have that $s^{Bob} \otimes \textit{try-take} \not\models h$, and hence $s \not\models K_{\omega(\textit{try-take})}((\textit{try-take}))h$ (using (1) and Proposition 1). This shows that, as expected, $(\textit{try-take})$ is not an implicitly coordinated plan for Π.

In the introduction, we noted that the solution to this problem would be for Anne to make sure to announce the location of the key to Bob. Let us now treat this formally within the developed framework. We need to give Anne the possibility of announcing the location of the key, so we define a new planning task $\Pi' = \langle s_0, A', \omega, \gamma \rangle$ with $A' = \{\textit{try-take}, \textit{announce}\}$. Here *announce* is the action $\odot\, e : \langle m, \top \rangle$ with $\omega(\textit{announce}) = \textit{Anne}$. In DEL, this action is known as a *public announcement* of m. It can now easily be formally verified that

$$s \models K_{Anne}((\textit{announce}))K_{Bob}((\textit{try-take}))h.$$

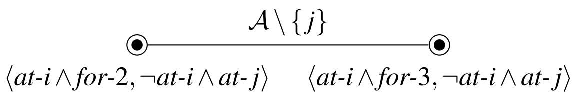
In words: Anne knows that she can announce the location of the key, and that this will lead to a situation where Bob knows he can attempt to take the key, and he knows that he will be successful in this attempt. In other words, $(\textit{announce}, \textit{try-take})$ is indeed an implicitly coordinated plan to achieve that Bob has the key, consistent with our informal analysis in the introduction of the paper.

Example 6. Consider a situation with agents $\mathcal{A} = \{1, 2, 3\}$ where a letter is to be passed from agent 1 to one of the other two agents, possibly via the third agent. Mutually exclusive propositions $\textit{at-}1, \textit{at-}2, \textit{at-}3 \in P$ are used to denote the current carrier of the letter, while $\textit{for-}1, \textit{for-}2, \textit{for-}3 \in P$ denote the addressee. In our example, agent 1 has a letter for agent 3, so *at*-1 and *for*-3 are initially true.

$s_0 =$ 

In s_0, all agents know that agent 1 has the letter (*at*-1), but agents 2 and 3 do not know who of them is the addressee (*for*-2 or *for*-3). We assume that agent 1 can only exchange letters with agent 2 and agent 2 can only exchange letters with agent 3. We thus define the four possible actions $a_{12}, a_{21}, a_{23}, a_{32}$, with a_{ij} being the composite action of agent i publicly passing the letter to agent j and privately informing him about the correct addressee (the name of the addressee is on the envelope, but only visible to the receiver). I.e.

$a_{ij} =$


$\mathcal{A}\setminus\{j\}$
$\langle$*at-i* $\wedge$ *for*-2, $\neg$*at-i* $\wedge$ *at-j*$\rangle$
$\langle$*at-i* $\wedge$ *for*-3, $\neg$*at-i* $\wedge$ *at-j*$\rangle$

Given that the joint goal is to pass a letter to its addressee, the global planning task then is $\Pi = \langle s_0, A, \omega, \gamma\rangle$ with $A = \{a_{12}, a_{21}, a_{23}, a_{32}\}$, $\omega(a_{ij}) = i$ for all i, j, and $\gamma = \bigwedge_{i\in\{1,2,3\}}$ (*for-i* $\rightarrow$ *at-i*). Consider the action sequence (a_{12}, a_{23}): Agent 1 passes the letter to agent 2, and agent 2 passes it on to agent 3. It can now be verified that ${s_0}^1 \models K_1((a_{12}))K_2((a_{23}))\gamma$ and ${s_0}^i \not\models K_1((a_{12}))K_2((a_{23}))\gamma$ for $i = 2, 3$. Hence (a_{12}, a_{23}) is an implicitly coordinated plan for agent 1, but not for agents 2 and 3. This is because in the beginning agents 2 and 3 do not know to whom of them the letter is intended and hence cannot verify that (a_{12}, a_{23}) will lead to a goal state. However, after agent 1's execution of a_{12}, agent 2 can distinguish between the possible addressees at run time, and find his subplan (a_{23}), as contemplated by agent 1.

