A Framework for Coherence-Based Multi-Agent Adaptivity

Nick Lacey;Henry Hexmoor
Computer Science and Computer Engineering
University of Arkansas
{nlacey,hexmoor}@uark.edu

Abstract

In this paper we present research which extends previous work concerning the application of philosophical theories to agent knowledge base (AKB) design. We show how the theories and techniques presented in this paper allow multiple agents to adapt to a dynamic environment, and pursue long-term goals and intentions. A system is implemented and described which tests the ability of agents based on the framework to act responsively, pro-actively, and cooperatively in a multi-agent environment.

1 Introduction

A *situated* agent is one which exists within an environment. Such agents receive information about their environment through their sensory equipment, and are able to perform actions in their environment through their effectors. In the case of both human and artificial agents, the world which the agent must interact with will often contain more complexity than can be represented within the internal model of the agent. This means that both human and artificial agents will face problems concerning the accuracy of their model in relation to their environment.

It is possible to form parallels between the problems faced by a situated agent and the questions posed by various areas of philosophy, especially Epistemology, Metaphysics, and the Philosophy of Language. The particular path taken through the philosophical maze in each of these areas has implications for the design and implementation of an artificial agent. As Van Inwagen points out, *everyone* has philosophical beliefs, whether or not they are explicitly aware of them (van Inwagen, 1993). As the designers of most artificial agents are not philosophers, they are implicitly basing their agents on the philosophical assumptions which come bundled with common sense, which Van Inwagen calls the *"Common Western Metaphysic"*.

In (Lacey, 2000), and (Lacey and Lee, 2001), it was shown that it was possible to construct two radically different philosophical positions and then construct artificial agents based on these positions. The performance of the agents was then compared, and it was concluded that the philosophical foundations on which an artificial agent was based do indeed affect its design, implementation, and behaviour.

In this paper we concentrate less on the philosophical motivations for this research, and more on extending the earlier work so as to produce agents which are capable of addressing more complex tasks.

This paper is structured as follows. In Section 2 we briefly describe the design, implementation, and behaviour of the original agent. In Section 3, we describe how the addition of intentions and the ability to represent social concepts to the agent's ontology renders the revised agent, agent **CX**, *pro-active* and able to operate in a multi-agent environment. Section 4 then describes experiments that have been carried out in order to test the new abilities of agent **CX**, the results of which are discussed in Section 5. Finally, Section 7 offers some concluding remarks concerning this work.

2 The Design and Implementation of Agent SH

Agent **SH** was designed to be **s**trongly **h**olistic. This means that the meaning of every belief within its ontology is, as much as possible, determined in relation to *every* other belief. This was implemented using a belief revision mechanism based on explanations, coherence and constraint satisfaction.

Agent **SH**'s low-level perceptions were organised and given meaning using high-level explanations of the agent's environment. The most coherent explanation at any given time was taken to be the correct explanation of the agent's current sensor data.

2.1 Explanations

The primary method of inference used by agent **SH** that of *Explanation-Based Backward Chaining*. The ultimate goal of the backward chaining process is to find the overall explanation which bests explains the current input data. The value of the explanation may be derived from several alternative explanations, each one representing an al-

ternative high-level conception of the current state of the agent's environment.

The first explanation that the agent will attempt to backward chain is the *Default Explanation*. This represents the most coherent explanation of all information received by the agent prior to the present moment. If the agent is able to find the value of the *Default Explanation* without violating any constraints, then it has no need to investigate any of the alternative explanations, and can adopt the default explanation.

Alternatively, the agent may find that no value for default explanation can be found without violating constraints. This indicates the agent's internal model is out of step with its environment, so some adjustment is required. If this occurs, as many alternative explanations as necessary are explored. The alternative explanations are derived and then holistically evaluated by comparing their *integrity* scores. This approach mirrors Thagard and Millgram's concept of the holistic assessment of competing hypotheses, and is based on philosophical approaches to truth and justification based on coherence.[1] (Thagard and Millgram, 1997)

Explanations are *Meta-Level* beliefs, meaning that they concern the relationships between other beliefs, rather than directly representing states of affairs in the agent's environment. Every explanation may incorporate other, lower-level explanations, as well as domain-dependent beliefs concerning facts and relations about the agent's environment. Alternative explanations will usually provide differing mechanisms to derive the same pieces of data. This means that if a piece of sensory data that is necessary for the default explanation is unavailable, the agent may be able to use its knowledge of its environment to derive the missing data from another source.

The precise nature of the explanations used by a given agent will depend upon the environment within which the agent is to operate. Coherence can be used to provide a measure of the "best" explanation. The concepts of *integrity* and *centrality* are used to measure the utility of a possible explanation. Integrity and centrality are used to measure the value of an entire knowledge base, and an individual belief, respectively.

If the ontology of the agent were to be visualised as a sphere, the central beliefs would be the meta-beliefs while the outermost beliefs would be state dependent assertions.[2] By associating each belief with the ontological level to which it belongs, the agent has access to a computationally inexpensive method of deriving the centrality of each belief.

