Temporal Action Logic for Question Answering in an Adventure Game

Martin MAGNUSSON and Patrick DOHERTY
Department of Computer and Information Science
Linköping University, 581 83 Linköping, Sweden
E-mail: {marma,patdo}@ida.liu.se

Abstract. Inhabiting the complex and dynamic environments of modern computer games with autonomous agents capable of intelligent timely behaviour is a significant research challenge. We illustrate this using our own attempts to build a practical agent architecture on a logicist foundation. In the ANDI-Land adventure game concept players solve puzzles by eliciting information from computer characters through natural language question answering. While numerous challenges immediately presented themselves, they took on a form of concrete and accessible problems to solve, and we present some of our initial solutions. We conclude that games, due to their demand for human-like computer characters with robust and independent operation in large simulated worlds, might serve as excellent test beds for research towards artificial general intelligence.

Keywords. Temporal Action Logic, computer games, natural language understanding, artificial general intelligence, natural deduction, planning, epistemic reasoning

1. Introduction

Two topics that have seen a recent boost of interest are research on artificial general intelligence (AGI) and the use of modern computer games as AI research test beds. There is much to say in favour of combining these trends, though we confine ourselves to two important observations. First, games are readily accessible both for the scientist who can use existing games with exposed APIs, or relatively easily implement entirely new games, and for the peer researcher or student who can download *and experiment with* the software themselves. Second, their demand for human-like behaviour in complex environments necessitates a certain amount of generality in any proposed solution. Game environments are much more complex than classical benchmark problems such as the blocks world, which are often criticised for their limited scope (e.g. by Hayes [1]). In contrast, most computer games are incompatible with simplifying assumptions such as the (in)famous closed world assumption and call for many of the capabilities needed for general intelligence such as an agent architecture that integrates everything from perception to action, robustness and responsiveness in sometimes unpredictable environments, goal-directed action planning, multi agent communication, reasoning about knowledge and how to obtain it, and natural language understanding for dialog interaction.

Our own work involves research on topics relevant to an adventure game project where a human player solves simple puzzles through natural language question answering dialogs with ANDIs, agents with Automated Natural Deduction based Intelligence, who inhabit ANDI-Land. Present day games almost universally adopt straight jacketed exchanges typically featuring a choice between three canned sentences, two of which are humorous sidetracks and one that will move the dialog forward to the next set of sentences. Our aim is to eliminate the forced linearity of scripted dialogs through artificial intelligence technology. Rather than mindlessly trying all alternatives, we would have the player think[1]. The reader is encouraged to evaluate the results we describe below through experimentation with our demonstrator available for download (as a Windows binary) at `www.andi-land.com`.

The long term aim is wide coverage natural language understanding, which requires both extensive knowledge of the topics under discussion and the capability to reason with it. Such demands can ultimately only be satisfied by true AGI, while our efforts to date are certainly not in that ballpark. But our initial experiences with ANDI-Land indicate that a computer game setting enables an incremental approach where reasonably difficult challenges can be attacked while keeping the long term goal in mind. The work presented below is, for this reason, based on a logicist foundation. We believe the best way to approach general intelligence is by formulating most types of reasoning in a unified proof system for deductive and non-monotonic types of inference in a, not necessarily purely classical, logical formalism expressive enough to capture all the subtleties and distinctions that humans make in their reasoning. If successful, such an endeavour will allow the use of efficient specialized reasoning processes when applicable, yet always providing the option to fall back on more general but less efficient methods of proof in new and unforeseen situations.

Rather than expanding further on this nebulous conjecture we will discuss the specific research problems that immediately suggested themselves when we initiated work on our question answering adventure game concept, in Section 2. Section 3 presents example dialogs from an in-game scenario that illustrate some capabilities of the architecture built in response to the challenges. A hopelessly inadequate selection of related work, squeezed into Section 4, will have to make do for orienting our efforts in relation to others'. Finally, Section 5 concludes with a look towards the future.

2. ANDI-Land

ANDI-Land consists of an isometric graphical representation of a forest that can be explored by a player through a keyboard controlled avatar. The forest is inhabited by intelligent agents with which the player can initiate question answering conversations, and who sometimes proactively do so themselves in order to further their own goals. There is a puzzle element to make interaction interesting, but unlike most other adventure type games, solving puzzles through a process of eliminating all alternatives is not feasible since the natural language input is not sufficiently restrictive. Implementing this concept requires providing ANDI-Land agents with a genuine understanding of questions posed to them and equipping them with knowledge of their virtual world from which to deduce answers. Only the coordination of linguistic and semantic processing can make this possible.

