Knowledge and Planning in an Action-Based Multi-Agent Framework: A Case Study

Bradley Bart[1], James P. Delgrande[1], and Oliver Schulte[2]

[1] School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6, {bbart,jim}@cs.sfu.ca

[2] Department of Computing Science , University of Alberta, Edmonton, Alberta Canada T6G 2E1, oschulte@cs.ualberta.ca

Abstract. The situation calculus is a logical formalism that has been extensively developed for planning. We apply the formalism in a complex multi-agent domain, modelled on the game of Clue. We find that the situation calculus, with suitable extensions, supplies a unified representation of (1) the interaction protocol, or structure of the game, (2) the dynamics of the knowledge and common knowledge of the agents, and (3) principles of strategic planning.

1 Introduction

The situation calculus is a logical formalism originally developed for planning by a single agent but more recently extended to deal with multiple agents and knowledge. In this paper we use a variant of the game of Clue as a testbed for gauging the power of the situation calculus in an epistemic, multi-agent setting. This has the potential to contribute to several areas of AI, such as the design of intelligent agents, game playing, and the formalism of the situation calculus itself. The situation calculus provides a general language for specifying interactions of a software agent; it can also be used to represent an agent's reasoning. Thus the situation calculus provides an integrated description of the action capabilities of agents and their reasoning and decision-making mechanisms. Similarly, in game playing the situation formalism can represent the rules of the game as well as knowledge about agents' strategies. Conversely, the connection with games opens up the possibility of applying efficient algorithms from games research for finding optimal strategies in multi-agent planning problems.

This paper focuses on issues concerning multi-agent interactions in the situation calculus. A novel aspect of this work is that we deal with knowledge that is *common* to a group of agents. We address these issues in a variant of the game of Clue, described below. Clue is a game in which the agents—players—have to discover the state of the world, rather than change it. We use the situation calculus to represent three aspects of the game:

1. The rules—what players can do when.
2. Information—what the players know at various stages of the game, including (objective) knowledge of the domain together with knowledge opf other agent's knowledge.

3. Planning—how they can exploit that knowledge to find strategic plans for playing the game.

Most of the paper deals with the first two aspects. We found that the situation calculus is a remarkably natural formalism for describing a game structure. For representing the knowledge of the players during the game we employ an epistemic extension of the situation calculus that axiomatizes a knowledge fluent [**?**]. We require several extensions beyond the single-agent epistemic version of the situation calculus. First, we need an agent parameter for knowledge, to distinguish whose knowledge is referred to. Secondly, strategic reasoning involves an agent's reasoning about another agent's knowledge, as well as *common knowledge*.

We concentrate on a variant of Clue here, called MYST. The next section introduces Clue and MYST, and the situation calculus. The third section develops our axiomatisation of the game, while the fourth section addresses reasoning issues. We conclude with a short discussion. Further details are found in [**?**].

2 Background

Clue and MYST: We ignore those aspects of Clue that are irrelevant to the general problems of knowledge representation and planning. The premise of Clue is that there has been a murder; it is each player's goal to determine the murderer, weapon, and location of the murder. Each suspect, and possible weapon and location, are represented by a card. Initially the cards are divided into their three sorts (suspect, weapon, location), and from each sort one card is randomly selected and hidden. These three cards determine the details of the crime. The remaining cards are dealt to the players. At the outset a player sees only her own hand, and thus knows what cards she has been dealt, but not what the other players have received. Each player in turn asks one of the other players about a suspect, a weapon and a room. If the (queried) player has one of the queried cards, she shows it to the asker. The asker then knows that the player has that card; the other players know only that the (showing) player has *one* of the three cards. A player may guess the identity of the hidden cards at the end of their turn and, if correct, they win the game. The three hidden cards represent the state of the world. The joint knowledge of the players is sufficient to determine this information. However each player's goal is to learn the state of the world before the others do. Thus the game is of the same flavour as the "muddy children problem" [**?**], although more subtle and (we feel) interesting.

We reformulate the game of Clue as a simpler game that we call "MYST" (for "mystery"). In MYST there is a finite set of cards, but without the three sorts in Clue. There are m cards hidden in a "mystery pile" and n are given to each of p players. Hence there is a total of $k = m + (n \times p)$ cards. On their turn, a player asks a question about q cards of the form, "Do you have one of cards: $c_1, \ldots, c_q$?" This player is called the "poser". If the next player has one of these cards, they (privately) show the poser the card and the poser's turn is over. If the answer is "no", then the next player in turn is presented with the same query. After asking his question, a player may guess the contents of the mystery pile. If correct, he wins the game; otherwise, the player is relegated to answering posed queries only. The game ends if a player determines the contents of the mystery pile or if all players have been eliminated by unsuccessful guesses.

