Report from Dagstuhl Seminar 14032

Planning with Epistemic Goals

Edited by

Thomas Ågotnes[1], Gerhard Lakemeyer[2], Benedikt Löwe[3], and Bernhard Nebel[4]

1 **Universitetet i Bergen, NO,** thomas.agotnes@infomedia.uib.no
2 **RWTH Aachen, DE,** gerhard@kbsg.rwth-aachen.de
3 **Universiteit van Amsterdam, NL and Universität Hamburg, DE,** b.loewe@uva.nl
4 **Universität Freiburg, DE,** nebel@uni-freiburg.de

Abstract

This report documents the outcomes of Dagstuhl Seminar 14032 "Planning with epistemic goals". It brought together the communities of so far relatively separate research areas related to artificial intelligence and logic: automated planning on the one hand, and dynamic logics of interaction on the other. Significant overlap in motivation, theory and methods was discovered, and a good potential for cross fertilization became apparent.

Seminar January 12–15, 2014 – http://www.dagstuhl.de/14032
1998 ACM Subject Classification F.4.1 Mathematical Logic
Keywords and phrases planning, epistemic logic, modal logic
Digital Object Identifier 10.4230/DagRep.4.1.83
Edited in cooperation with Christian Becker-Asano

1 Executive Summary

Thomas Ågotnes
Gerhard Lakemeyer
Benedikt Löwe
Bernhard Nebel

Automatic planning is a subarea of Artificial Intelligence that was initiated in the 70s. The main idea was to develop efficient methods to generate action plans, for example for robot missions. The initial attempts were based on first order logic. However, most approaches quickly adapted simpler logics and focused on search techniques. The recent years have brought a huge advance on scalability by employing smart search techniques such as heuristic search, SAT, BDDs, and other techniques. Currently, planning researchers explore widening the scope of planning tasks and to connect back to logic oriented approaches of describing dynamics such as GOLOG. At the same time, planning researchers strive to capture planning settings that are more challenging than the classical setting. For instance, planning under uncertainty and planning taking into account beliefs are current research topics.

The research area of dynamic logics of interaction is part of the larger field of applied and interactive logic: the use of logical methods in order to formalize procedures in social and communication contexts. The systems are typically based on the semantics of modal logic, and often focus on information (ex)change and the dynamics of knowledge and beliefs.

Paradigmatic examples are public announcement logic and dynamic epistemic logic. One of the main technical features is the incorporation of agency and events into the modal framework as encapsulated by the notion of product update. Recently, some authors have proposed to use the ideas (or, more generally, the methodology) of dynamic approaches to logic for planning.

Epistemic goals, or more generally, goals that have to be expressed in some intensional language (epistemic, doxastic, deontic, others) have been discussed in several papers in the logic community, but are mostly absent from automatic planning. The development of a research community dealing with these goals in planning will require a close interaction between the two involved communities. The main goal of this workshop was to bring the two communities together and develop a vision of the mid-term goals of such a collaboration. In order to facilitate this, the organizers decided to arrange the workshop around work in four groups: after four tutorial lectures by Gerhard Lakemeyer, Hans van Ditmarsch, Thomas Bolander, and Hector Geffner on Monday, the participants were split up into four groups labelled APPL, BENCH, COMP, and LANG. Tuesday was largely reserved for work in the groups and for preparing the group reports included in this report. Tuesday evening also saw a concert in the *Weisser Saal* with François Schwarzentruber playing the piano and Hans van Ditmarsch playing the cello. The final day had some short presentations and a closing discussion.

For the four themes, the organizers had provided some guiding questions, but left the discussion open for the group participants:

APPL *Applying epistemic planning in the real world.* Theme coordinator: Ron Petrick; group participants: Maduka Attamah, Christian Becker-Asano, Martin Holm Jensen, Benedikt Löwe, Sheila McIlraith, Leora Morgenstern, and François Schwarzentruber. Guiding questions: What are promising applications that will convince the outside world to use epistemic planning? Which areas outside of academia could be interested in epistemic planning? How do we get other academic disciplines (such as roboticists) interested in epistemic planning? Can we come up with a concrete research plan for such an application within the next three years?

BENCH *Establishing benchmarks and concrete goals for epistemic planning.* Theme coordinator: Bernhard Nebel; group participants: Carmel Domshlak, Hector Geffner, Malte Helmert, Andreas Herzig, Jörg Hoffmann, Jérôme Lang, and Hans van Ditmarsch. Guiding questions: Can we come up with standardized problems to measure and compare systems for epistemic planning? Which standardized problems could help to calibrate the expressive power of epistemic planning formalism? What currently unsolved problems will serve as milestones and success criteria for the next three to five years?

COMP *Taming the complexity of epistemic planning.* Theme coordinator: Thomas Bolander; group participants: Gerhard Lakemeyer, Yongmei Liu, Robert Mattmüller, Sunil Simon, Jan van Eijck, and Yanjing Wang. Guiding questions: Which aspects of epistemic planning are responsible for the increase of computational complexity? Are there fragments of epistemic planning that allow for an efficient implementation? Can we devise sufficiently expressive planning formalisms that still have acceptable complexity?

LANG *Finding adequate languages for epistemic planning.* Theme coordinator: Thomas Ågotnes; group participants: Guillaume Aucher, Mikkel Birkegaard Andersen, Jens Claßen, Tiago de Lima, Valentin Goranko, and Gabriele Röger. Guiding questions: Which formalisms are adequate to represent epistemic planning problems? Can we devise languages for epistemic planning that are intuitive to understand and use? Can we extend existing plan definition languages with epistemic features? Are the epistemic logics we have sufficiently expressive to serve as a basis for such planning formalisms?