This technique also addresses a common criticism of coherence based justificational structures, namely that they are unable to account for the differing epistemological importance attached to difference classes of belief. The spherical model provides a clear basis for holding that high-level core beliefs are more important to the agent's ontology that perceptual beliefs on the periphery, and hence are less likely to be revised.

[1] For more information on coherence as an approach to truth and justification, see (Audi, 1993) and (Kirkham, 1995).

[2] This structure was inspired by Quine (Quine, 1980).

2.2 Constraints

Constraints are used to alert the agent to the presence of an inconsistency. Their role can be seen as three-fold:

1. Constraints can be viewed as a shortcut, as they allow the agent to detect an inconsistency as early as possible in the knowledge base derivation process.

2. Constraints allow the high-level representation of states of affairs that cannot obtain in the agent's environment. If a particular constraint is violated under the current explanation, the agent can immediately infer that the explanation is flawed, and thus can invest its energies elsewhere.

3. Due to the high-level nature of constraints, it is possible to associate a particular recovery plan with every constraint. By following the recovery plan, the agent may be able to produce a new explanation which successfully resolves the current problem.

Constraints relevant to the agent's domain are provided by the agent designer. As such, constraints represent pre-processed knowledge provided to the agent. Pollock (Pollock, 1998) notes that this technique obviates the agent from having to concern itself with all the intricacies of the domain, and risk encountering the frame problem. While we accept this point, it should be noted that the agent's design is flexible enough that it may be possible to allow it to modify its constraints or add new ones as a result of experience, and indeed this is an interesting area for future developments.

3 Adding Intentions and Social Notions to the Architecture

Agent **SH** was responsive, rather than pro-active. The agent was able to perceive and react to changes in its environment, but was not able to represent or act on long-term plans or goals. Jennings, Sycara and Wooldridge (Jennings et al., 1998) define a responsive system as one which is able to perceive its environment and respond to changes which occur in it.

As such, responsiveness can be contrasted with pro-activity. A pro-active agent does not merely respond to its environment, but is able to exhibit goal-directed behaviour. The approach we used to create a pro-active agent based on agent **SH** was to add *intentions* to the agent's ontology. The BDI approach to agent design recognises the primary importance of beliefs, desires, and intentions (Wooldridge, 2000).

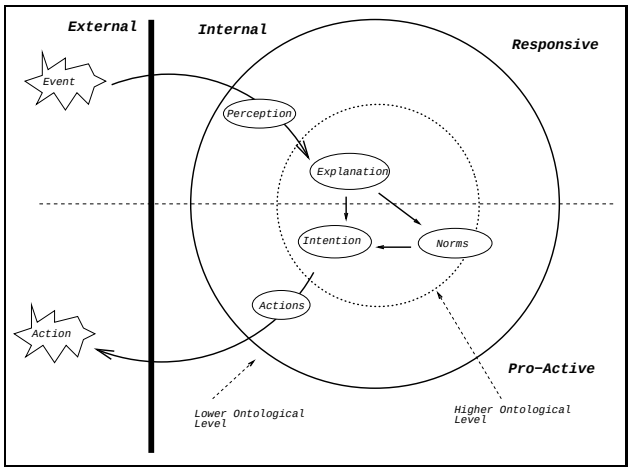

Figure 1: Adding Intentions and Norms to Agent **SH**

Intentions occupy a middle ground between desires and actions which allow the agent to focus on a manageable number of achievable short or long-term goals which will maximise, to the best of the agent's knowledge and abilities, its long-term utility. In other words, intentions allow an agent to be goal-driven, rather than event-driven (Schut and Wooldridge, 2001).

Thus, adding intentions to the architecture in which agent **SH** is based renders the agent pro-active, as well as responsive. The agent created as a result of the extensions to agent **SH** has been called agent **CX**, as it is based on an extended form of coherence.

As shown in Figure 1, the addition of intentions to the architecture of agent **SH** creates a framework based on a two-stage cognitive process. This figure also shows that norms have been added to the agent's ontology. The social aspects of agent **CX** are discussed in Section 3.5. The first stage is responsive, in that the agent uses explanation-based backward chaining to form the most coherent explanation of its current sensor data.

The second stage, however, is pro-active. During this stage, agent **CX** uses the high-level explanations generated during the first stage as a guide when forming an *intention tree*. Intentions relevant to the current domain are combined using backward chaining, in much the same way as explanations were combined during the responsive stage of the agent's cognitive cycle.

3.1 Intentions, Plans, and Actions

An interesting feature of this architecture is that the agent does not represent *plans* explicitly. This is not to suggest that the agent does no planning. Clearly, some form of planning is essential if the agent is to achieve long-term goals. However, in the architecture we suggest intentions and actions take on the role of plans, such that there is no longer a requirement for any explicit plans above and beyond intentions and actions.