The Situation Calculus: The intuition behind the situation calculus is that the world persists in one state until an *action* is performed that changes it to a new state. Time is discrete, one action occurs at a time, time durations do not matter, and actions are irreducible entities. *Actions* are conceptualised as objects in the universe of discourse, as are *states* of the world. Hence, states and actions are *reified*. That is, the action of, for example, moving block a from block b to block c is an object.

The constant s_0 refers to the initial state, and $do(A, s)$ is the state resulting from doing action A in state s. Thus $do(stack(a, b), s_0)$ is the state resulting from a *stack* action performed on a and b in situation s_0. The fact that, after performing the stack action, "a is on b" could be represented by $on(a, b, do(stack(a, b), s_0))$. Time-varying predicates, such as *on*, are referred to as *fluents*. Actions have *preconditions* specifying the conditions under which an action can be performed, and *successor state axioms* giving the effects of an action. The predicate $Poss(A, s)$ is used by convention to "collect" the preconditions for action A in situation s. So for *stack* we can express that the preconditions are (1) the hand is holding the block to be stacked and the block to be stacked onto has a clear top:

$$Poss(stack(X, Y), s) \leftrightarrow inhand(X, s) \wedge clear(Y, s).$$

The fluent $on(X, Y, s)$ is true in a state resulting from X being stacked on Y so long as the stack action was possible:

$$Poss(stack(X, Y), s) \rightarrow on(X, Y, do(stack(X, Y), s)).$$

The only other time that an *on* is true in a non-initial state is when it was true in the previous state, and was not undone by an action:

$$Poss(A, s) \wedge A \neq unstack(X, Y) \wedge on(X, Y, s). \rightarrow on(X, Y, do(A, s)).$$

This last axiom is called a *frame axiom*, and specifies what remains unchanged during an action.

Hayes and McCarthy [**?**] originally proposed the situation calculus; we use the version from [**?**], making use of the formalisation of knowledge in [**?**], with variants that we describe later. A multiple-agent version of the situation calculus is described in [**?**]. There, information exchanges are modelled via "send" and "receive" commands. Here in contrast we axiomatise operations that result in a change of knowledge for an agent. Thus for example, if an agent *shows* another a card, then the second *knows* the value of the card.

3 Representing MYST in the Situation Calculus

In this section, we formalize the game of MYST by specifying a set of axioms in the language of the situation calculus. Of particular interest is the knowledge fluent that describes what players know at various stages of the game.

3.1 Situation Calculus Terms Used in the Formalisation of MYST

Constants: We assume enough arithmetic to define a sort *natural_num* with constants $0, 1, .., n, ...$ to have their intended denotation. We extend the situation calculus with two more sorts: the sort *player* and the sort *card*. We introduce the constants described in the following table.

Constant Symbol(s)	Sort	Meaning
p	*natural_num*	total number of players
k	*natural_num*	total number of cards
n	*natural_num*	number of cards in each player's hand
m	*natural_num*	number of cards in mystery pile
q	*natural_num*	number of cards in a query
$p_1, .., p_p$	*player*	p_i denotes player i.
$c_1, ..., c_k$	*card*	c_i denotes card i

To encode the fact that we deal with a finite set of distinct players and cards, we adopt a unique names assumption (UNA) and domain closure assumption (DCA) with respect to these sorts. That is, for the set of players we add axioms

$$\begin{array}{l} \mathrm{UNA}_P : (p_i \neq p_j) \quad \text{for all} \quad 1 \leq i \neq j \leq p. \\ \mathrm{DCA}_P : \forall x.\ \mathit{player}(x) \equiv (x = p_1 \vee \cdots \vee x = p_p). \end{array}$$

Analogous axioms (UNA_C, DCA_C) are adopted for the set of cards. We have a further constant s_0 to denote the initial situation in MYST, which obtains immediately after the cards have been dealt.

Since the above predicates conceptually define a finite set of players (and cards), we adopt a set theoretic notation for players (and cards). Adopting a set notation—which we could embed in first-order logic—will make the axiomatisation neater and the language more mathematically familiar. Henceforth we will use the following notation

$C := \{c_1, ..., c_k\}$ the set of all cards

$P := \{1, ..., p\}$ the set of all players (denoted by integers).