In the final discussion, the participants discussed the immediate future of the interaction between the two fields. One idea was to edit a special issue of the journal *Annals of Mathematics and Artificial Intelligence*, and the seminar organisers are currently in negotiation with the journal editors about that. Thomas Bolander, Hans van Ditmarsch, Jan van Eijck, and R. Ramanujam are planning a follow-up meeting at the Lorentz Center in Leiden in the spring of 2015, and we hope to reconvene with many of the Dagstuhl participants at that meeting.

2 Table of Contents

3 Overview of Talks

3.1 The Situation Calculus and Golog

Gerhard Lakemeyer (RWTH Aachen, DE)

In this tutorial I present the basics of the situation calculus and the action programming language Golog. The situation calculus was invented by John McCarthy as a rich logical language to represent dynamical domains and to reason about action and change. Here I consider the variant introduced by Ray Reiter, as it has the very desirable feature that it comes equipped with a solution to the frame problem in the form of so-called successor state axioms, which determine precisely how the values of fluents change from one situation to another as a result of performing an action. An important benefit of successor state axioms is that they allow us to solve the projection problem, that is, reasoning about what is true after a sequence of actions, by performing regression, which reduces a query about the future to a query about the initial situation. After going over these and other properties of the situation calculus, I then move on to Golog, which has many features from imperative programming such as while-loops, but also constructs allowing for non-deterministic choice. The latter enables a user to encode planning problems as part of a Golog program. In the tutorial I briefly go over the semantics of the various constructs, which are all defined within the situation calculus. I also briefly sketch how knowledge and sensing can be handled in Golog and the situation calculus.

3.2 Epistemic Planning: The DEL approach

Thomas Bolander (Technical University of Denmark – Lyngby, DK)

In my talk I will present one of the possible approaches to planning with epistemic goals: planning based on Dynamic Epistemic Logic (DEL). I will show how planning based on DEL, henceforth called "epistemic planning", generalises classical propositional planning in the most obvious way. Epistemic planning is essentially obtained by generalising both the states and actions of propositional planning to multi-sets of such states and actions – and then add indistinguishability relations for each agent. In my talk I will introduce the general framework of epistemic planning, relate it to classical planning, and discuss complexity results for the plan existence problem. Multi-agent epistemic planning is undecidable, already with 2 agents and no ontic actions. Some relevant fragments have been shown to be decidable, but the quest for fragments of epistemic planning with decent complexities still goes on.

3.3 Merging DEL and ETL for epistemic planning

Yanjing Wang (Peking University – Beijing, CN)

Joint work of Wang, Yanjing; Yanjun Li
Main reference Y. Wang, Y. Li, "Not All Those Who Wander Are Lost: Dynamic Epistemic Reasoning in Navigation," in Proc. of Advances in Modal Logic 2012 (AiML'12), pp. 559-580, College Publications, 2012.
URL http://www.aiml.net/volumes/volume9/Wang-Li.pdf

In this talk, we start with a general approach of axiomatizing dynamic epistemic logics using epistemic temporal axioms, as developed in [1] and [2]. This method does not rely on whether the dynamic epistemic logic in question is reducible to a fragment without the dynamic operators. We identify four important (and meaningful) axioms underlying the standard public announcement logic and action model based dynamic epistemic logic: the invariance of propositional valuation, the definition of executability of actions, no miracles and perfect recall. It turns out that the first two are not necessary in order to obtain a complete logic using our method, and this gives us the technical preparation of a new dynamic epistemic logic on transition systems with a uncertainty set. The crucial idea is to include temporal information in the model but still handle the epistemic updates, in spirit, as in the semantics of DEL.

This approach is carried out initially in [3], and its model is essentially the non-probabilistic model of contingent planning in AI. We completely axiomatize the logic, demonstrate its normal form and show the decidability of the logic. The plan verification can be then turned into a model checking problem of this logic, and the plan existence problem can be turned into a model checking problem of a PDL-like extension of this logic. The advantages of using this modal logic approach to continent planning include the following: more general (epistemic) goals are handled (without increasing complexity), (conditional) plans are specified formally thus relationship (abstraction, refinement, equivalence) between plans can be studied precisely, and this gives us a basic common platform to compare the complexity of different planning problems by restricting/extending the model or the logical language.

References

1 Wang, Y. and Cao, Q. (2013). On axiomatizations of public announcement logic. *Synthese*, 190(1):103–134.
2 Wang, Y. and Aucher, G. (2013). An alternative axiomatization of DEL and its applications. In Proceedings of *IJCAI* 2013:1147–1154.
3 Wang, Y. and Li, Y. (2012). Not all those who wander are lost: dynamic epistemic reasoning in navigation. In *Proceedings of Advances in Modal Logic 2012*, volume 9, pp. 559–580, College Publications.

3.4 Automata Techniques for Epistemic Protocol Synthesis

Guillaume Aucher (INRIA Rennes – Bretagne Atlantique, FR)

In this work we aim at applying automata techniques to problems studied in Dynamic Epistemic Logic, such as epistemic planning. To do so, we first remark that repeatedly executing ad infinitum a propositional event model from an initial epistemic model yields a relational structure that can be finitely represented with automata. This correspondence, together with recent results on uniform strategies, allows us to give an alternative decidability proof of the epistemic planning problem for propositional events, with as by-products accurate upper-bounds on its time complexity, and the possibility to synthesize a finite word automaton that describes the set of all solution plans. In fact, using automata techniques enables us to solve a much more general problem, that we introduce and call epistemic protocol synthesis.