Figure 1 shows the proposed relationship between explanations, intentions, and actions. Perceptions arrive from the external world, and are processed to arrive at an explanation. As shown in the diagram, this process is responsive, rather than pro-active. This is because, during this stage, the agent is making sense of what has occurred in its environment, rather than making any plans concerning what action to take in the future. However, while the explanation derivation stage is responsive, it is *not* passive. This is because the perceptions are not accepted at face value and used as the basis for the explanation. Rather, the perceptions are taken as one possible source of data concerning the agent's environment. The version of events concerning the environment which will eventually be accepted depends on results of the explanation derivation process which, far from being passive, requires varying amounts of processing, depending on the ease with which new data can be integrated into the agent's existing ontology.

Extending the design of the agent to allow the representation and manipulation of intentions renders the agent *pro-active*, as well as responsive. This is because the agent can use its intentions to represent long-term goals which it must use some degree of planning to bring about.

As shown in Figure 1, we have placed intentions at a higher level within the ontology than beliefs concerning particular actions. This reflects the fact that intentions are more centrally held than beliefs about actions. Small day-to-day occurrences cause us to constantly revise our planned actions without affecting our longer-term intentions.

For example, my intention this morning, like most mornings, was to get to work. This intention is normally executed by constructing a plan which involves me walking to work. However, this morning I looked out of the window and noticed that it was raining heavily. I therefore revised my plan, in that I decided to drive to work instead of walking, but my intention was unchanged. The reason my intention was unchanged was that this was a regular event that I knew from experience how to deal with. However, if I had experienced a highly unusual event this morning, such as an absence of gravity, it is possible that this would have caused me to revise not only my current plan but also my intention.

Our suggestion, then, is that the agent should be able to distinguish situations which do warrant the revision of intentions from those that do not. Furthermore, the agent should, on the whole, exhibit a reluctance to alter its intentions. This is because the act of intention reconsidering consumes resources which could be used elsewhere, so the agent should be discouraged from intention reconsideration except when this becomes necessary.

3.2 The Advantages of High-Level Intentions

At the sensory perception level there is a great deal of low-level information available to the agent. The dependencies, implications, and meaning of this information

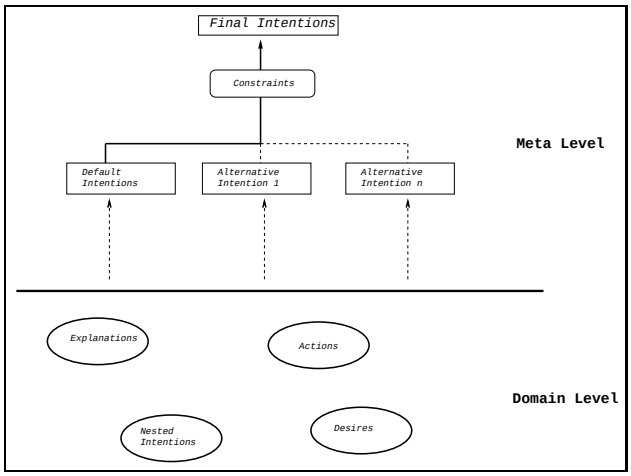

Figure 2: Intentions Provide Guidance Through the Action Search Space

are not contained in the low-level perceptual data. This makes it difficult for the agent to build a complex model solely on the basis of this information. However, high-level explanations *do* contain the connections between various pieces of sensor data which the agent must refer to when constructing its model.

Thus, high-level explanations provide methods of combining this low-level data in different ways, depending on which explanation is chosen. If the agent is able to, it will use the default explanation. If this is not possible, alternate explanations are generated and holistically compared.

The central idea that will be put forward in this section is that, within the coherence framework described in Section 2, *intentions can be viewed as the pro-active equivalent of explanations*. Competing alternative explanations are used to guide the backward chaining process which eventually yields a maximally coherent interpretation of the agent's environment. Similarly, competing alternative intentions are used to guide the planning process, and will yield a set of intentions which are maximally coherent with the agent's existing beliefs.

Just as dealing with low-level sensor data alone makes perception difficult, so dealing with low-level actions alone makes planning difficult. By considering its actions at the intention level, the agent is better equipped to deal with dependencies and inconsistencies that may exist between its planned actions.

This is not to suggest that the agent will have to generate a competing set of intentions on the basis of every interpretation of the environment. Indeed, the ability of this model to provide a well-defined yardstick by which to determine when intention reconsideration is necessary is part of its appeal.

Rather, the default interpretation will always be carried forward from the previous interpretation. If nothing has changed, the agent can go ahead and use its default set of intentions. Even if the interpretation has changed, an agent may still be able to use the default intentions. Whether or not this is possible will depend on whether or not any of the constraints which link the interpretations and intentions have been violated. If they have not, then the default set of intentions is still valid. If constraints have been violated, then the intention reconsideration process must begin.

3.3 Representing Intentions

An intention encapsulates the actions required to bring it about. It is assumed that the actions encapsulated within an intention are internally coherent. This assumption allows us to verify the coherence of the pro-active side of the agent only at the level of the agent's intentions.