3.3 Conditional Plans

Sequential plans are often not sufficient to solve a given epistemic planning task. In particular, as soon as branching on nondeterministic action outcomes or obtained observations becomes necessary, we need *conditional* plans to solve such a task. Unlike Andersen, Bolander, and Jensen [4], who represent conditional plans as action trees with branches depending on knowledge formula conditions, we represent them as policy functions $(\pi_i)_{i\in\mathcal{A}}$, where each π_i maps minimal local states of agent i into actions of agent i. We now define two different types of policies, *joint policies* and *global policies*, and later show them to be equivalent.

Definition 9 (Joint policy)**.** Let $\Pi = \langle s_0, A, \omega, \gamma\rangle$ be a cooperative planning task. Then a *joint policy* $(\pi_i)_{i\in\mathcal{A}}$ consists of partial functions $\pi_i : S_i^{\min} \rightarrow A$ satisfying for all states s and actions a: if $\pi_i(s) = a$ then $\omega(a) = i$ and a is applicable in s.

Definition 10 (Global policy)**.** Let $\Pi = \langle s_0, A, \omega, \gamma\rangle$ be a cooperative planning task. Then a *global policy* π is a mapping $\pi : S^{\mathrm{gl}} \rightarrow \mathcal{P}(A)$ satisfying the requirements *knowledge of preconditions* (KOP), *per-agent determinism* (DET), and *uniformity* (UNIF):

(KOP) For all $s \in S^{\mathrm{gl}}$, $a \in \pi(s)$: a is applicable in $s^{\omega(a)}$.

(DET) For all $s \in S^{\mathrm{gl}}, a, a' \in \pi(s)$ with $\omega(a) = \omega(a')$: $a = a'$.

(UNIF) For all $s, t \in S^{\mathrm{gl}}, a \in \pi(s)$ with $s^{\omega(a)} = t^{\omega(a)}$: $a \in \pi(t)$.

Proposition 3. *Any joint policy $(\pi_i)_{i\in\mathcal{A}}$ induces a global policy π given by*

$$\pi(s) = \left\{\pi_i(s^i) \mid i \in \mathcal{A} \textit{ and } \pi_i(s^i) \textit{ is defined}\right\}.$$

Conversely, any global policy π induces a joint policy $(\pi_i)_{i\in\mathcal{A}}$ given by

$$\pi_i(s^i) = a \quad \textit{for all } (s, A') \in \pi,\ a \in A' \textit{ with } \omega(a) = i.$$

Furthermore, the two mappings $(\pi_i)_{i\in\mathcal{A}} \mapsto \pi$ (mapping joint policies to induced global policies) and $\pi \mapsto (\pi_i)_{i\in\mathcal{A}}$ (mapping global policies to induced joint policies) are each other's inverse.

Proof. First we prove that the induced mapping π as defined above is a global policy. Condition (KOP): If $a \in \pi(s)$ then $\pi_i(s^i) = a$ for some i, and by definition of joint policy this implies a is applicable in s^i. Condition (DET): Assume $a, a' \in \pi(s)$ with $\omega(a) = \omega(a')$. By definition of π we have $\pi_i(s^i) = a$ and $\pi_j(s^j) = a'$ for some i, j. By definition of joint policy, $\omega(a) = i$ and $\omega(a') = j$. Since $\omega(a) = \omega(a')$ we get $i = j$ and hence $\pi_i(s^i) = \pi_j(s^j)$. This implies $a = a'$. Condition (UNIF): Assume $a \in \pi(s)$ and $s^{\omega(a)} = t^{\omega(a)}$. By definition of π and joint policy, we get $\pi_i(s^i) = a$ for $i = \omega(a)$. Thus $s^i = t^i$, and since $\pi_i(s^i) = a$, we immediately get $\pi_i(t^i) = a$ and hence $a \in \pi(t)$. We now prove that the induced mappings $(\pi_i)_{i\in\mathcal{A}}$ defined above form a joint policy. Constraint (KOP) ensures the applicability property as required by Definition 9, while the constraints (DET) and (UNIF) ensure the right-uniqueness of each partial function π_i. It is easy to show that the two mappings $(\pi_i)_{i\in\mathcal{A}} \mapsto \pi$ and $\pi \mapsto (\pi_i)_{i\in\mathcal{A}}$ are each other's inverse, using their definition. □