[1] Although whether this constitutes an enjoyable game experience depends, of course, on the player.

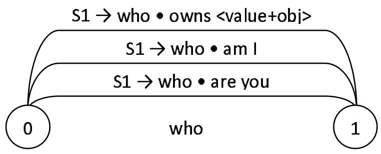

Figure 1. Active edges in the chart resulting from parsing “who” show the possibilities for the next word in the sentence, “owns”, “am”, and “are”.

2.1. Interactive Natural Language Input

As we pointed out in Section 1, it would be unrealistic to expect both broad and deep natural language understanding sooner than the development of AGI. First, no one has yet been able to construct a grammar with adequate coverage of the entire English language. The problem is not just that players of the game will not be able to express themselves in whatever way they want, but worse, nor will they receive any hints as to how to rephrase themselves in a way that the parser will understand. Second, the knowledge of the game characters will, while aiming at generality, realistically start out as downright narrow. Most sentences would simply be outside the area of competence of the game characters, and they would have to respond with an honest “I don’t know”.

These two problems threaten to reduce the adventure to a sort of guessing game where the player, more or less blindly, would have to search for sentences that both avoid the equivalent of a "parse error" message and whose semantic meaning happens to produce something other than a puzzled look on the respondent’s face. Using a very large wide coverage grammar like the English Resource Grammar [2] would seem to help alleviate the first problem, but at the cost of worsening the second. The semantical form it produces is not detailed enough to suffice for automated reasoning and question answering. Our project would be quite stranded if these problems left no room for incremental progress building on a modest start.

ANDI-Land incorporates a unique form of interactive natural language input that deal with both problems by guiding the player towards sentences that parse correctly and fall within the AIs competence areas. For example, the sentence “who is the lumber’s owner” is not supported by our current grammar, but the sentence “who owns the lumber” is. Even though this restriction is arbitrary, the player is spared much frustration due to the interactive parser. As soon as the player starts typing, a chart parser (as described in [3])[2] starts producing partial parses that cover as many words as possible. Though the initial word “who” does not constitute a complete sentence, the resulting parse chart still contains useful information. Specifically, by looking at active chart edges we can collect all words that would advance the parse if they occurred immediately succeeding the current input. According to Figure 1, the words “owns”, “am”, and “are” constitute all the possible continuations of the current input, and choosing among them effectively circumvents all sentences that have no chance of resulting in a complete parse. The process could be likened to the widespread T9 mobile phone text input, except that the system understands the grammar of entire sentences rather than just the correct spelling of words. Furthermore, by limiting the grammar to concepts covered by the agent’s background knowledge we can ensure that most of the input sentences are answered intel-

[2]The parser and grammar are extremely simple. We would like to improve them at a later date, perhaps modifying the Linguistic Knowledge Builder [2] for our interactive mode.

Natural language question	Who owns the lumber?
Input logical form	$\exists \textit{ans}\, [\textit{value}(\mathrm{now}, \mathrm{owner}(\mathrm{lumber})) = \textit{ans}]$
Answer variable binding	$\textit{ans} = \mathrm{djak}$
Answer logical form	$\textit{value}(\mathrm{now}, \mathrm{owner}(\mathrm{lumber})) = \mathrm{djak}$
Natural language response	Djak owns the lumber.

Figure 2. The process from natural language question to natural language response is largely symmetrical, thanks to a "reversible" grammar.

ligently. Compared to scripted dialogs, the interactive input method presents the player with a multitude of choices, even using a very small grammar, while also allowing for gradual improvements in language coverage.

2.2. Reversible Natural Language Grammar

Another challenge, natural language generation, is presented by communication in the opposite direction. Each grammar rule has an associated lambda expression that represents its meaning. Meaning fragments are combined by lambda application that eventually, after a complete parse of the sentence, results in formulas of first-order logic. These formulas are directly amenable to automated reasoning to produce an answer expression, encoding the response in first-order logic. Of course, we would prefer the game characters to reply in plain English. Shieber's uniform architecture [4] for both parsing and generation addresses this difficulty with minimal machinery. It effectively makes the natural language grammar reversible with relatively small modifications to the basic chart parsing algorithm. There is no need for separate generation algorithms. Furthermore, extensions to the input grammar that help the ANDI-Land inhabitants understand new words or grammar automatically increase their proficiency in using them in speech too. Figure 2 illustrates the question answering process using the example sentence from the previous section. A question is parsed into a formula that contains an *answer variable*. Its value, found through the theorem proving techniques described in Section 2.6, can be used to instantiate the query to form an answer expression. Finally, the chart parser is run in "reverse" to produce a natural language response to the original question.