Intentions are combined to form a rooted intention tree, similar to the explanation tree described in Section 2. The paths from the root intention to any of the leaves represent alternative paths through the intention space.

Intentions incorporate the following concepts:

Action Templates A data structure containing templates which describe how actions can be formed which will bring about the intention.

Preconditions States that must obtain in the world before the actions can be implemented.

Postconditions States that will obtain in the world once the actions have been implemented.

Constraints A list of constraints which are associated with the intention.

While the concepts of constraints and preconditions may seem similar, they serve different functions within the intention reconsideration process. The purpose of the preconditions and post-conditions is to guide the backward chaining process. They do this by allowing the backward chaining engine to match the preconditions of a required intention with the post-conditions of an intention which must precede it. In this manner executable intention paths can be formed.

While preconditions are used during the intention forming process, constraints are used to *verify* the consistency of an intention set with respect to the given explanation. Preconditions are used to allow the agent to combine intentions in an effective manner, and as such will usually concern lower level aspects of the agent's environment than will be represented in constraints. However, this does not mean that constraints and preconditions will necessarily be independent. Indeed, as they both concern representations of a state of affairs that must obtain before another state of affairs can obtain, there may be some similarities between preconditions and constraints which the agent will be able to exploit.

3.4 Intentions Give Meaning to Actions

Intentions combine to form coherent methods of achieving the desired goal. They are then translated into actions which are put into effect to achieve the desired intention. Note that while individual intentions are meaningful, individual actions have little meaning when taken in isolation from their related intentions. This reflects the distinction between human intentions and actions.

For example, consider my intention to type the word "for". At the intention level, I formulate the intention to type the word. At the action level, however, the actions required to carry out this intention are *all* of the form"Move hand horizontally to a certain position, then move fingers vertically."

The point is that very little meaning can be attached to actions in themselves. In order to determine the meanings of actions, we must refer to the intentions which the actions represent an attempt to bring about. This means that when reasoning about actions, we should be reasoning at the *intention* level, rather than the action level.

Grosz, Hunsberger, and Kraus argue that agents operating in a multi-agent environment can hold intentions without knowing *themselves* how to bring the intention about (Grosz et al., 1999). While this definition of intention may not be completely compatible with ours, we do agree that actions and intentions are distinct, mutually supporting entities. Thus, we are not arguing that intentions should be seen as *replacing* actions, as actions will always be necessary in order to bring about the agent's intentions. Rather, we suggest that agents should construct and manipulate plans at the intention level, rather than the action level.

Intention constraints are used by the system to determine whether variations between intended and actual execution have rendered the current set of intentions obsolete. If they have not, then intention reconsideration is not necessary. If they have, then the agent must reconsider its intentions.

Whether or not the intention constraints succeed, action constraints are applied prior to executing each action. Action constraints are designed to detect cases where the overall intention is still valid, but the original action which was originally associated with that constraint must be varied. This variation is entirely local to the current intention, and does not affect the rest of the system. In effect, this mechanism adds a degree of responsiveness to the system.

3.5 Representing Norms of Behaviour

In (Hexmoor and Beavers, 2002), Hexmoor and Beavers discuss extending the traditional BDI approach to agent design by also considering the concepts of values, obligations, and norms. They conclude that adding these new modalities increases the robustness and flexibility of the agents that are produced.

We take values to be abstract beliefs that guide the behaviour of an agent, while norms are specific instances of behaviour. In this paper we incorporate norms directly into the agent architecture, but we consider obligations and values to be implicitly represented.

The relationships between norms and intentions are represented using relations and constraints. Relations are used to represent the fact that an agent should behave in a certain way in a particular situation, and in a different way in a different situation.

It is also worth noting that norms are directly related to intentions, as opposed to actions. This is because, as described earlier, very little meaning can be attached to low-level actions, so relationships concerning the social make-up of agents must be defined at the intention level.

For example, consider an agent who takes goods from a store without paying for them. Whether or not this agent is violating any norms or values depends entirely on what its intention was. If the agent was intending to steal the goods, then it has clearly violated a norm of behaviour that would be associated with the value that stealing is wrong. If, on the other hand, the agent was intending to take the goods home on approval, and was acting with the consent of the store owner, then no norms or values have been violated.

When combined with the explanation-based backward chaining mechanism described above, this framework allows the agent to construct intention trees which are based on the adoption of a set of default norms. The constraint violation detection mechanism will ensure that any events in the environment which are inconsistent with the default model will be flagged as such, and will cause the model to be updated accordingly.

4 Implementation and Experiments

In order to test the architecture described in this paper, a system capable of constructing, executing, and manipulating intention trees was implemented in Prolog.

4.1 Experimental Domain

The experimental domain that was used was chosen to be as simple as possible while nonetheless requiring that intention trees be constructed and modified as appropriate. The domain that was chosen to be the basis of these experiments was that of a crude discgolf simulation. Discgolf is similar to golf, but players throw plastic discs instead of hitting golf balls. The environment consists of a rectangular area. At the beginning of the simulation, the disc is placed on a tee. The agent must formulate an intention tree which will permit it to throw the disc into the basket using as few throws as possible.