By Proposition 3, we can identify joint and global policies, and will in the following move back and forth between the two. Notice that Definitions 9 and 10 allow a policy to distinguish between modally equivalent states. A more sophisticated definition avoiding this is possible, but is beyond the scope of this paper. Usually, a policy π is only considered to be a solution to a planning task if it is closed in the sense that π is defined for all non-goal states reachable following π. Here, we want to distinguish between two different notions of closedness: one that refers to all states reachable from a centralized perspective, and one that refers to all states considered reachable when tracking perspective shifts. To that end, we distinguish between centralized and perspective-sensitive successor functions.

We take a *successor function* to be any function $\sigma : S^{\mathrm{gl}} \times A \to \mathcal{P}(S^{\mathrm{gl}})$. Successor functions are intended to map pairs of states s and actions a into the states $\sigma(s,a)$ that can result from executing a in s. Which states can result from executing a in s depend on whether we take the objective, centralized view, or whether we take the subjective view of an agent. An agent might subjectively consider more outcomes possible than are objectively possible. We define the *centralized successor function* as $\sigma_{cen}(s,a) = Globals(s \otimes a)$. It specifies the global states that are possible after the application of a in s. If closedness of a global policy π based on the centralized successor function is required, then no execution of π will ever lead to a non-goal state where π is undefined. Like for sequential planning, we are again interested in the decentralized scenario where each agent has to plan and decide when and how to act by himself under incomplete knowledge. We achieve this by encoding the perspective shifts to the next agent to act in the *perspective-sensitive successor function* $\sigma_{ps}(s,a) = Globals(s^{\omega(a)} \otimes a)$. Unlike $\sigma_{cen}(s,a)$, $\sigma_{ps}(s,a)$ considers a state s' to be a successor of s after application of a if *agent* $\omega(a)$ *considers* s' *possible* after the application of a, not only if s' *is actually possible from a global perspective*. Thus, $\sigma_{cen}(s,a)$ is always a (possibly strict) subset of $\sigma_{ps}(s,a)$, and a policy π_{ps} that is closed wrt. $\sigma_{ps}(s,a)$ must be defined for *at least* the states for which a policy π_{cen} that is closed wrt. $\sigma_{cen}(s,a)$ must be defined. This corresponds to the intuition that solution existence for decentralized planning with implicit coordination is a stronger property than solution existence for centralized planning. For both successor functions, we can now formalize what a strong solution is that can be executed by the agents. Our notion satisfies the usual properties of strong plans [11], namely closedness, properness and acyclicity.

Definition 11 (Strong Policy)**.** Let $\Pi = \langle s_0, A, \omega, \gamma\rangle$ be a cooperative planning task and σ a successor function. A global policy π is called a *strong policy* for Π with respect to σ if

(i) *Finiteness:* π is finite.

(ii) *Foundedness:* for all $s \in Globals(s_0)$, (1) $s \models \gamma$, or (2) $\pi(s) \neq \emptyset$.

(iii) *Closedness:* for all $(s,A') \in \pi$, $a \in A', s' \in \sigma(s,a)$, (1) $s' \models \gamma$, or (2) $\pi(s') \neq \emptyset$.

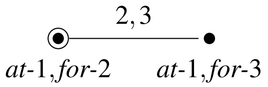
Note that we do not explicitly require acyclicity, since this is already implied by a literal interpretation of the product update semantic that ensures unique new world names after each update. It then follows from (i) and (iii) that π is proper. We call strong plans with respect to σ_{cen} *centralized policies* and strong plans with respect to σ_{ps} *implicitly coordinated policies*.

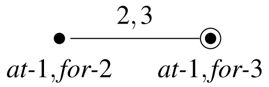
Analogous to Proposition 2, we want to give a semantic characterization of implicitly coordinated policies. For this, we first define a *successor of a state* s_0 *by following a policy* π to be a state $s \in Globals(s'_0 \otimes a)$ for arbitrary states $s'_0 \in Globals(s_0)$ and arbitrary actions $a \in \pi(s'_0)$. We can then show that if π is an implicitly coordinated policy for a planning task Π, each successor state s of the initial state s_0 either will already be a goal state, or there will be some agent $i \in \mathcal{A}$ who can find an implicitly coordinated policy for his own associated planning task prescribing an action for himself.