References

1 Aucher, G.; Maubert, B.; Pinchinat, S., *Automata Techniques for Epistemic Protocol Synthesis*, Proc. of the 2nd Int'l Workshop on Strategic Reasoning, Grenoble, France, pp. 97–103, 2014.

3.5 What does it mean to know a number

Jan van Eijck (CWI – Amsterdam, NL)

The talk is about the application of DEL model checking to the analysis of cryptographic security protocols. As a step towards that, I will analyze the question of the title. I will make a proposal for how to represent knowledge of large numbers in Kripke models, and I will use this representation to model a protocol for secret key distribution over an insecure network in DEL. In the conclusion, I will make a connection with epistemic planning.

3.6 Epistemic modal logic with robots and cameras

François Schwarzentruber (IRISA – Rennes, FR)

In this talk, we present a concrete version of epistemic modal logic where agents are located in the plane. Formulas of the language are formulas of epistemic modal logic, that is to say we have construction of the form "$K_a\phi$" meaning that "agent a knows the property ϕ". Atomic propositions are grounded in the sense that they denote physical properties as "agent a sees agent b", "agent a wears a hat", etc. We may add dynamic extensions to the language as public announcements or action models of dynamic epistemic logic. We evaluate formulas in a given world described by the positions of agents in the plane. From a world, we infer a Kripke structure and this Kripke structure is used to give a semantics to formulas. The model checking problem is defined as follows: given a description w of the positions of the agents and a formula phi, does phi hold in w? We show that the model checking problem is decidable [1] but the exact complexity is an open issue. In the case where there is common knowledge of the positions of the agents (but not the angle of view), we show that the model checking problem is PSPACE-complete [2].

References

1 Balbiani, P.; Gasquet, O.; Schwarzentruber, F., *Agents that look at one another*, Logic Journal of the IGPL, 21:3, 438–467, 2013.

2 Gasquet, O.; Goranko, V.; Scharzentruber, F., *Big Brother Logic: Logical modeling and reasoning about agents equipped with surveillance cameras in the plane*, Proc. of AAMAS'2014, 2014.

3.7 Epistemic Protocols for Gossip

Maduka Attamah (University of Liverpool, GB)

The focus of much research in dynamic epistemic logic, and more generally in epistemic and temporal modal logics, is Analysis: given a well-specified input epistemic state, and some well-specified dynamic process, compute the output epistemic state. We focus on Synthesis: given a well-specified input epistemic state, and desirable output, find the process transforming the input into the output. The process found is the epistemic protocol. In this talk we discuss protocol synthesis within a specific epistemic planning problem scenario, namely, the gossip problem.

The gossip problem describes a scenario with a group of N agents where each agent knows a unique secret. The agents can *call* each other, that is a pairwise communication in which the calling pair reveal all the secrets they know to each other. The goal is such that by means of a sequence of calls, all the agents learn the secret of every other agent. Studies in literature include variations of this problem in graph theory and matrix algebra , with results about the number of calls to needed to fully distribute the secrets[1]. We focus on variations of the gossip problem in which an agent communicates with some other agent based on some knowledge (or epistemic) property it has, or on some epistemic property of another agent in the group of agents. Hence we study epistemic protocols for gossip. These kinds of protocols are relevant in many real world peer-to-peer networking applications involving autonomous agents, for which a clear understanding of the underlying information and knowledge dynamics is needed.

In our work [2] we introduce a variant of Dynamic Epistemic Logic (DEL) for describing epistemic gossip protocols, we introduce semantic objects and semantics for interpreting such protocols and give various logical properties of these protocols. We investigate properties such as expected execution length; termination (that is, whether the protocol succeeds in fully distributing all the secrets among the agents); and we develop a framework under which such properties can be described and checked.

References

1. S.M. Hedetniemi, S.H.; Hedetniemi, S.T.; Liestman, A.L., *A survey of gossiping and broadcasting in communication networks*, Networks, 18, 319–349, 1988.
2. Attamah, M.; van Ditmarsch, H.; Grossi, D.; van der Hoek, W., *Knowledge and Gossip*, under submission.

4 Working Groups

4.1 Working Group on "Applications of epistemic planning in the real world" (APPL)

Ron Petrick and Christian Becker-Asano

4.1.1 Epistemic planning applications: theory and practice

One of the key difficulties surrounding the current state of epistemic planning and its ability to be a useful tool for developing real-world applications is the inherent cultural differences that appear to exist between researchers in the formal logic and automated planning communities. On the one hand, the planning community should become more familiar with the variety of formal methods coming from the epistemic logic community, and what they offer to the traditional planning concerns of representation and computation, while on the other hand, the epistemic logic community should look to the representational and algorithmic concerns of the planning community in order to identify what features of formal theories could be useful in practice. Without this deeper understanding of concepts and methods a successful collaboration seems less likely.

Looking beyond the core formal logic and planning communities, it may also be helpful to engage or foster connections with other disciplines where epistemic planning can potentially play a key role:

- robotics, especially in the areas of human-robot interaction and social robotics
- computer gaming and simulation, especially the area of interactive storytelling
- multiagent systems
- cognitive science
- (social) psychology

Engagement with industry may also be possible, however, the wider "usefulness" of epistemic planning may rest on our ability to design, develop, and maintain tools that have the potential to be adopted beyond the core epistemic planning research community.

4.1.2 Relevant features for applied epistemic planning

The following list of key features was identified as most relevant to the goal of applying epistemic planning to real-world applications:

4.1.2.1 High-order reasoning

Epistemic planning becomes increasingly relevant in cases which necessitate some level of nested reasoning (e.g., $K_a K_b P$). However, reasoning with nested knowledge operators becomes more difficult as the depth of such operators increases, with most complex reasoning in everyday situations often limited to at most depth three. Identifying and reconciling the distinction between such situations, and those of classical logical reasoning problems such as the Muddy Children example, is an important difference between theory and practice.