2.3. Temporal Action Logic

We said that the logical forms resulting from parsing were amenable to automated reasoning. Work within the methodology of formal logic provides a comprehensive tool set for correct reasoning. However, the standard philosophical logic turns out to be inadequate to support the thinking processes of *active* characters in *dynamic* environments. Researchers in cognitive robotics are therefore creating new powerful logics that are applicable to commonsense reasoning about action and change as well as more traditional logical reasoning. We have chosen to work with one such logic, the Temporal Action Logic (TAL), which adopts an intuitive explicit time line to describe actions and their effects in a changing environment. The origins of TAL can be found in the Features and Fluents framework developed by Sandewall [5], but it was a new characterization in terms of first-order logic with circumscription, by Doherty [6], that made automated reasoning possible. Many extensions since have turned TAL into a very expressive language capable of representing, among other things, actions with durations, context-dependent and non-deterministic actions, concurrency, and action side-effects.

But the most important feature of TAL might be its *occlusion* concept that serves as a flexible tool to deal with important aspects of the frame problem, which has long haunted logical approaches to AI. Properties and relations that may change over time are modelled by *fluents*, and the value v of a fluent f can be linked to a time point t on the time line using a function *value*$(t, f) = v$. Some agent a (denoted $self$ when doing the thinking) carrying out an action c during time interval i is specified by *Occurs*(a, i, c). The following formula relates a fluent f's value at the starting and ending time points of a time interval i, unless the fluent is occluded, as specified by *Occlude*(i, f):

$$\forall i, f\ [\neg \mathit{Occlude}(i, f) \rightarrow \mathit{value}(\mathit{start}(i), f) = \mathit{value}(finish(i), f)] \tag{1}$$

The role of circumscription is the minimization of action occurrences and occlusion to implement the blanket assumption that no unexpected actions occur and fluents' values persist over time. Exceptions are specified by explicit action occurrences and their occlusion of fluents they affect, thus releasing them from the frame assumption that their values remain unchanged. E.g., if the game character Djak was to sell the lumber he possesses in Figure 2, the fluent $owner(lumber)$ would be occluded during any interval that overlaps the interval during which the selling occurrs, and Formula 1 would not be applicable.

2.4. Reasoning and Planning

However, one of the most important forms of reasoning is not supported by the TAL framework as described above, namely proactive *planning* to achieve goals. E.g., if Djak's (modest) goal in life is the possession of lumber, he could reason that going to a shop and buying lumber is one possible plan to satisfy it. But his reasoning must allow the consideration of different actions before committing to any particular plan. Djak should not commence going to the store before considering whether the store actually sells lumber or not, since if it does not he might have to resort to an alternative sequence of actions such as cutting down some trees himself. However, committed knowledge about the set of actions is a prerequisite to automated reasoning using the circumscription account of TAL [7]. In contrast, we would like the set of actions to be a *consequence* of reasoning. This was accomplished in previous work [8] in a constraint logic programming setting. Rather than circumscribing a fixed set of actions, we use constraints to keep track of assumptions that depend on the set of actions, and reevaluate those assumptions when the set of actions change. The mechanism was cast as deduction, but the same principles are recast as abduction in a new first-order theorem proving setting described in Section 2.6. Thus equipped, Djak is both able to answer questions and plan his actions through automated reasoning with TAL.

2.5. Epistemics

Game agents, however, must face an additional complication when planning their actions. Inhabitants of the game world can not reasonably be assumed to possess complete knowledge of their entire world, and even if they did, the dynamic nature of game environments would quickly make this knowledge obsolete. The closed world assumption that is at the foundation of many classical planning systems is not applicable. Instead, an intelligent agent must reason with incomplete information and, significantly, plan to obtain additional information when needed. E.g., suppose another ANDI-Land agent, Keypr, owns a shop. Although Keypr is all out of lumber, he could sell Djak an axe to use to cut down a tree with. Being an intelligent and proactive fellow, Djak might come up with the following plan fragment (excluding the tree cutting part):

$$\begin{aligned}\exists i_1, i_2\, [&\mathit{Occurs}(\mathrm{self}, i_1, \mathrm{walk}(\mathit{value}(\mathit{start}(i_2), \mathrm{location}(\mathrm{keypr})))) \wedge \\ &\mathit{Occurs}(\mathrm{self}, i_2, \mathrm{buy}(\mathrm{axe}, \mathrm{keypr})) \wedge \\ &\mathit{finish}(i_1) = \mathit{start}(i_2)]\end{aligned}$$

Though, what if Djak does not *know* Keypr's location? The plan is still correct in the sense that if Djak executed it, the intended effects would manifest. The problem is that it is not *executable*. There is no way Djak can (willingly) walk to Keypr's location without knowing what that location is, but we have as of yet no means to express this additional *knowledge precondition*. What is needed is an epistemic logic that includes a notion of knowledge.