Proposition 4. *Let π be an implicitly coordinated policy for a planning task $\Pi = \langle s_0, A, \omega, \gamma \rangle$ and let s be a non-goal successor state of s_0 by following π. Then there is an agent $i \in A$ such that $\pi(s)$ contains at least one of agent i's actions and π is an implicitly coordinated policy of $\Pi(s^i)$.*

Proof. The existence of an action $a \in \pi(s)$ with *some* owner i follows directly from the closedness of implicitly coordinated policies. We need to show that π is also implicitly coordinated for $\Pi(s^i)$. Finiteness and closedness of π still hold for $\Pi(s^i)$, since π was already finite and closed for Π, and this does not change when replacing s_0 with s^i. For foundedness of π for $\Pi(s^i)$, we have to show that π is defined and returns a nonempty set of actions for all global states $s' \in Globals(s^i)$. For s itself, we already known that $a \in \pi(s)$. By uniformity, since all such s' are indistinguishable from s for agent i, π must assign the same action a to all states $s' \in Globals(s^i)$. □

Example 7. Consider again the letter passing problem introduced in Example 6. Let $s_{0,2}$ and $s_{0,3}$ denote the global states that are initially considered possible by agent 2.

$s_{0,2} =$ 

$s_{0,3} =$ 

With $s_{1,3} = s_{0,3} \otimes a_{12}$, a policy for agent 2 is given by $\pi_1 = \{s_{0,3} \mapsto a_{12}, s_{0,2} \mapsto a_{12}\}, \pi_2 = \{s_{1,3} \mapsto a_{23}\}$. After the contemplated application of a_{12} by agent 1 (in both cases), agent 2 can distinguish between $s_{1,2} = s_{0,2} \otimes a_{12}$, where the goal is already reached and nothing has to be done, and $s_{1,3}$, where agent 2 can apply a_{23}, leading directly to the goal state $s_{1,3} \otimes a_{23}$. Thus, π is an implicitly coordinated policy for Π^2. While in the sequential case, agent 2 has to wait for the first action a_{12} of agent 1 to be able to find its subplan, it can find the policy $(\pi_i)_{i \in \mathcal{A}}$ in advance by explicitly planning for a run-time distinction.

In general, strong policies can be found by performing an AND-OR search, where AND branching corresponds to branching over different epistemic worlds and OR branching corresponds to branching over different actions. By considering modally equivalent states as duplicates and thereby transforming the procedure into a graph search, space and time requirements can be reduced, although great care has to be taken to deal with cycles correctly.

4 Experiments

We implemented a planner that is capable of finding implicitly coordinated plans and policies[1], and conducted two experiments: one small case study of the Russian card problem [12] intended to show how this problem can be modeled and solved from an individual perspective, and one experiment investigating the scaling behavior of our approach on private transportation problems in the style of Examples 6 and 7, using instances of increasing size.

4.1 Russian Card Problem

In the Russian card problem, seven cards numbered $0, \ldots, 6$ are randomly dealt to three agents. Alice and Bob get three cards each, while Eve gets the single remaining card. Initially, each agent only knows its own cards. The task is now for Alice and Bob to inform each other about their respective cards using only public announcements, without revealing the holder of any single card to Eve. The problem

[1] Our code can be downloaded at `https://gkigit.informatik.uni-freiburg.de/tengesser/planner`

was analyzed and solved from the global perspective by van Ditmarsch et al. [14], and a given protocol was *verified* from an individual perspective by Ågotnes et al. [1]. We want to *solve* the problem from the individual perspective of agent Alice and find an implicitly coordinated policy for her. To keep the problem computationally feasible, we impose some restrictions on the resulting policy, namely that the first action has to be Alice truthfully announcing five possible alternatives for her own hand, and that the second one has to be Bob announcing the card Eve is holding. Without loss of generality, we fix one specific initial hand for Alice, namely 012. From a plan for this initial hand, plans for all other initial hands can be obtained by renaming. For simplicity, we only generate *applicable* actions for Alice, i.e. announcements that include her true hand 012. This results in the planning task having a total of 46376 options for the first action, and 7 for the second action. Still, the initial state s_0 consists of 140 worlds, one for each possible deal of cards. Agents can only distinguish worlds where their own hands differ. Alice's designated worlds in her associated local state of s_0 are those four worlds in which she holds hand 012.