An advantage of this domain is that while the distinction between intentions and actions is clear at the conceptual level, at the implementational level translation from the intention level to the action level is very simple. Intentions concern an attempt to throw a disc to particular

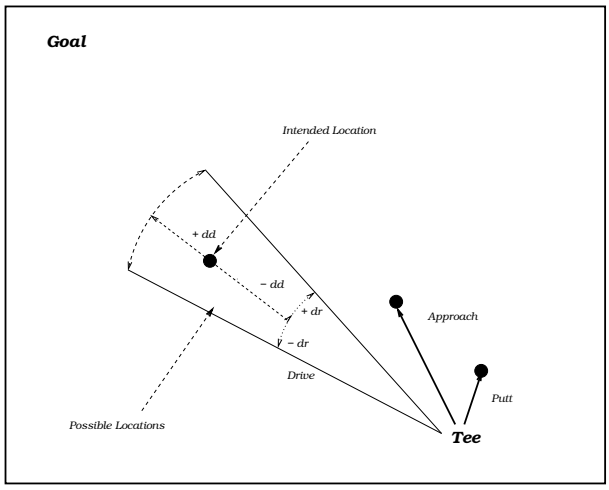

Figure 3: The Different Intentions Available to the Agent

location, specified by (x, y) coordinates. The action required to bring about the intention concerns throwing the disc in a particular direction and aiming to cover a particular distance. Deriving the distance and direction values, given the target (x, y) values and the disc's current position, is a matter of simple geometry.

The agent has three types of intention available to it, as summarised in Table 1. The intentions represent different types of throw. Throws which will move the disc further are less accurate, while the more accurate throws do not cover as much distance. This effect was achieved by adding a certain amount of error to the agent's intended throw. This is illustrated in Figure 3. The representation of the possible actual locations of the disc after a throw are based on Shanahan's concept of the circle of uncertainty (Shanahan, 1996).

The amount of error added to each throw is controlled by two values.

- dr represents the number of degrees by which the actual direction of the throw may vary from the intended direction.
- dd represents the number of units by which the distance of an actual throw may vary from its intended distance. This figure represents a percentage of the actual intended distance. For example, if the intended distance of a throw is 300 units, and $dd = 5\%$, then the actual distance of the throw will be accurate to within ± 60 units.

The dr and dd values used for the experiments described here are given in Table 1. As these experiments represent a proof of concept rather than an attempt at a realistic simulation, dr and dd were set to the same value in each experiment.

As described above, agent **CX** forms intention-level plans by constructing intention trees. Once the agent has constructed the intention tree, the agent begins to formulate actions which will bring about the intention. It does this using the action template associated with each intention. In this domain, all intentions were associated with the `throw` action. This implements the requirement that intentions should constitute meaningful segments of the agent's plan in their own right, while individual actions need not be meaningful when considered in isolation from their associated intentions.

Thus, intentions take the form:

```
[drive,[x,y]]
```

where x and y are the coordinates of the disc's intended location *after* the intention has been executed. For leaf intentions, this value will be the coordinates of the target, while for intermediary intentions, the intended disc position will be a function of the maximum distance of the intended throw and the disc's position prior to the intention.

Intention	**Max Distance**	dr **and** dd **for Experiment**		
		1	**2**	**3**
Putt	20	0	0.5	1
Approach	200	0	1	2
Drive	300	0	1	5

Table 1: The Intentions Available to the Agent in the Discgolf Simulation

Actions, on the other hand, have the following format:

```
[throw,D,B]
```

where D is the distance of throw required and B is the bearing at which the throw should be aimed in order to reach the position specified by the intention. Thus, the distinction in semantic level between intentions and actions is clear: intentions concern x, y coordinates in the environment, while actions concern the strength and direction of a throw.

The constraints associated with the intentions summarised in Table 1 are used to ensure that inaccuracies in throws do not necessarily lead to a change in intention. However, if the error placed on a throw is large enough, intention reconsideration may become necessary. This is achieved by calculating the distance from the disc's current position to the sub-goal specified by the intention. If this distance is greater than the maximum range of the current intention, then the current intention, and all the intentions which follow it, must be reconsidered.

The constraints associated with actions are used to allow the agent to cope with minor variations between the intended and actual course of events which do not require intention reconsideration. If the actual position of the disc prior to the current intention is different from the intended position, but still within the range of the intended throw, then a new action is created. This action will re-use the throw type specified by the intention, but will calculate distance and direction values based on the disc's actual position.