Variables: We introduce variables ranging over components of MYST. We need two more sorts: *set_cards* for a set of cards, and *set_players* for a set of players. We will use variables as follows.

Symbol	Meaning
i, j	players ($i, j \in P$);
	typically i is the poser and j the responder
c_x	single card ($c_x \in C$)
G	subset of players ($G \subseteq P$)
Q	set of cards in a question ($Q \subseteq C$)
M	set of cards in a guess about the mystery pile ($M \subseteq C$)
C_j	set of cards held by player j ($C_j \subseteq C$)
C_0	set of cards in mystery pile ($C_0 \subseteq C$)
Σ	generic set of cards ($\Sigma \subseteq C$)
a	an action
s	a situation

We will not explicitly list "type-checking" predicates to ensure that c_x is a card (for instance).

Actions: The following is the list of action functions and their informal descriptions. The sequence in which actions may occur is defined by the predicate $Poss(a, s)$ below. Note that the first argument always represents the player performing the action.

Action function symbol	Meaning
$asks(i, Q)$	Player i asks question Q
$no(j)$	Player j says *no* to question Q
$yes(j)$	Player j says *yes* to question Q
$shows(j, i, c_x)$	Player j shows card $c_x \in Q \cap C_j$ to player i
$guess(i, M)$	Player i guesses that $C_0 = M$
$noguess(i)$	Player i makes no guess
$endturn(i)$	Player i ends his turn

Fluents: The following is a list of fluents and their informal descriptions. The evaluation of the fluents will depend on the situation s. Their truth values may change, according to successor state axioms.

Fluents Describing the Location of Cards:

$H(i, c_x, s)$: Player i holds card c_x.

$H(0, c_x, s)$: The mystery pile holds card c_x.

Fluents Describing Knowledge:

$Know(i, \phi, s)$: Player i knows ϕ.

$C(G, \phi, s)$: ϕ is common knowledge for all players in $G \subseteq P$.

$C(G, \phi, s)$ has the interpretation that, not only do all the players in G know that ϕ, but every player in G knows that the others know this, that they know that each knows this, and so on. There are well-known difficulties with axiomatizing a common knowledge operator, and well-known solutions as well (cf.[**?**]). We don't address these issues, but simply assume a language with a common knowledge modal operator.

Fluents Describing the State of the Game:

$In(i, s)$: Player i has not yet been defeated due to a wrong guess.
$Question(Q, s)$: Question Q was the most recently asked question.
$Gameover(s)$: The game is over.

Without going into the details, we may assume the presence of axioms that ensure that at most one query is asked per situation, that is, that $Question(Q, s)$ holds for at most one query Q.

Fluents Describing the Turn Order and Phases:

$Turn(i, s)$: It is player i's turn.
$AnsTurn(j, s)$: It is player j's turn to answer the question.

As with queries, we assume the presence of axioms that ensure that it is exactly one player's turn and exactly one player's "answer turn".

Fluents Describing the Phases:

$AskPhase(s)$: It is the ask phase.

Similarly we have fluents for the answer phase ($AnsPhase(s)$), show phase ($ShowPhase(s)$), guess phase ($GuessPhase(s)$), and end phase ($EndPhase(s)$). Any situation s is in exactly one of these phases; we assume axioms that enforce this specification.

3.2 Axioms

The Initial Situation s_0: Different initial situations are possible depending on the initial random distribution of the cards. [**?**, **?**] modify the situation calculus to allow different initial situations by defining a predicate $K_0(s)$ that applies to situations that might be initial ones for all the agent knows. Our approach is different but equivalent: We use the single constant s_0 to refer to whatever initial situation results from dealing the cards, and represent the players' uncertainty by describing what fluents they do and do not know to hold in the initial situation. The following table of *initialization axioms* describes those fluent values common at s_0 in all games.

Initialization Axiom	Meaning
$\forall i.In(i, s_0)$	No player has been eliminated
$\forall Q.\neg Question(Q, s_0)$	No one has asked a question
$AskPhase(s_0)$, $Turn(1, s_0)$	Player 1 is in the AskPhase of her turn
$\neg Gameover(s_0)$	The game is not over

The cards C are partitioned among the players and the mystery pile. The following axioms are the *partition axioms* for C. Here and elsewhere, free variables are understood to be universally quantified.

Exclusiveness $H(i, c_x, s_0) \rightarrow \forall j \neq i.\neg H(j, c_x, s_0)$.

If player i holds card c_x, then no other player j (or the mystery pile) holds c_x. If the mystery pile holds card c_x, then c_x is not held by any player.