Our planner needs approximately two hours and 600MB of memory to come up with a solution policy. In the solution, Alice first announces her hand to be one of 012, 034, 156, 236, and 245. It can be seen that each of the five hands other than the true hand 012 contains at least one of Alice's and one of Bob's cards, meaning that Bob will immediately be able to identify the complete deal. Also, Eve stays unaware of the individual cards of Alice and Bob since she will be able to rule out exactly two of the hands, with each of Alice and Bob's cards being present and absent in at least one of the remaining hands. Afterwards, Alice can wait for Bob to announce that Eve has either card 3, 4, 5 or 6.

4.2 Mail Instances

Our second experiment concerns the letter passing problem from Examples 6 and 7. We generalized the scenario to allow an arbitrary number of agents with an arbitrary undirected neighborhood graph, indicating which agents are allowed to directly pass letters to each other. As neighborhood graphs, we used randomly generated Watts-Strogatz small-world networks [30], exhibiting characteristics that can also be found in social networks. Watts-Strogatz networks have three parameters: The number N of nodes (determining the number of agents in our setting), the average number K of neighbors per node (roughly determining the average branching factor of a search for a plan), and the probability β of an edge being a "random shortcut" instead of a "local connection" (thereby influencing the shortest path lengths between agents). We only generate *connected* networks in order to guarantee plan existence.

We distinguish between the example domains MAILTELL and MAILCHECK. To guarantee plan existence, in both scenarios the actions are modeled such as to ensure that the letter position remains common knowledge among the agents in all reachable states. The mechanics of MAILTELL directly correspond to those given in Example 6. There is only one type of action, publicly passing the letter to a neighboring agent while privately informing him about the final addressee. This allows for sequential implicitly coordinated plans. In the resulting plans, letters are simply moved along a shortest path to the addressee. In contrast, in MAILCHECK, an agent that has the letter can only check if he himself is the addressee or not using a separate action (without learning the actual addressee if it is not him). To ensure plan existence in this scenario, we allow an agent to pass on the letter only if it is destined for someone else. Unlike in MAILTELL, conditional plans are required here. In a solution policy, the worst-case sequence of successively applied actions contains an action passing the letter to each agent at least once. As soon as the addressee has been reached, execution is stopped.

Experiments were conducted for both scenarios with different parameters (Table 1). For finding sequential as well as conditional plans, our implementation uses breadth-first search over a regular graph

Table 1: Runtime evaluation for randomly generated MAILTELL and MAILCHECK instances

(a) MAILTELL, $K = 4$, $\beta = 0.1$

agents	10	20	30	40	50
direct path	1.4	2.3	3.1	3.7	3.6
time/s	0.02	0.14	0.65	2.38	5.02

(b) MAILCHECK, $K = 2$, $\beta = 0.1$

agents	10	15	20	25	30
full path	10.4	16.1	21.7	27.6	33.2
time/s	0.02	0.18	1.14	8.03	38.76

and over an AND-OR graph, respectively. For each set of parameters, 100 trials were performed. For MAILTELL, *direct path* denotes the average shortest path length between sender and addressee, while for MAILCHECK, *full path* denotes the average length of a shortest path passing through all agents starting from the sender.

While the shortest path length between sender and addressee grows very slowly with the number of agents (due to the shortcut connections in the network), the shortest path passing through all agents roughly corresponds to the number of agents. Since these measures directly correspond to the minimal plan lengths, the observed exponential growth of space and time requirements with respect to them (and to the base K) is unsurprising.

Note also that in both scenarios, the number of agents determines the number of worlds (one for each possible addressee) in the initial state. Since the preconditions of the available actions are mutually exclusive, this constitutes an upper bound on the number of worlds per state throughout the search. Thus we get only a linear overhead in comparison to directly searching the networks for the relevant paths.