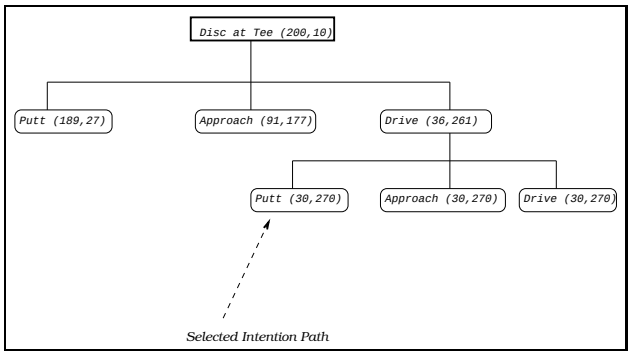

Figure 4: The Intention Tree Constructed In Experiments 1 and 2

5 Results

The description of the experiments is divided into two sections. Section 5.1 describes three single-agent experiments, while Section 5.2 describes a multi-agent experiment. The results presented in this section have been rounded to 0 decimal places in the case of (x, y) coordinates, and 1 decimal place in the case of directions.

5.1 Single Agent Experiments

Experiment 1 did not involve any errors. As such the throws associated with each intention landed with 100% accuracy, meaning there was no need for intention or action reconsideration. The intention tree constructed during Experiment 1 is shown in Figure 4.

With dr and dd set to 0, throws at the action level represented these intentions with 100% accuracy. The intentions and actions used in the experiments are shown in Table 2. Note that both throws used in Experiment 1 share the same direction, as the disc is moving along a perfectly straight line from the tee to the goal.

Exp.	Intention			Action - Throw	
	Type	x	y	**Distance**	**Direction**
1	Drive	36	261	300	56.8
	Putt	20	270	11	56.8
2	Drive	36	261	300	56.8
	Putt	20	270	10	43.7
3	Drive	36	261	300	56.8
	Approach	20	270	23	28.8

Table 2: Intentions and Corresponding Actions from Experiments 1,2, and 3

The purpose behind Experiment 2 was to investigate the ability of the architecture to formulate new actions without intention reconsideration. In order to do this a small amount of error was added to each throw, as summarised in Table 1.

Constraints associated with each action are used to ensure that the action originally represented by the intention is still valid. In this case, this was done by checking whether the disc was actually at the location it should be at when the action is undertaken.

The experiment was successful, in that the agent was able to move the disc from the tee to the goal without reconsidering its intentions, despite the fact that the disc never landed exactly where it was intended to. As would be expected, as the error added to the throws was produced randomly, results varied between different runs. The execution of the final throw, which will usually be a putt, was unsuccessful in some cases. In these cases, a new intention had to be created in order to accomplish the goal. In cases where the second throw was successful, the intention tree resulting from Experiment 2 was exactly the same as that resulting from Experiment 1, shown in Figure 4.

Results from a representative run of Experiment 2 are given in Table 2. The first intention and action are carried out as normal. After the first throw, the agent realises that the disc is not at the intended location, namely $(36, 261)$. However, the distance between the disc's actual location and the intended location of intention 2 is such that a putt is still an applicable intention. However, as the disc is not where it should be, the original intention must be brought about using a different action.

In the example shown, the putt required by intention 2 was successful. In cases where this putt was not successful, an additional putt intention was generated, as follows:

```
[putt,[20,270]]
```

The intention is unchanged, as the goal is still that of placing the disc in the target. This intention will be translated into an action, such as:

```
[throw,2,3.4]
```

This process is repeated until the throw is successful. The actual parameters of the throw will clearly vary depending on where the disc lands after each attempt. The agent usually required between 2 and 4 throws to reach the target.

This corresponds to the approach humans take when attempting to bring about a difficult intention. For example, when trying to unlock a door at night, my intention is to place the key in the lock. My first action is to move the key toward the lock, and to attempt to insert it. If my first attempt fails, my intention *remains* that of placing the key in the lock, while at the action level I am moving the key in various directions until I feel that the key has entered the lock.

The purpose of Experiment 3 was to add sufficient error to the throws to cause the agent to reconsider its intentions. Whether or not intention reconsideration was necessary was represented using the following constraint: If the distance between the disc's actual position and its

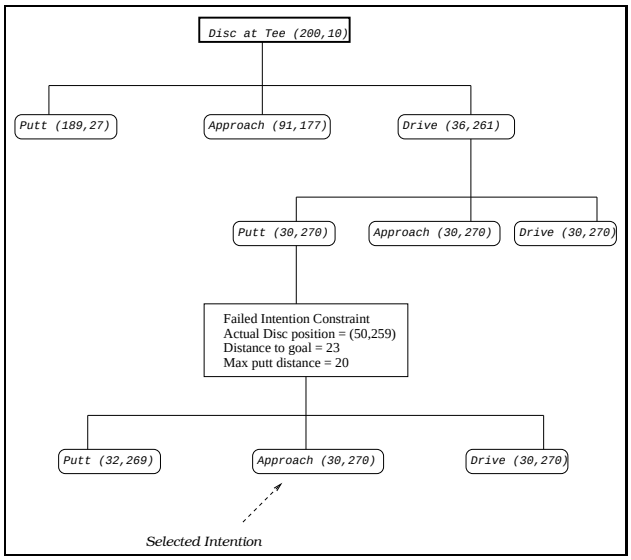

Figure 5: The Intention Tree Constructed In Experiment 3

intended position was greater than the maximum distance of the intended throw, then intention reconsideration was necessary.