4.1.2.2 Tractable

From a computational complexity perspective, a real-world problem must be modelled in such a way that lends itself to tractable reasoning and execution. This is particularly important if the application is meant to be real-time compatible and involves an aspect of human-robot interaction where response time is a key concern. At the same time, the application scenario needs to be non-trivial from both a formal and practical view, to be of interest to both the logic and planning communities.

4.1.2.3 Multi-agent planning

In the best case, the application scenario should support epistemic planning that involve multiple agents, since such scenarios are more likely to lend themselves to complex situations involving non-trivial epistemic properties. At the same time, with arbitrary groups of agents (e.g., beyond 5 agents) the problem become more difficult from an implementation point of view, especially if all agents are autonomous and have different sets of capabilities. It is also important to note that the case of a single agent should not be completely dismissed, and single agent epistemic planning scenarios may still provide a valuable testbed for both the theoretic and algorithmic approaches.

In addition to these central features, the following features should also be considered.

4.1.2.4 Coalition logics

In order for multiple agents to reach a shared goal the integration and application of coalition logics in epistemic planning seems necessary. This also gives rise to questions of mechanism design and other game theoretic concepts.

4.1.2.5 Distributed vs. centralised planning and coordination of multiple agents

From a distributed systems point of view, questions of synchronisation and knowledge sharing over the network might come into play. This might further complicate the development of robust, practical solutions due to the additional complications of assuming or establishing properties like common knowledge over a network.

4.1.2.6 Knowledge of capability

Modelling first and, especially, higher order knowledge of capability opens up new application scenarios of interest to both core communities. Knowing that someone has the capability of doing a certain action enables new and interesting types of goals in planning which could be used in practical applications.

4.1.2.7 Long-term interaction

The robotics community's increasing interest in long-term interaction with robotic systems provides a potentially useful testbed for the epistemic planning community. In particular, the type of repeated, multi-agent interaction required in such applications would clearly benefit from the availability of a solid framework built on epistemic planning.

4.1.2.8 Adaptability

Any framework that is developed as a joint outcome of the two communities should be generic enough to be easily adaptable to new application scenarios, both within and without the core epistemic planning community.

Table 1 displays some of these relevant features and identifies particular properties of the application design process which are potentially affected by these features.

Table 1 Relevant features and particular properties of the application design process which are potentially affected by these features.

Desired Features				
	Represent.	Reasoning	Comput.	Strategic
Multiple agents	X			
Common/distrib. knowledge	X	X		
High order (e.g. depth 3 nesting)	X			
Complex goals (e.g. nesting negation, etc.)	X			
Knowledge (e.g. capability)	X	X		
Coalitions		X		
Tractability			X	
Decidability			X	
Computability			X	
Long-term interaction				X
Adaptability				X

4.1.3 Applications

A set of possible applications were identified as potential testbeds for epistemic planning, which attempt to look beyond the type of standard toy benchmark problem which is common in many communities:

- Strategy development
  - Narrative generation
  - Strategic game playing
  - Business strategy planning
- Navigating social domains
  - Social robots
  - Conversational agents
  - Tutoring systems
- Security and enforcement
  - Network security
  - International and organised crime detection
- Internet-of-things
- Role playing games
- Algorithmic game theoretic applications

4.1.4 Example

As a specific example we consider an international crime scenario and the type of reasoning that is common in this case.

- Jones is trying to track down the many people who were accomplices in the latest bombing at a Baghdad market.
- What Jones knows:
  - X or Y detonated the bomb.
  - W1 shipped the detonator to X or W2 shipped the detonator to Y
- What Jones needs to know:
  - X or Y?
  - W1 or W2?

- How Jones can find out:
  - Check database to find out the characteristics of bomb which will let Jones know whether X or Y.
  - Visit a chat room to find out whether W1 or W2 has been shipping detonators.
  - Constraint: Jones must not let anyone know that he has found out (knows) that W1 (W2) has shipped detonator

The challenge: Can we develop a language or theory that supports this type of reasoning? Can we apply automated planning to this problem?

4.2 Working Group on "Establishing benchmarks and concrete goals for epistemic planning" (BENCH)

Bernhard Nebel and Hans van Ditmarsch

The goal of this working group was to identify a set of (scalable) examples and benchmarks that could be used to demonstrate the usefulness of epistemic planning, to inspire work on adjusting the expresseiveness of epistemic planning formalisms, and most importantly to inspire the design of new methods and systems that are able to solve epistemic planning problems. In the long run, we hope that our work leads to the creation of a new branch in the international planning competition.

Of course, it would be useful to spell out the benchmarks using a formal logical/planning language. However, for now, we decided to have simple sketches of such problems using natural language.

As epistemic planning problems, we understand here planning problems which involve reasoning about epistemic states, even when the goals of such planning problems are not epistemic. For example, when playing Cluedo, winning the game does not necessarily involve an epistemic goal (similarly for other games). However, for solving such games, reasoning about epistemic states is either necessary or profitable.

4.2.1 Classification / Dimensions

Differently from classical planning, there is a large number of dimensions along which benchmarks could be classified. These dimensions include:

- **Number of agents**: The main consus was that for epistemic planning one would need at least 2 agents, whereby perhaps only one agent is able to act. However, during the plenary discussion also arguments came up for the need of epistemic reasoning in the single-agent case.
- **Cooperative versus adversarial setting**: Scenarios could be that a group of agents tries to achieve a common goal or that each agent might want to achieve its own goal, which might conflict with the goals of the other agents.
- **Centralized versus distributed**: Plans can be global in the sense that they spell out the actions for all the agents (and are generated by a central system) or plans could be generated decentralized by each agent in isolation. As mentioned in the discussion, this difference might vanish when we consider online planners that only determine the next action. Furthermore, a question is how many agents we are able to control; the other agents (e.g. a customer in a Helpdesk setting) choose their actions non-deterministically.