5 Conclusion and Future Work

We introduced an interesting new cooperative, decentralized planning concept without the necessity of explicit coordination or negotiation. Instead, by modeling all possible communication directly as plannable actions and relying on the ability of the autonomous agents to put themselves into each other's shoes (using perspective shifts), some problems can be elegantly solved achieving implicit coordination between the agents. We briefly demonstrated an implementation of both the sequential and conditional solution algorithms and its performance on the Russian card problem and two letter passing problems.

Based on the foundation this paper provides, a number of challenging problems needs to be addressed. First of all, concrete problems (e.g. epistemic versions of established multi-agent planning tasks) need to be formalized with a particular focus on the question of which kinds of communicative actions the agents would need to solve these problems in an implicitly coordinated way. As seen in the MAILTELL benchmark, the dynamic epistemic treatment of a problem does not necessarily lead to more than linear overhead. It will be interesting to identify classes of tractable problems and see how agents cope in a simulated environment. Another issue that is relevant in practice concerns the interplay of the single agents' individual plans. In our setting, the agents have to plan individually and decide autonomously when and how to act. Also, when it comes to action application, there is no predefined notion of agent precedence. This leads to the possibility of *incompatible* plans, and in consequence to the necessity for agents having to *replan* in some cases. While our notion of implicitly coordinated planning explicitly forbids the execution of actions leading to *dead-end situations* (i.e. non-goal states where there is no implicitly coordinated plan for any of the agents), replanning can still lead to *livelocks*. Both the conditions leading to livelocks and individually applicable strategies to avoid them need to be investigated.

References

[1] Thomas Ågotnes, Philippe Balbiani, Hans P. van Ditmarsch & Pablo Seban (2010): *Group announcement logic*. Journal of Applied Logic 8(1), pp. 62–81, doi:10.1016/j.jal.2008.12.002.

[2] Alexandre Albore, Hector Palacios & Hector Geffner (2009): *A Translation-Based Approach to Contingent Planning*. In: Proc. IJCAI 2009, pp. 1623–1628.

[3] Mikkel Birkegaard Andersen (2015): *Towards Theory-of-Mind agents using Automated Planning and Dynamic Epistemic Logic*. Ph.D. thesis, Technical University of Denmark.

[4] Mikkel Birkegaard Andersen, Thomas Bolander & Martin Holm Jensen (2012): *Conditional Epistemic Planning*. In: Proc. JELIA 2012, pp. 94–106, doi:10.1007/978-3-642-33353-8_8.

[5] Mikkel Birkegaard Andersen, Thomas Bolander & Martin Holm Jensen (2015): *Don't Plan for the Unexpected: Planning Based on Plausibility Models*. Logique et Analyse 58(230), doi:10.2143/LEA.230.0.3141807.

[6] Guillaume Aucher & Thomas Bolander (2013): *Undecidability in Epistemic Planning*. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013).

[7] Johan van Benthem, Jan van Eijck, Malvin Gattinger & Kaile Su (2015): *Symbolic Model Checking for Dynamic Epistemic Logic*. In: Proceedings of the 5th International Workshop on Logic, Rationality, and Interaction (LORI 2015), pp. 366–378, doi:10.1007/978-3-662-48561-3_30.

[8] Thomas Bolander & Mikkel Birkegaard Andersen (2011): *Epistemic planning for single and multi-agent systems*. Journal of Applied Non-Classical Logics 21(1), pp. 9–34, doi:10.3166/jancl.21.9-34.

[9] Michael Brenner & Bernhard Nebel (2009): *Continual planning and acting in dynamic multiagent environments*. Autonomous Agents and Multi-Agent Systems 19(3), pp. 297–331, doi:10.1007/s10458-009-9081-1.

[10] Nils Bulling & Wojciech Jamroga (2014): *Comparing variants of strategic ability: how uncertainty and memory influence general properties of games*. Autonomous Agents and Multi-Agent Systems 28(3), pp. 474–518, doi:10.1007/s10458-013-9231-3.

[11] Alessandro Cimatti, Marco Pistore, Marco Roveri & Paolo Traverso (2003): *Weak, Strong, and Strong Cyclic Planning via Symbolic Model Checking*. Artificial Intelligence 147(1–2), pp. 35–84, doi:10.1016/S0004-3702(02)00374-0.