An intention tree representing a sample run of Experiment 3 is shown in Figure 5. For the sake of simplicity, a run has been selected here in which the agent successfully completed the task in 2 throws. Most of the runs in Experiment 3 required between 3 and 5 throws.

After the first throw, the disc ends up at $(50, 259)$. This is too far for the intended putt, so intentions must be reconsidered. Using the disc's actual position as the starting point, the agent produces a new set of intentions. A putt is indeed too short to reach the target, but an approach throw may be successful. The approach intention is selected, and translated into a distance and direction value, as shown in Table 2.

Despite the relative simplicity of the domain, these experiments show that it is possible for an agent to construct an intention tree in order to bring about a long-term goal. A small amount of variation when executing an intention will not necessarily require intention reconsideration, but the agent will be willing and able to reconsider its intentions on the fly if this becomes unavoidable.

5.2 Extending the Experiments to the Multi-Agent Scenario

Agent **CX** is able to model and represent the intentions of other agents. The explanation-based component of the agent uses observations of the actions of other agents to form the most coherent explanation of what the other agents in the environment are attempting to do. Once the intentions of the other agents have been decided, agent **CX** can adapt its intentions accordingly. The precise nature of this adaptation will depend on the application domain, and on the relationship between the agents.

In cooperative discgolf, two players play on the same team. Both players make a throw, and then decide which was the best throw. Both players then play from the overall best throw on the previous shot, and so on. If the first player plays a good safe shot, then the second player is free to attempt the risky shot, as the team has nothing to loose. On the other hand, if the first player's shot is unsuccessful, the second player must also attempt a safe shot.

In our simulation, agents playing cooperatively may play in accordance with two norms of behaviour, namely `safe` and `risky`. In the multi-agent scenario, each of the three intentions described in Section 5.1 is associated with either a safe or a risky norm of behaviour. Safe intentions have lower dr and dd values and a shorter range, while risky intentions have higher dr and dd values and a higher maximum range. The dr, dd, and maximum range values associated with the two norms in Experiment 4 are shown in Table 3.

Intention	**Max Distance**		dr **and** dd	
	Safe	Risky	Safe	Risky
Putt	20	40	1	5
Approach	200	300	2	10
Drive	300	400	5	20

Table 3: The Intentions Available to the Agents in Experiment 4

As in the single agent case, both agents construct intention trees. As shown in Table 3, the likely range and error values for each intention depend on whether the agent has adopted a `safe` or `risky` norm of behaviour for that shot. Both agents construct their intention trees assuming that they will be using the `safe` norm. If conditions arise which allow an agent to play under the `risky` norm, this condition will be detected by a constraint violation, and the agent will revise its intention tree appropriately.

A feature of this framework is that as long as all agents in an environment share the same ontology, they will all construct *exactly* the same intention trees.

This is due to the fact that this framework is not based on probabilities. The operation of the intention tree construction and revision mechanism may be complex, but it is completely deterministic and repeatable. This means that agents sharing the same starting ontology, and receiving the same perceptions from the environment, will construct the same intention trees. This allows multi-agent cooperation in complex domains without the need for inter-agent communication.

The cycle used by the two agents is as follows:

1. Determine which of the two players, **A** or **B**, made the best throw on their last turn. This is determined by measuring the distance form both discs to the target. Store the best disc location in (bx, by)

2. Both **A** and **B** now play from (bx, by). The agent

that made the best throw on the previous turn plays first.

3. The agent to play first makes their throw. The first agent to throw will always be using the `safe` intentions.

4. The second agent to throw observes where the first agent's disc landed. Based on this information, it has two choices:

   - If the first throw was good, the team of agents now has nothing to lose. This means that the second agent to throw is free to adopt a risky style of play, and so alters its intention tree so as to select the appropriate `risky` intention.
   - If the first throw was not good, the second agent must play a safe shot, rather than a risky one.

Turn	Agent	Intention			
		Type	Norm	x	y
1	**A**	Drive	Safe	36	261
	B	Drive	Risky	30	270
2	**A**	Approach	Safe	30	270
	B	Approach	Risky	30	270
3	**A**	Putt	Safe	30	270
	B	-	-	-	-

Table 4: Intentions and Norms From Experiment 4

Results from a representative run of Experiment 4 are given in Table 4. The explanation for these results is as follows: At the start of play, agent **A** threw first, using a `safe` norm of behaviour. The disc ended up at position $(63, 254)$. This qualifies as a good throw, meaning that agent **B** is now free to adopt a `risky` norm for the same throw. This is reflected by the fact that agent **B** is intending to throw the disc all the way into the basket on the first throw.