- **Turn-taking versus simultaneous action**: Actions might take place simultaneously or there might be a turn-taking mechanism.
- **Simple epistemic knowledge, higher order knowledge, common knowledge**: The nesting depth of epistemic propositions can be limited to 1, to any fixed n, or it can be unbounded. Furthermore, we may have common knowledge (or even relativized common knowledge).
- **Communication & physical actions**: Actions can be only physical actions, which can be observed by the other agents, or only communicating actions (private or public announcements), or both.
- **Lying vs. only truthful statements**: Most work in DEL assumes that all communications by agents are truthful. However, there are, of course, interesting cases where (potential) lying plays an important role and where the goal can be to uncover lies of one agent.
- **Puzzle-mode versus shallow reasoning**: While all the dimensions mentioned so far can be connected to language features or restrictions in the formulation of the problem, this dimension is not as easy to capture. What is meant here is that an epistemic planning problem might be a difficult to solve puzzle even for humans (such as the muddy children puzzle), while other problems can be solved by any ordinary human quite easily. And we might want to address the latter ones in the beginning.

4.2.2 Benchmarks

The ideas for benchmarks could be classified into two broad sets. The first set consists of general frameworks, where we believe epistemic planning could be applied and where suitable specializations of the general idea might lead to concrete benchmark instances. In other words, there is probably a lot of work necessary to produce concrete benchmarks. The second set is more concrete, but still need significant work in order to produce a set of scalable benchmark problems.

4.2.2.1 General ideas for benchmarks

- (Video) games with joint action – perhaps even only small portions of the games
- Help desk (modeling beliefs of customers)
- Soap opera planning (or narrative planning in general)
- Agents meeting in a grid with partial observability and communication

4.2.2.2 Specific benchmark ideas

- **(Knowledge-based) gossip protocols**, with shortest or longest execution sequences (where longest execution sequences correspond to a very basic form of soap opera, people exchanging secrets until everybody knows everything, drawing out the process for as long as possible):
  Gossip protocols are protocols for peer-to-peer communication in networks, popularly known as 'telephone calls'. In network theory gossip protocols have been investigated in depth since the 1970s, where the paradigm is that a global scheduler assigns calls to pairs of agents. On the assumption that each agent or process has a local state of information, its 'secret', the goal is for all agents to know all secrets as quickly as possible. When calling each other, agents exchange all the secrets they know. For n agents each holding a secret, the minimum is $2n - 4$, and the maximum (on the assumption that actual information growth takes place during a call) is $n(n-1)/2$. There are also stochastic / probabilistic approaches. Now assume that we switch from a global scheduler to agent-based scheduling

(assuming some random selection of the next agent to make a call, simulating everybody rushing to reach the phone to make a call but only one agent succeeding). For four agents a, b, c, d with secrets A, B, C, D, the shortest protocol to distribute all secrets is ab;cd;ac;bd. For that, it is essential that caller c of the second call calls d and not a or b, which would also be informative. Now if we assume that, as c was not involved in the first call, c cannot distinguish a from b and d (the first call could, from c's perspective, have been between a and b, or a and d, or b and d), c has no reason to prefer calling d over a and b. If the second call is ca instead of cd, the shortest sequence to distribute all secrets now has length 5, not the minimum 4 (observe that d, who now only knows his own secret, has to call or be called by three other agents). We can consider knowledge-based gossip protocols where agents call other agents based on their knowledge, e.g.: (i) call an agent whose secret you do not know, or, (ii) call an agent such that you consider it possible to learn a new secret from that agent, or, (iii) call an agent such that you know that you will learn a new secret from that agent, or (iv) call an agent that you consider possible (or likely) to know many secrets already. (For another example, given the first two calls ab;ac, b now can call a again and learn the secret C in that call. So, b may have a reason to call another agent whose secret he already knows. That other agent may have learnt other secrets in the mean time.) Similar consideration abound in network theory, but not from an agent-based planning perspective. Not much is known about synthesizing such knowledge-based gossip protocols. What is there expected execution length? What is the minimum length of a knowledge-based gossip protocol? A wealth of variations investigated in network theory can be similarly modelled for knowledge-based gossip protocols. Postconditions may be general knowledge of all secrets (as above), but also, on the assumption of common knowledge of the protocol, common knowledge of all secrets. Yet other variations consider different agents following different protocols. Who learns all secrets first? What is the best choice of protocol, and against what other protocols executed by other agents? A standard reference on gossip protocols is [1].

- **Extended Wumpus** world with multiple agents, decentralized, with constraints on communication and sensing abilities
- **BW4T**: Blocksworld for teams
- **Epistemic blocksworld** (Thomas Bolander), where blocks have inscriptions only part of which are visible for each agent.
- **Communicating prisoners**: Agents (prisoners), who want to communicate privately without revealing anything to other agents (guards), where there are constraints on when communication can be tapped into
- **Active muddy children**: A variation of the muddy children problem, where the children are allowed to communicate
- **Cluedo**: Here we may want to consider the game in its full complexity as well as restricted scenarios and simplifications.
  Cluedo (for Americans, Clue), is a murder-mystery board game wherein six partying guests are confronted with a dead body, and they are all suspected of the murder. The game board depicts the different rooms of the house wherein the murder is committed, and there are also a number of possible murder weapons. Six suspects (such as Professor Plum), nine rooms, and six possible murder weapons. These options constitute a deck of 21 cards, one of each kind is drawn and are considered the real murderer, murder weapon, and murder room. The other cards are shuffled (again), and distributed to the players. The game consists of moves that allow for the elimination of facts about card ownership, until the first player to guess the murder cards correctly has won. When on the game