Exhaustiveness $\bigvee_{i=0}^{p} H(i, c_x, s_0)$.

Every card is held by at least one player (or the mystery pile).

Set Size for Players(SSA)

$\forall i \in \{1..p\}.\exists \Sigma.|\Sigma| = n \wedge (\forall x.H(i, c_x, s_0) \Leftrightarrow c_x \in \Sigma)$.

For player i, there is a set of n cards containing just the cards held by i.

Set Size for the Mystery Pile

$\exists \Sigma.|\Sigma| = m \wedge (\forall c_x.H(0, c_x, s_0) \iff c_x \in \Sigma)$.

There is a set of m cards containing just the cards in the mystery pile.

Preconditions: It is straightforward to define the preconditions of actions in terms of the fluents. We do not have space to give all the definitions in detail; instead, we specify the preconditions for the *asks* action as an example—the other preconditions are analogous. Player i can ask a question iff

1. it is her turn and
2. the game is in the AskPhase.

Thus we have

$$Poss(asks(i, Q), s) \equiv Turn(i, s) \wedge AskPhase(s).$$

Successor State Axioms: We next describe the successor state axioms for the fluents. We begin with the *card holding fluent.*

The cards held by the players do not change over the course of the game.

$$H(i, c_x, do(a, s)) \equiv H(i, c_x, s) \qquad \text{for} \quad i \in \{0..p\}.$$

The fluent H is independent of the situation argument, and so we abbreviate $H(i, c_x, s)$ by $H(i, c_x)$. The fact that the card holdings are the same from situation to situation formally captures the fact that the world remains "static" as the game continues, so that players are not reasoning about changes in the world, but only about increasing information about a fixed but unknown constellation.

Next, we represent *turn taking*. Let $before(i) = i - 1$ if $i > 1$, and $before(1) = p$. Then we have the following axiom. A player's turn does not change until the previous player has taken the *endturn* action; the previous player is given by the *before* function.

$$Turn(i, do(a, s)) \equiv Turn(before(i), s) \wedge a = endturn((before(i)) \vee \\ Turn(i, s) \wedge \neg(a = endturn(i))$$

Other axioms describe the other fluents; we omit the details.

Axioms for Knowledge in MYST: We conceive of the players as perfect reasoners. Every player knows all tautologies and is able to derive all consequences of a set of formulas. As well, every player knows all the rules and axioms; see [**?**] for a full characterization. Although these assumptions do not do justice to the limitations of human and computational players, it makes the analysis of strategies mathematically easier.

Game theorists distinguish broad classes of games according to their epistemic structure. We locate our discussion of knowledge in MYST in these general game-theoretic terms; this will give an indication of the size of the class of multi-agent interactions that falls within our analysis. We shall give informal descriptions of the game-theoretic concepts, with a fairly precise rendering of the concept in terms of knowledge fluents for MYST. Game theory texts give precise definitions in game-theoretic terms; see for example [**?**].

A game has *complete information* if the rules of the game are common knowledge among the players. This is indeed the case for MYST; in the situation calculus, we can capture the complete information by stipulating that all the axioms describing the game structure is common knowledge in every situation. To illustrate, we have that $C(P, In(i, s_0), s_0)$ holds for $p \in \{1..p\}$ —it is common knowledge that at the beginning every player is in the game. A game has *perfect information* just in case every player knows the entire history of the game when it is his turn to move. Chess is a game of perfect information; MYST is *not*. For example, players don't know the entire initial distribution of cards, which is part of the history of the game. A game features *perfect recall* if no player forgets what she once knew or did. We express perfect recall by stipulating that once a player knows a fact in a situation, she continues to know it. Thus the general class of games for which something like our axiomatization should be adequate includes the class of games of complete, imperfect information with perfect recall.

The fluent $Know(i, \phi, s)$ expresses that player i knows that ϕ in situation s.[1] First, the players know which cards they hold in the initial situation s_0.

Axiom 1 (Knowledge Initialization) $Know(i, H(i, c_x, s_0), s_0)$.