The most common way of introducing such a notion of knowledge is in the form of a modal operator *Knows* with a possible worlds semantics. This can be done while remaining in classical logic by encoding the possible worlds and the accessibility relation between them explicitly in the object language, as e.g. in Moore's pioneering work [9]. But these approaches are associated with some limitations that make them unsuitable as *general* frameworks of epistemic reasoning, as pointed out e.g. by Morgenstern [10]. She proposes an alternative treatment that introduces *Knows* as a "syntactic" predicate, which accepts quoted formulas as arguments. Quotation can be seen as an extreme form of reification where any formula can be turned into a term. It appears to be both simpler and more intuitive than possible world semantics in many contexts. Unfortunately, quotation is associated with the risk of paradoxes. While it is true that unrestricted quotation leads to the paradox of the Knower [11], there are methods for avoiding these problems (a particularly interesting one is Perlis' [12], which still allows for self-referential formulas). Our work, although adopting the syntactic quotation framework in anticipation of requirements of generality, has not yet proceeded far enough to utilize the additional expressivity afforded by syntactical treatments of knowledge over modal variants, a fact that guarantees consistency [13] and allows us to remain uncommitted as to which more general treatment to give preference to.

Equipped with the ability to represent knowledge explicitly we add a precondition to walking that one should know where the destination is. We can also make use of the *Knows* predicate in action effects, thereby formalizing knowledge producing actions and putting us in a position where planning for knowledge acquisition is possible. Adding an action for asking another agent (such as the player!) about a fluent's value enables Djak to come up with a plan that is both executable and that has the intended effect:

$$\begin{aligned}\exists i_1, i_2, i_3\, [&\mathit{Occurs}(\mathrm{self}, i_1, \mathrm{askValue}(\mathrm{player}, \mathrm{location}(\mathrm{keypr}))) \wedge \\ &\mathit{Occurs}(\mathrm{self}, i_2, \mathrm{walk}(\mathit{value}(\mathit{start}(i_3), \mathrm{location}(\mathrm{keypr})))) \wedge \\ &\mathit{Occurs}(\mathrm{self}, i_3, \mathrm{buy}(\mathrm{axe}, \mathrm{keypr})) \wedge \\ &\mathit{finish}(i_1) = \mathit{start}(i_2) \wedge \mathit{finish}(i_2) = \mathit{start}(i_3)]\end{aligned}$$

2.6. Natural Deductive Theorem Proving

Many agent architectures are built on a logic programming foundation, as was our previous work [8]. Logic programs incorporate some of the power of theorem proving while remaining relatively simple and allowing a high degree of control over the inference mechanism. But a fundamental limitation of Prolog is the assumption of complete knowledge, which, as we noted in Section 2.5, is unreasonable in complex computer games. In the interest of overcoming this limitation one can augment Prolog with meta-interpreters or other add-ons. Though when setting the sights for general intelligence it seems to us that augmenting Prolog will, over time, gradually approach general first-order theorem proving but in a roundabout and unnecessarily complicated way.

An alternative approach is to start with a first-order resolution theorem prover and complement it with special purpose modules that make some types of reasoning highly efficient. This is the method taken by the Cyc team, who have gone one step further and given up completeness in favour of efficiency and expressiveness [14]. Our (limited) experience with resolution suggests to us that it is not quite the natural fit with commonsense reasoning that one would hope. For example, the need to compile the knowledge base into clause form destroys potentially useful structural information that was previously implicit in the syntactic form of knowledge and rules, and the use of a single proof rule based on *reductio ad absurdum* could be incompatible with the defeasible reasoning that has turned out to be so important to commonsense reasoning [15].