Agent **B**'s first throw is less successful, ending up at $(157, 50)$. The distance from agent **B**'s throw to the target is 254, compared with a distance of 36 for agent **A**'s throw, so both agents move to $(63, 254)$.

Agent **A** throws first on turn 2, as it threw the best shot on the previous turn. The distance to the target is 36, which is within the range of a safe approach or a risky putt. However, as this is the first shot for this turn, agent **A** has no choice but to play the safe approach. Agent **A**'s shot ends up at $(27, 273)$, which is only 4 away from the target. This also qualifies as a good shot, so agent **B** is once again free to play a risky shot. Agent **B**'s shot ends up at $(43, 317)$, which is considerably worse than agent **A**'s shot, so both agents move to $(27, 273)$. Agent **A** now faces an easy putt, and hits the target on the first attempt.

These results show that agent **B** was able to change its style of play, depending on the results obtained by its team-mate. In this instance, the performance of the team was not helped by agent **B**'s risky shots, but it is nonetheless clear to see why agent **B**'s choice of actions was correct, as the potential gains for the team outweighed the potential losses for the individual.

5.3 Experimental Results - Summary

Our interest in this domain stems from the fact that while it is relatively simple to implement, it nonetheless yields the following interesting features:

- Agents can operate either individually or as part of a team. The best course of action for an individual agent is not necessarily the best course of action when the agent is operating as part of a team. Agent **CX** is able to modify its behaviour when operating as part of a team, as an agent playing individually would never be wise to adopt a risky style of play.
- The agents are homogeneous. Each agent can intend to play in one of two different way. Which norm the agent chooses to adopt depends on the actions of its team-mate. As shown above, agent **CX** is able to dynamically adapt to new norms of behaviour at run-time.
- The agents can plan pro-actively at the intention level, can react to small variations in disc position at the action level, and are able to adapt to different styles of play depending on the actions of their team-mate.

6 Acknowledgements

This work is supported by AFOSR grant F49620-00-1-0302.

7 Conclusions

In this paper we have shown how an agent architecture which was based on philosophical approaches to truth and justification can be extended to produce a framework for intelligent agent design. We have shown that agents based on this framework are capable of responding to events in the environment and of exhibiting goal-directed pro-active behaviour.

In addition, this framework facilitates the design of agents which are capable of forming robust intention-level plans which can cope with small variations in the environment without necessitating plan regeneration. When the environment changes to such an extent that a new plan is necessary, the agent is capable of detecting this situation and constructing a plan which addresses the problem

it was facing. A feature of this architecture is that multiple agents can cooperate on the same problem without communicating, as they will each form identical intention trees as long as they share the same starting ontology.

Thus, we conclude that the proposed framework can be used to design and implement situated adaptive agents which can exhibit responsive, pro-active, and co-operative behaviour in a multi-agent environment.

References

Robert Audi. *The Structure of Justification*. Cambridge University Press, Cambridge, UK, 1993.

Barbara Grosz, Luke Hunsberger, and Sarit Kraus. Planning and acting together. *AI Magazine*, 20(4):23–34, 1999.

Henry Hexmoor and Gordon Beavers. In search of simple and responsible agents. In *Proceedings of the GSFC Workshop on Radical Agents*, MD, 2002.

Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and development. *International Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

Richard L. Kirkham. *Theories of Truth: A Critical Introduction*. The MIT Press, Cambridge, Mass, 1995.

Nicholas Lacey. *Investigating the Relevance and Application of Epistemological and Metaphysical Theories to Agent Knowledge Bases*. PhD thesis, University of Wales, Aberystwyth, 2000.

Nicholas Lacey and Mark Lee. The implications of philosophical foundations for knowledge representation and learning in agents. In Daniel Kudenko and Eduardo Alonso, editors, *Proceedings of the AISB 01 Symposium on Adaptive Agents and Multi-Agent Systems*, pages 13–24, York, UK, 2001. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

John Pollock. Planning Agents. In Rao and Wooldridge, editors, *Foundations of Rational Agency*. Kluwer, 1998.

Willard Van Orman Quine. Two dogmas of empiricism. In *From a Logical Point of View : 9 logico-philosophical essays*, chapter 2, pages 20–46. Harvard University Press, Cambridge, Mass, 1980.

Martijn Schut and Michael Wooldridge. Principles of intention reconsideration. In *AGENTS'01*, Montreal, Canada, 2001. ACM.

Murray Shanahan. Noise and the common sense informatic situation for a mobile robot. In *AAAI/IAAI, Vol. 2*, pages 1098–1103, 1996.

Paul Thagard and Elijah Millgram. Inference to the best plan: A coherence theory of decision. In A. Rom and D.B. Leake, editors, *Goal-driven learning*, pages 439–454. MIT Press, Cambridge, MA, 1997.

Peter van Inwagen. *Metaphysics*. Dimensions of Philosophy Series. Oxford University Press, Oxford, 1993.

Michael Wooldridge. *Reasoning About Rational Agents*. MIT Press, Cambridge, MA, 2000.