Now for the knowledge successor state axioms. Since we assume that the players have perfect recall, we stipulate that knowledge once gained is not lost. Formally, let ϕ be a nonepistemic fluent that does not contain a $Know$ fluent. The case of special interest to the players is the fluent $H(i, c_x)$ (player i holds card x). The next axiom says that knowledge about ϕ is *permanent* in the sense that once gained, it is never lost later.

$$\begin{array}{c} \forall i, s, s'.(s' \sqsubseteq s \wedge Know(i, \phi, s')) \ \rightarrow Know(i, \phi, s) \\ \forall i, s, s'.(s' \sqsubseteq s \wedge Know(i, \neg\phi, s')) \ \rightarrow \ Know(i, \neg\phi, s) \end{array} \tag{1}$$

Inductively, it can be seen that knowing any of the knowledge of the form (1) is also permanent, and so on. Therefore Axiom (1) holds for the common knowledge fluent C as well as $Know$. Most of the reasoning about strategy rests on common knowledge between agents, that is, on the C fluent, rather than the separate knowledge of the agents expressed by the $Know$ fluent.

Players obtain new knowledge only when one player shows a card to another.

[1] See [**?**, **?**] for details. Suffice it to note that $Know$ is defined in first-order logic by explicitly axiomatising an (equivalent of an) accessibility relation [**?**].

Axiom 2 $do(shows(j, i, c_x), s) \rightarrow C(\{i, j\}, do(shows(j, i, c_x), s), s)$.

Thus when player j shows a card to player i, it is common knowledge between them that this action took place. Note that it is then also common knowledge between i and j that player j holds card c_x. For one of the preconditions of $shows(j, i, c_x)$ is that j holds card c_x, and since the preconditions are common knowledge between the players, it is common knowledge that $(do(shows(j, i, c_x), s) \rightarrow holds(j, c_x, s)$.

When player j shows player i a card, it becomes common knowledge among the other players that j has at least one of the cards mentioned in i's query, although the other players won't necessarily know which card. Our axiomatization is powerful enough to represent the differential effect of showing cards on the various players, but for lack of space we do not go into the details here.

4 Deriving Knowledge in MYST

We state a result that follows from the definition of Clue within the axiomatized framework. This result describes what a player must know to prove the existence of a card in the mystery pile, and thus guides the derivation of winning strategies.

Theorem 3. *Player i knows that a card c_x is in the mystery pile just in case i knows that none of the other players hold c_x. In symbols,*

$$Know(i, H(0, c_x), s) \equiv \forall j.Know(i, \neg H(j, c_x), s).$$

Furthermore, player i knows which cards are in the mystery pile just in case he knows which cards are not in the mystery pile. In symbols,

$$\forall c_x \in C_0.Know(i, H(0, c_x), s) \equiv \forall c_y \notin C_0.Know(i, \neg H(0, c_y), s).$$

The result follows more or less immediately from the partition axioms. The result establishes two subgoals for the main goal of determining that a card c_x is in the mystery pile: The first, sanctioned by the first part of the theorem, is to determine portions of the pile directly from "no" responses. The second, following the second part of the theorem, is to determine the locations of the $k - m$ cards outside the mystery pile from "yes" responses and then, by the set size axiom (SSA), deduce the m cards in the mystery pile. In either case, the set size axiom is crucial for drawing conclusions about the location of cards.

These observations are fairly obvious to a human analyzing the game. The point is that through our formalization of the game structure, a computational agent with theorem-proving capabilities can recognize these points and make use of them in planning queries.

In a multi-agent setting, optimal plans have an interactive and recursive structure, because an optimal plan for agent i must typically assume that agent j is following an optimal plan, which assumes that agent i is following an optimal plan ... Game-theoretic concepts that incorporate this recursive structure are the notion of Nash equilibrium and backward induction analysis (alpha-beta pruning) [?]. For restricted versions of MYST (for example, with two players only), we have determined the optimal backward induction strategies. Determining the Nash equilibria of MYST is an open question for future research.

We have also analysed aspects of the complexity of reasoning in MYST. Our analysis so far indicates that the computational complexity of this reasoning becomes intractable as the size of the game increases, but is quite manageable in relatively small spaces such as that of the original Clue game.

5 Conclusion

Clue and its variant MYST offer a number of challenges to a planning formalism for multi-agent interactions. We must represent the rules governing the interaction, uncertainty about the initial distribution of cards, the effects on knowledge and common knowledge of *show* actions, and assumptions about the reasoning of the agents, such as perfect recall. We showed that the epistemic version of the situation calculus, extended with a common knowledge operator, can represent all these aspects of the agents' interaction in a unified, natural and perspicuous manner. The formal representation permits agents to reason about each other's knowledge and their own, and to derive strategies for increasing their knowledge to win the game. Our results confirm the expectation that the situation calculus will be as useful for planning in multi-agent interactions in a game-theoretic setting as it has been for single-agent planning.