Still, resolution completely dominates the field of automated theorem proving, but it is not the only contender. One particularly interesting alternative is *automated natural deduction*. Rather than compiling the agent's knowledge into clause form, such a theorem prover works with the "natural form" directly. And the rule set is extensible, thereby supporting the addition of special purpose rules, e.g. for defeasible reasoning. Moreover, whether the term "natural" is grounded in any relation between the deductive system and human reasoning is an exciting prospect explored by Rips, who argues a positive verdict [16].

In light of these considerations we have opted for our ANDI-Land inhabitants to "think" using an automated natural deduction theorem prover. Input formulas use the quantifier free form described by Pollock [17] and Rips [16]. This eliminates the somewhat cumbersome natural deduction rules for quantifier elimination and introduction while still preserving the knowledge base's natural form to a large extent. Most importantly, it provides the opportunity to work with unification and enables the use of *answer extraction* for question answering by binding answer variables to values as exemplified in Figure 2. Rather than a select few inference rules there is a set of *forward* rules (four at the moment), which are applied whenever they become applicable, and a set of *backward* rules (currently eleven of them), which are used in a goal-directed search for a proof. Finally, equality is dealt with through a system of rewrite rules, and temporal relations are added to a general temporal constraint network [18], exemplifying the use of special purpose reasoning mechanisms for efficiency.

A novel proof rule worthy of mention is a special abduction rule that allows relations from a set of *abducibles* to be assumed rather than proven, as long as doing so does not lead to inconsistency. This "natural abduction" rule forms the basis of the mechanism for non-monotonic reasoning and planning. As an example, consider the following natural deduction proof fragment (where the justifications in the right margin denote (P)remises, (H)ypotheses, the agents background (K)nowledge, and row numbers):

1	$\textit{value}(\text{12:00}, \mathrm{location}(\text{self})) = \mathrm{loc}(1, -1)$	P
2	$\textit{start}(i_{37}) = \text{12:00}$	P
3	$\textit{finish}(i_{37}) = \text{13:00}$	P
4	$\neg \textit{Occlude}(i_{37}, \mathrm{location}(\text{self}))$	H
5	$\textit{value}(\text{13:00}, \mathrm{location}(\text{self})) = \mathrm{loc}(1, -1)$	$1-4, K$
6	$\textit{Occurs}(\text{self}, i_{38}, \mathrm{walk}(\mathrm{loc}(0, 0)))$	H
7	$\textit{value}(\textit{finish}(i_{38}), \mathrm{location}(\text{self})) = \mathrm{loc}(0, 0)$	$6, K$
8	$\forall i\, [\neg \textit{Occlude}(i, \mathrm{location}(\text{self})) \rightarrow \neg \textit{Overlap}(i, i_{38})]$	$6, K$
9	$\neg \textit{Overlap}(i_{37}, i_{38})$	$4, 8$

The agent starts at the location with coordinate $\langle 1, -1 \rangle$ at noon, as in Row 1. Suppose the agent needs to remain at the same location at 1 p.m. One way of proving this would be to use persistence. The location fluent is only persistent if it is not occluded, and while the agent has no knowledge about whether it is occluded or not, $\neg$*Occlude* is an abducible and may thus be *assumed*. Rows 2-4 introduces a new interval constant and indicates the assumption using a natural deduction style vertical line in the margin. Suppose further that the agent, for some other reason, needs to visit location $\langle 0, 0 \rangle$. The only way of proving this would be if a walk action destined for that coordinate occurred. When planning, *Occurs* is also abducible, so the agent assumes such an action in Row 6. The effect on the agent's location is recorded by Row 7. Walking should occlude the location fluent, but instead of stating that the fluent is occluded in any interval that overlaps the walk action, Row 8 uses the contra-position, stating that any interval that has assumed the location to be persistent must not overlap with the action of walking. This triggers the forward modus ponens rule to produce Row 9, partially ordering the two intervals to avoid any conflict between the persistence of the agent's location, and the agent's moving about. The non-overlap constraint is automatically added to the temporal constraint network. If it is impossible to order i_{37} and i_{38} so that they do not overlap in any way, the network becomes inconsistent, and the prover needs to backtrack, perhaps cancelling the most recent assumption. The abduction rule thus enables both defeasible conclusions about the persistence of fluents and, simultaneously, planning of new actions.