board a room is reached by a player, that player may then voice a suspicion, such as 'I think Ms. Scarlett did it, with a knife, in the kitchen'. This question is addressed to another player and interpreted as a request to admit or deny ownership of these cards for that player. If the addressed player doesn't have any of the requested cards, she says so, but if a player holds at least one of the requested cards, she is obliged to show exactly one of those to the requesting player, and to that player only. The four other players cannot see which card has been shown, but of course know that it must have been one of the three. Both denying ownership (i) and showing a card (ii) are epistemic actions, where the action of showing a card has 'interesting' epistemic postconditions, that may also increase the modal complexity of the underlying model. The question about card ownership is successively addresses to all other players until one of them answer the question (with a (i) or (ii) action.) During his turn a player may also make a guess for the murder cards (iii) – this can be done once in the game only, and is not done by saying it aloud but by writing the three cards on a piece of paper and then checking with the three cards on the table (the murder cards) if the guess is correct. If the guess is incorrect, that player has lost and the game continues. (If the guess had been said aloud, then even with an incorrect guess the game would now be over.) A final epistemic action is that of passing on your turn to the next player (iv), which is, on the assumption of perfect rationality, an admission of ignorance of the actual murder cards. For a knowledge analysis it is common to abstract from the aspect of the game board and thus allow questions about any room, and also only to allow correct guesses of the murder cards, that is, you win when you know what the murder cards are. There are a number of implementations of Cluedo and a Dagstuhl 'Planning with Epistemic Goals' participant currently working on one is Tiago de Lima, CRIL, Lens.

- **PIT**: Another game. Pit is another example of a card games with logical dynamics involving (strict) subgroups of all players. In the Pit game (for trading pit – it's a market simulation card game) the players try to corner the market in coffee, wheat, oranges, or a number of other commodities, and it is like the 'Family Game' in that each of these commodities are distributed over the players in the form of cards, and the first player to gather a full suit of cards (i.e., nine cards) of any commodity, wins. The game moves consist of two players exchanging cards. (In other words, unlike Cluedo, it is a planning problem with information change as well as ontic/factual change.) This goes as follows. A requirement to exchange cards is that they are of the same suit. Players shout the number of cards they wish to exchange, simultaneously, and two playing shouting the same number may then make a change. For example, John has 2 apples, 3 oranges, and some other cards, Mary has 2 oranges and yet other cards. John could have shouted 1, 2, or 3 (changing some but not all of the cards of the same suit is also allowed), but goes for 2, and Mary goes for 2 as well. Shout, shout, ... And they make an exchange. John now has 5 orange cards! Still not 9, but better than before. The exchange action is somewhat similar to the move of showing a card in Cluedo: two players gain subgroup common knowledge, in this case, of the new ownership of the exchanged cards. The other players only learn that two players each have at least two cards of the same suit. This rules out some card distributions. This exchanging of cards continues until somebody gathers his suit of nine cards. This can, in principle, go on forever: it is an extensive game of imperfect information, with infinite (but highly repetitive, there is much symmetry) game tree branches. For more information, see [2, 3].

References

1 Hedetniemi, S.M.; Hedetniemi, S.T.; Liestman, A.L., *A survey of gossiping and broadcasting in communication networks*, Networks, 18:4, 319–349, 1988.
2 van Ditmarsch, H., *The Logic of Pit*, Synthese, 149:2, 343–374, 2006.
3 van Ditmarsch, H., *Some game theory of Pit*, Proc. of 8th PRICAI, LNAI 3157, 946–947, 2004.

4.3 Working Group on "Planning with Epistemic Goals: Complexity of the task" (COMP)

Jan van Eijck and Thomas Bolander

4.3.1 Definition of 'Planning with Epistemic Goals'

Paradigm for generic planning task: find your way in a grid towards a specified goal. Paradigm for single-agent generic epistemic planning task: find your way in a grid towards a specified goal, given that your initial position is unknown. Paradigm cases for multi-agent generic epistemic planning tasks: depend on the nature of the other agents (co-operative, non-cooperative, mixed), but can also be modelled as navigation tasks in grids.

Different kind of goals: find your goal in the worst case in the minimum number of steps (optimal planning vs fig).

Background assumption in any planning task is some kind of closed world assumption: the agent knows (or the agents know) the limited repertoire of things that can happen. In an epistemic context, the closed world assumption and your goal without violating any constraints (satisficing planning will need a more sophisticated formulation than that of Ray Raiter ("Whatever is not currently known is false"). Instead of this: "Whatever is not currently known about ... satisfies the following constraints ... ."

This is closely related to the classical frame problem in AI: how to model the fact that things in the environment do not change arbitrarily? In an epistemic context, this becomes: how to model the fact that agents know that things in the environment do not change arbitrarily?

4.3.2 Looking for fragments with decent complexity

4.3.2.1 Brief Reminder

The plan existence problem for planning with epistemic goals has, more often than not, an unmanageable complexity. Already single-agent S5 planning with partial observability is 2EXP-complete, and multi-agent planning is undecidable (even with only 2 agents and no ontic actions). It is therefore essential to consider various restrictions that allow the complexity of epistemic planning to be controlled, so that it can become practically feasible. Below we will consider restrictions obtained by considering fragments of epistemic planning: restrictions on representation, reasoning, plan types.