[12] Hans P. van Ditmarsch (2003): *The Russian Cards Problem*. Studia Logica 75(1), pp. 31–62, doi:10.1023/A:1026168632319.

[13] Hans P. van Ditmarsch, Wiebe van der Hoek & Barteld Kooi (2007): *Dynamic Epistemic Logic*. Springer Heidelberg, doi:10.1007/978-1-4020-5839-4.

[14] Hans P. van Ditmarsch, Wiebe van der Hoek, Ron van der Meyden & Ji Ruan (2006): *Model Checking Russian Cards*. Electronic Notes in Theoretical Computer Science 149(2), pp. 105–123, doi:10.1016/j.entcs.2005.07.029.

[15] Edmund H. Durfee (2001): *Distributed Problem Solving and Planning*. In: Proc. ACAI 2001, pp. 118–149.

[16] Jan van Eijck (2004): *Dynamic epistemic modelling*. CWI. Software Engineering [SEN].

[17] Malik Ghallab, Dana S. Nau & Paolo Traverso (2004): *Automated Planning: Theory and Practice*. Morgan Kaufmann.

[18] Wojciech Jamroga (2004): *Strategic Planning through Model Checking of ATL Formulae*. In: ICAISC, pp. 879–884, doi:10.1007/978-3-540-24844-6_136.

[19] Wojciech Jamroga & Thomas Ågotnes (2007): *Constructive knowledge: what agents can achieve under imperfect information*. Journal of Applied Non-Classical Logics 17(4), pp. 423–475, doi:10.3166/jancl.17.423-475.

[20] Wojciech Jamroga & Wiebe van der Hoek (2004): *Agents that Know How to Play*. Fundam. Inform. 63(2–3), pp. 185–219.

[21] Filippos Kominis & Hector Geffner (2015): *Beliefs in Multiagent Planning: From One Agent to Many*. In: *Proc. ICAPS 2015*, pp. 147–155.

[22] Benedikt Löwe, Eric Pacuit & Andreas Witzel (2011): *DEL planning and some tractable cases*. In: *Proc. LORI 2011*, pp. 179–192, doi:10.1007/978-3-642-24130-7_13.

[23] Christian Muise, Vaishak Belle, Paolo Felli, Sheila McIlraith, Tim Miller, Adrian R. Pearce & Liz Sonenberg (2015): *Planning over Multi-Agent Epistemic States: A Classical Planning Approach*. In: *Proc. AAAI 2015*, pp. 3327–3334, doi:10.1.1.684.2394.

[24] Raz Nissim & Ronen I. Brafman (2014): *Distributed Heuristic Forward Search for Multi-agent Planning*. *Journal of Artificial Intelligence Research* 51, pp. 293–332, doi:10.1613/jair.4295.

[25] Héctor Palacios & Hector Geffner (2009): *Compiling Uncertainty Away in Conformant Planning Problems with Bounded Width*. *Journal of Artificial Intelligence Research* 35, pp. 623–675, doi:10.1613/jair.2708.

[26] Ronald P. A. Petrick & Fahiem Bacchus (2002): *A Knowledge-Based Approach to Planning with Incomplete Information and Sensing*. In: *Proc. AIPS 2002*, pp. 212–222, doi:10.1.1.16.7462.

[27] Ronald P. A. Petrick & Fahiem Bacchus (2004): *Extending the Knowledge-Based Approach to Planning with Incomplete Information and Sensing*. In: *Proc. ICAPS 2004*, pp. 2–11.

[28] Ronald P. A. Petrick & Mary Ellen Foster (2013): *Planning for Social Interaction in a Robot Bartender Domain*. In: *Proc. ICAPS 2013*, pp. 389–397.

[29] David Premack & Guy Woodruff (1978): *Does the chimpanzee have a theory of mind?* *Behavioral and Brain Sciences* 1(4), pp. 515–526, doi:10.1017/S0140525X00076512.

[30] Duncan J. Watts & Steven H. Strogatz (1998): *Collective dynamics of 'small-world' networks*. *Nature* 393(6684), pp. 440–442, doi:10.1038/30918.