The use of the contrapositive form illustrates a case where two logically equivalent formulas have different effects in the system due to their surface form. If the occlusion had been expressed as $\forall i\, [\textit{Overlap}(i, i_{38}) \rightarrow \textit{Occlude}(i, \mathrm{location}(\text{self}))]$, nothing would have triggered the non-overlap constraint. This, in turn, illustrates another important point. If the non-overlap constraint would make the temporal constraint network inconsistent, failing to trigger it could result in the agent failing to discover that one of its assumptions is unreasonable. This would not be a cause of unsoundness, since we are still within the sound system of natural deduction, but it might result in plans and conclusions that rest on impossible assumptions. A conclusion Φ depending on an inconsistent assumption would in effect have the logical form $\perp \rightarrow \Phi$, and thus be tautological and void. This is to be expected though since consistency is not even semi-decidable for first-order logic. The most we can hope for is for the agent to continually evaluate the consistency of its assumptions, improving the chances of them being correct over time, while regarding conclusions as tentative [15].

Another novelty is an execution rule linked to the agent's action execution mechanism, which is used to put plans into effect. Instead of sending the entire plan to a "dumb" execution module, we use the execution rule in "proving" that the plan is executed, thereby enabling the full reasoning power of the natural deduction prover to be

You steer your avatar Magni eastward and stumble upon another ANDI-Land character:

M⟩ Hello!
K⟩ Hello!
M⟩ Who are you?
K⟩ I am Keypr.
M⟩ What do you own?
K⟩ I own the axe.
M⟩ What is the axe's price?
K⟩ The axe's price is 5 gold.
M⟩ What is my wealth? (thinking)
M⟩ My wealth is 4 gold.
M⟩ Goodbye!
K⟩ Goodbye!

Dismayed by a sense of acute poverty, you continue to investigate the great forest. South-west lives another character, and as soon as he spots you, he comes running:

D⟩ Hello!
M⟩ Hello!
D⟩ Who owns the axe?
M⟩ Keypr owns the axe.
D⟩ What is Keypr's location?
M⟩ Keypr's location is 1 screen east and 2 screen north.
D⟩ Goodbye!
M⟩ Goodbye!

Before you have a chance to ask his name, he hurries northward. Curious, you follow. At Keypr's, you observe the following dialog:

D⟩ Hello!
K⟩ Hello!
D⟩ Sell the axe to me.
K⟩ OK.
D⟩ Goodbye!
K⟩ Goodbye!

Somewhat envious of the axe-wielding stranger, you follow him back and watch him start applying the axe to the trunk of a tree. Determined to know his identity you confront him:

M⟩ Hello!
D⟩ Hello!
M⟩ Who are you?
D⟩ I am Djak.
M⟩ What happened?
D⟩ I bought the axe from Keypr.
M⟩ What do you own?
D⟩ I own the axe.
M⟩ Goodbye!
D⟩ Goodbye!

While you watch eagerly as Djak strikes the tree, it suddenly disappears:

M⟩ Hello!
D⟩ Hello!
M⟩ What do you own?
D⟩ I own the axe and I own the lumber.
M⟩ What is the lumber's price?
D⟩ The lumber's price is 3 gold.
M⟩ Sell the lumber to me.
D⟩ OK.
M⟩ Goodbye!
D⟩ Goodbye!

Acting as a middle man, you revisit Keypr to try to sell the lumber:

M⟩ Hello!
K⟩ Hello!
M⟩ What is the lumber's price?
K⟩ The lumber's price is 6 gold.
M⟩ Buy the lumber from me.
K⟩ OK.
M⟩ What is my wealth (thinking)
M⟩ My wealth is 7 gold.

Intrigued by Djak and Keypr's limited displays of intelligence, but convinced that more must be possible, you vow to research AI in games!

Figure 3. This scenario from the ANDI-Land adventure game concept involves planned use of speech acts to satisfy the knowledge preconditions of buying an axe.

used in finding the exact action parameters for each step of the plan. Consider, e.g., Djak's plan to ask the player about Keypr's location in Section 2.5. Depending on the player's reply, further reasoning might be needed to convert this reply into a form that is suitable to pass as an argument to the action execution mechanism. This reasoning might depend on background knowledge about local geography and, in general, any amount of deliberation might be required during execution of a plan that involves knowledge acquisition, a fact respected by our execution proof rule.

3. A Dialog Scenario

Figure 3 illustrates all the components working together through example dialogs from ANDI-Land. The scenario revolves around our friends Djak and Keypr, from previous sections, but starts with the human player's avatar Magni in the middle of a thick forest. Djak's plan was automatically generated by the natural deductive theorem prover and its abduction rule, while the plan execution and all dialogs are straight from a running game session.