4.3.2.2 Representational Constraints

A first way to constrain the planning fragments is by putting constraints on the representation used to specify actions, goals, preconditions and postconditions.

- Possible restrictions on action types: public/private/partially observable announcements, public/private/partially observable propositional assignments, sensing actions (specified in simple terms), propositional actions, actions that don't produce exponential growth of the input epistemic model (**safe** actions).
- Possible restrictions on goal types: goals of limited depth, only positive goals.
- Another possible restriction is by putting a bound on the epistemic depth of reasoning, e.g., by only considering models up to depth k.
- Finally, it is possible to impose frame conditions: S5, K45, KD45, and so on.

Question: What action types do we even want to consider? Relates to (APPL) and (BENCH).

4.3.2.3 Limited reasoning

- Limited models of belief.
- Alternative logical bases (e.g. 3-valued logic).

4.3.2.4 Limited plan types

- E.g. existence of plans of *polynomially bounded length.*
- *Protocol restrictions* (what actions are allowed when).
  Some examples: Don't allow actions increasing depth beyond . . . Do not allow exact same action twice. Make announcements whenever depth of ignorance is becoming too large.

4.3.3 Pragmatic Approaches to Deal with High Complexity

How to get good practical running times despite high complexity of the general task?

- Compact representations (OBDDs, symmetry reduced models, . . .)
- Heuristics (delete-relaxation, abstraction, assumption of full observability, pruning techniques/preferred operators, . . .)
- Online planning, replanning
- Learning techniques, sampling (UCT, . . .)
- Alternative encodings/compilations (SAT, QBF, translation to classical planning, . . .)
- Domain-dependent planning (HTNs, Golog, . . .)
- Search approaches: forward search (with or without backtracking), regression (backwards from the goal).

4.3.4 Methodological Choices

	qualitative	quantitative
logical language based		
without logical language		

- Logical Approach vs Non-linguistic Approach: do we have a formal language to specify the goal?
- Quantitative methods versus qualitative methods: do we use probability or not?
- Markov decision processes versus conformant/contingent planning (all the logical approaches).
- Logical approaches: syntactic vs semantic.
  - Does the logical language allow talk about probabilities? (E.g., DEL vs Probabilistic DEL)

  - Syntactic approach: Knowledge bases as sets of formulas. In this category: situation calculus.
  - Semantic approach: satisfiability checking and model checking. In the model checking category: DEL update with result model checking.
- Offline planning vs online planning. Online planning may or may not use probabilistic methods. All non-deterministic planning is most often implemented as online planning, because of the combinatorial explosion of action outcomes (except conformant planning which is non-deterministic planning without any observability, and hence no feedback from actions to guide replanning).
- Navigation tasks with uncertainty about the location of the agent in a grid seem to call for online planning: plan consists of a repertoire of actions to carry out, and the plan evolves on the basis of the result of these actions.

4.3.5 A First Research Goal

An obvious first item on the list of research goals is the development of a uniform framework to list, classify and compare epistemic planning problems. A possible approach: fit all problems in the same mould by transforming them all into model checking problems using some expressive language L. A candidate language for this could be epistemic PDL.

Next, the problems can be classified according to criteria like the following:
- Number of agents, nature of their observational powers
- Properties of the models: classification according to deterministic or not, but also: knowledge or belief models, etc.
- Language to specify the preconditions, postconditions and goal(s): fragments of L.
- Format of the plan: just a sequence of actions, or conditionals allowed, or even loops? How does the language for preconditions, postconditions and goals figure in the plan format?

On the basis of this classification, complexity can be properly compared.

4.4 Working Group on "Languages for Epistemic Planning" (LANG)

Thomas Ågotnes

4.4.1 Epistemic planning in two communities

Language plays at least three roles in (epistemic) planning:
1. Describing a domain, in terms of actions and their epistemic pre- and postconditions.
2. Describing an initial (epistemic) state and goal condition. A planning domain together with an initial state and goal condition defines a particular instance of a planning problem to be solved.
3. Describing the result of planning: plans.

This gives us at least two variants of the epistemic planning problem. In type 1 epistemic planning the actions have epistemic pre- and postconditions, but the goal is not epistemic. In type 2 epistemic planning also the goal is epistemic. For planning algorithms this distinction does not seem to be important.

The research goals and cultures of the planning and logic communities differ significantly. Logic research focuses on expressive power and considers a wide range of sometimes complex phenomena; in planning simplicity and computational tractability are important. It highly depends on the requirements of the planning competition (IPC) which language features are widely supported by planning systems.

Planning problems are usually modelled as single-agent problems, rather than multi-agent. There are arguments that single-agent epistemic planning is not interesting, but it appears that there are counter-examples to this (for example, a robot could achieve the goal of knowing whether the vase is broken by dropping it, which is counterintuitive). For the multi-agent case, again, there are several variants. In the simplest variant only one agent can act but these actions can possibly affect the beliefs of other agents. His goals might involve epistemic properties of several agents. In more complex scenarios more than one agent can act.

4.4.2 Application Example

Devising epistemic planning languages heavily depends on the kind of applications that one wants to support. In order to decide on which features such a language would have to support, one would have to identify case studies of application scenarios where epistemic planning would be useful. Suppose a domestic robot in an office environment with several people (i.e. other agents) in it. The goal of the agent is to get to know which and how many people there are in the building, introduce itself to them, and achieve that everybody knows (common knowledge?) that lunch will be at 12:00 in the dining hall.

- is this sufficient as an interesting scenario?
- what are the kinds of epistemic goals that we should be able to represent?
- about what parts of the representation there could be possibly uncertainty (planning agent's knowledge, other agent's knowledge, preconditions and effects of actions,...)?
- how to represent/encode sensing, communication etc.?