4. Related Work

In a spirit similar to ours, Amir and Doyle have proposed the use of text adventure games as a vehicle of research in cognitive robotics [19]. But instead of intelligent agents acting in supporting roles to enhance a human player's experience, they consider what challenges an agent would face if trying to solve the adventure itself. The agent would start out with severely limited knowledge, not knowing what actions are available to it, what fluents it should use to represent the environment, nor even the purpose of the game. These are some significant challenges, though they say a computer game "[...] allows us to examine them in a controlled environment in which we can easily change the problems to be solvable, and then gradually increase the difficulty step by step". However, their proposition does not endorse any specific formalism or system.

Shanahan [20] proposes a logicist agent architecture that incorporates planning, perception, and a sense-plan-act loop, all formalized in the Event Calculus and executed through proof using abductive logic programming. The unified approach makes it possible to proactively deal with unexpected percepts in a robotic mail delivery domain, due to humans unpredictably blocking pathways by closing office doors. The robotic agent is able to intelligently adapt its behaviour by first reasoning about all percepts using abductive proof, forming explanations for sensor values that deviate from expectations in terms of actions by other agents or humans, and then adapting its plans to incorporate the new knowledge. Hierarchical planning is accomplished through the same abductive proof mechanism and allows timely reactions by only instantiating the abstract plan enough to figure out a first action, while leaving the rest a sketchy idea of how to achieve the goal.

Pollock goes further towards general intelligence and differentiates between goal-oriented agents, that solve tasks for which a metric of success can be defined, and anthropomorphic agents, that solve tasks that are too complex for it to be possible to identify such a metric [15]. Such agents must be based on a "general theory of rational cognition", and Pollock's OSCAR agent architecture is an attempt to embody such a theory in an implemented system. The central component is a natural deduction theorem prover for first-order logic that is capable of planning, reasoning about percepts and attaching certainty factors to premises and conclusions. But its most important feature is the mechanism for defeasible reasoning that can be used to deal with default reasoning and the frame problem. Unlike most other formalisms, which are only applicable to problems conforming to explicit restrictions that ensure computability, Pollock's anthropomorphic architecture can be applied to any problem. The inference engine reports solutions based on defeasible assumptions, while a search for evidence contradicting these assumptions continues, for which there can be no guarantee of termination.

Wang's NARS system is similar in that the underlying assumption is the lack of knowledge and resources sufficient to give optimal answers or even any correctness guarantees [21]. Instead, the system continually evaluates the available evidence and may "change its mind" about the best answer to a given query. NARS is based on a novel *categorical logic* that differs significantly from classical first-order logic, incorporates uncertainty, and deals with conflicting evidence at a fundamental level. While clearly aiming towards general intelligence, results to date seem limited to small benchmark problems.

One example of the relatively recent surge of interest in the use of computer games for AI research is the Soar/Games project. They report uncovering new research challenges after coupling the Soar artificial general intelligence architecture to Quake 2 and

Descent 3 [22]. Their emphasis is on generality in their attempts to build reusable rule bases for agent behaviour. Laird's and van Lent's enthusiasm for the use of computer games in AGI research is evident in their paper "Human-level AI's Killer Application: Interactive Computer Games" [23].

Finally, the 1996 computer game Creatures is an example of AI from the game industry rather than of academic origins. Its artificial life forms use neural net "brains" that can be trained through interaction with a human player, learn from interaction with their simulated world, or even from other creatures [24]. The success of Creatures is an affirmation of the possibility of incorporating AI technology into a commercial computer game.

5. Conclusions

We hope to have given the impression that our game concept is far from complete. On the contrary, when working with games interesting problems abound, and many of them call for new research in artificial general intelligence. Some old but still open questions that figure in our work are how to include perception, reasoning, planning, execution, and failure recovery in an integrated agent architecture, what to do about conflicting information, and how to deal with the accumulation of perceptions and knowledge in persistent agents without their reasoning slowing down to a crawl. ANDI-Land is fundamentally a multi agent setting and could involve cooperation between multiple agents, delegation of goals, and intelligent use of communication. These (and more) topics have concrete instantiations in the structure of the game environment that make them easier to think about, discuss, and hopefully to solve.

Traditional AI benchmark problems play an important role in clearly highlighting specific difficulties that any sufficiently general AI system will have to address. Games can serve to complement them by forcing an integrated view of autonomous agents in complex environments, and they possess many positive attributes such as ease of access for both researchers and their peers, variable challenge level ranging all the way from simple puzzle games to wide coverage natural language understanding, and the possibility for applications in the future commercial game industry where academic AI technology has so far failed to migrate (the prototypical exception being A* search).