Multi-agent planning is particularly interesting in adversarial scenarios, which the above is not. Alternatively, a surveillance/security robot could be considered that has to deal with intruders. Moreover, games (General Game Playing?) could be an interesting application area.

4.4.3 How can epistemic planning benefit from epistemic logic research?

A systematic approach: logic provides a systematic and generic framework which allows to compare and relate more easily the impact of the different parameters chosen (time dependencies between events, epistemic goals, adding probability,...) by planning researchers on the properties of a planning scenario (memoryless, non-deterministic effects, conditional, looping) as well as on the computational complexity of solving a planning task.

In epistemic logic a number of epistemic concepts that are more sophisticated than individual knowledge of propositions have been developed and studied:

- Higher-level knowledge
- Common knowledge, distributed knowledge and other types of group knowledge

Logic can also provide a more refined representation of uncertainty by means of:

- Probabilities (also interesting in the single-agent case)
- Qualitative representation: plausibilities, possibilities, rankings,... (also interesting in the single-agent case)

Logic studies phenomena which are not yet addressed in the planning community such as awareness.

All these above features can be used to increase the applicability of planning formalisms.

Moreover, the omniscience problem of epistemic logic is likely to arise in reformulations in logic of planning problems. It is likely that one of the large number of solutions dealing with the omniscience problem in epistemic logic will be applicable to planning problems.

These concepts are relevant both for expressing pre- and post-conditions of actions and epistemic goals. The epistemic logic community has developed a theoretically well-founded foundation for these concepts. Their computational properties are also well understood. Even though these concepts have not been considered in existing case studies, the planning community doesn't necessarily know what can be useful!

How does one express goals about things the agent is not aware of? And this is interesting even in the single-agent setting! E.g., the Mars rover must learn about all new green creatures it finds out there! How is finding out what others are trying to achieve expressed as a goal?

Logic has also developed a number of model theoretical tools that can possibly be applied to planning. An example is techniques for comparing models, which can be applied to characterise when planning problem descriptions are the same. It is well known that planning is sensitive to framing, i.e., representation of the problem.

One challenge is that classical planning formalisms are based on states as variable assignments. From a theoretical perspective, this is not expressive enough to model all epistemic multi-agent settings.

4.4.4 Some research challenges

- Find case studies with epistemic goals involving complex epistemic concepts (higher-level knowledge, common knowledge, etc.).
- Can existing planning description languages (e.g. PDDL, GDL-II) be extended with useful epistemic features in a natural way? (For instance, single-agent S5 DEL planning can implement NPDDL. What multi-agent features can be added to NPDDL? Can these be implemented in multi-agent S5 DEL?)
- Can these extended languages be implemented in epistemic planning frameworks (DEL-based, SitCalc-based, EventCalc-based)? What is the corresponding expressivity and/or complexity?
- How can we define an appropriate language for action models that would allow an easy specification of actions for planning problems?
- How can the expressive power of planning languages be characterised using the epistemic logic framework?
- How can the concepts and methods dealing with succinctness, compactness and elaboration tolerance from the planning community be imported within the epistemic logic framework?
- Can logical techniques for comparing models (simulations, bisimulations, model comparison games) be used to answer the question of when two planning problem descriptions are the same?
- Can planning formalisms be augmented by external "black-box" epistemic reasoning?
- A 'strong' plan guarantees that it will always achieve the goal, while a 'weak'-plan just says that it is possible. Is a logic of degrees of belief a suitable formalism for grading guarantees, as in 'this plan will achieve the goal most of the time', or 'I believe (though I do not know) that this plan will achieve the goal'? Is the complexity of synthesising a plan that is 'believed to work' better or worse than synthesising a strong or weak plan?
- What is the exact and formal connection between the knowledge-based programs of the epistemic logic community and the plans of the planning community?

Participants

- Thomas Ågotnes
University of Bergen, NO
- Maduka Attamah
University of Liverpool, GB
- Guillaume Aucher
INRIA Rennes – Bretagne Atlantique, FR
- Christian Becker-Asano
Universität Freiburg, DE
- Mikkel Birkegaard Andersen
Technical University of Denmark – Lyngby, DK
- Thomas Bolander
Technical University of Denmark – Lyngby, DK
- Jens Claßen
RWTH Aachen, DE
- Tiago de Lima
Artois University – Lens, FR
- Carmel Domshlak
Technion, IL
- Hector Geffner
UPF – Barcelona, ES
- Valentin Goranko
Technical University of Denmark – Lyngby, DK
- Malte Helmert
Universität Basel, CH
- Andreas Herzig
Paul Sabatier University – Toulouse, FR
- Jörg Hoffmann
Universität des Saarlandes, DE
- Martin Holm Jensen
Technical University of Denmark – Lyngby, DK
- Gerhard Lakemeyer
RWTH Aachen, DE
- Jerome Lang
University Paris-Dauphine, FR
- Yongmei Liu
Sun Yat-sen University – Guangzhou, CN
- Benedikt Löwe
University of Amsterdam, NL
- Robert Mattmüller
Universität Freiburg, DE
- Sheila McIlraith
University of Toronto, CA
- Leora Morgenstern
New York University, US
- Bernhard Nebel
Universität Freiburg, DE
- Ron Petrick
University of Edinburgh, GB
- Gabriele Röger
Universität Basel, CH
- Francois Schwarzentruber
IRISA – Rennes, FR
- Sunil Easaw Simon
Indian Institute of Technology – Kanpur, IN
- Hans Van Ditmarsch
LORIA – Nancy, FR
- Jan van Eijck
CWI – Amsterdam, NL
- Yanjing Wang
Peking University – Beijing, CN