The demand for human-like computer characters is by itself incentive to study all the key technologies needed for artificial general intelligence, making games an excellent test bed for AGI research. Even some forms of self-awareness would seem to be desirable to agents acting as if they were "real" live inhabitants of some fictional reality game world. Such a setting is a sort of Turing test where human players are not necessarily aware of which characters are artificial and which are other humans. It seems to us that research on game AI could function as a much needed road map towards the fields original vision.

Acknowledgements

This work is supported in part by the National Aeronautics Research Program NFFP04 S4203 and the Strategic Research Center MOVIII, funded by the Swedish Foundation for Strategic Research, SSF.

References

[1] Patrick J. Hayes. The naive physics manifesto. In Donald Michie, editor, *Expert Systems in the Micro-Electronic Age*, pages 242–270. Edinburgh University Press, 1979.

[2] Ann Copestake and Dan Flickinger. An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation LREC 2000*, Athens, Greece, 2000.

[3] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition, 2003.

[4] Stuart M. Shieber. A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 614–619, Budapest, Hungary, 1988.

[5] Erik Sandewall. *Features and Fluents: The Representation of Knowledge about Dynamical Systems*, volume 1. Oxford University Press, 1994.

[6] Patrick Doherty. Reasoning about action and change using occlusion. In *Proceedings of the Eleventh European Conference on Artificial Intelligence ECAI'94*, pages 401–405, 1994.

[7] Patrick Doherty and Jonas Kvarnström. *The Handbook of Knowledge Representation*, chapter 18. Elsevier, 2007. To appear.

[8] Martin Magnusson. *Deductive Planning and Composite Actions in Temporal Action Logic*. Licentiate thesis, Linköping University, September 2007. `http://www.martinmagnusson.com/publications/magnusson-2007-lic.pdf`.

[9] Robert Moore. Reasoning about knowledge and action. Technical Report 191, AI Center, SRI International, Menlo Park, CA, October 1980.

[10] Leora Morgenstern. A first order theory of planning, knowledge, and action. In *Proceedings of the 1986 conference on Theoretical aspects of reasoning about knowledge TARK 1986*, pages 99–114, 1986.

[11] David Kaplan and Richard Montague. A paradox regained. *Notre Dame Journal of Formal Logic*, 1(3):79–90, 1960.

[12] Donald Perlis. Languages with self-reference I: Foundations (or: we can have everything in first-order logic). *Artificial Intelligence*, 25(3):301–322, 1985.

[13] Jim des Rivières and Hector J. Levesque. The consistency of syntactical treatments of knowledge. In *Proceedings of the 1986 conference on Theoretical aspects of reasoning about knowledge TARK 1986*, pages 115–130, 1986.

[14] Douglas Lenat. CYC: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[15] John Pollock. Rational cognition in OSCAR. In Nicholas Jennings and Yves Lesperance, editors, *Intelligent Agents VI: Agent Theories, Architectures, and Languages*, volume 1757 of *Lecture Notes in AI*, pages 71–90. Springer Verlag, 2000.

[16] Lance J. Rips. *The psychology of proof: deductive reasoning in human thinking*. MIT Press, Cambridge, MA, USA, 1994.

[17] John Pollock. Natural deduction. Technical report, Department of Philosophy, University of Arizona, 1999. `http://www.sambabike.org/ftp/OSCAR-web-page/PAPERS/Natural-Deduction.pdf`.

[18] Itay Meiri. Combining qualitative and quantitative constraints in temporal reasoning. *Artificial Intelligence*, 87(1–2):343–385, 1996.

[19] Eyal Amir and Patrick Doyle. Adventure games: a challenge for cognitive robotics. In *AAAI'02 workshop on Cognitive Robotics*, 2002.

[20] Murray Shanahan. Reinventing shakey. In *Logic-Based Artificial Intelligence*, pages 233–253. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[21] Pei Wang. The logic of intelligence. In Ben Goertzel and Cassio Pennachin, editors, *Artificial General Intelligence*, pages 31–62. Springer, 2007.

[22] John E. Laird and Michael van Lent. Developing an artificial intelligence engine. In *Proceedings of the Game Developers' Conference*, pages 577–588, 1999.

[23] John E. Laird and Michael van Lent. Human-level AI's killer application: Interactive computer games. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 1171–1178, 2000.

[24] Stephen Grand, Dave Cliff, and Anil Malhotra. Creatures: Artificial life autonomous software agents for home entertainment. In *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 22–29, 1997.