Situated Language Understanding as Filtering Perceived Affordances

Peter Gorniak and Deb Roy
MIT Media Laboratory
20 Ames St.,Cambridge, MA 02139 USA
{pgorniak,dkroy}@media.mit.edu

Abstract

We introduce a computational theory of situated language understanding in which the meaning of words and utterances depend on the physical environment and the goals and plans of communication partners. According to the theory, concepts that ground linguistic meaning are neither internal nor external to language users, but instead span the objective-subjective boundary. To model the possible interactions between subject and object, the theory relies on the notion of *perceived affordances*: structured units of interaction that can be used for prediction at multiple levels of abstraction. Language understanding is treated as a process of filtering perceived affordances. The theory accounts for many aspects of the situated nature of human language use and provides a unified solution to a number of demands on any theory of language understanding including conceptual combination, prototypicality effects, and the generative nature of lexical items.

To support the theory, we describe an implemented system that understands verbal commands situated in a virtual gaming environment. The implementation uses probabilistic hierarchical plan recognition to generate perceived affordances. The system has been evaluated on its ability to correctly interpret free-form spontaneous verbal commands recorded from unrehearsed game play between human players. The system is able to "step into the shoes" of human players and correctly respond to a broad range of verbal commands in which linguistic meaning depends on social and physical context. We quantitatively compare the system's predictions in response to direct player commands with the actions taken by human players and show generalization to unseen data across a range of situations and verbal constructions.

1 Introduction

Language is often used to talk about the world. We easily refer to objects using expressions such as "door" or "the blue thing for making pizza that I gave you yesterday." The relationship that holds between language use and the world, variously and differently described by such terms as *reference*, *intentionality* and *aboutness*, has long been a central topic of study by linguists, psychologists and philosophers. Most theories posit an intermediary step between words and the world, usually labelled as a *concept*. However, theories differ on even the most fundamental matters such as whether a concept is a mental construct of the language user, or an independent abstract entity. Very few of these theories have been stated in computational terms amenable to mechanistic implementation and empirical evaluation.

There are two intertwined aspects of any theory of concepts: a description of the internal structure (if any) of a concept, and an account of how this structure comes to be about the world. In many cases, theories focus on the first and neglect the second, or at best give a vague answer to the second. Why is this? We suggest that at issue is the role of autonomy, an issue that is rarely considered a central aspect of conceptual structure. People interact with their immediate world for their own reasons, and maintain concepts about this world for their own functional ends. However, humans neither try to fully internalize a complete representation of the world Clark (1998), nor do they individually maintain all possible concepts of a community Putnam (1975). A theory of human concepts needs to support and explain the same type of autonomy: it must be able to generate functional concepts about a person's experience, yet rely on the environment and community to maintain most of the state of the world and shared meaning in general.

In short, theories of concepts often neglect to specify how the proposed mental structures attach to the world and gain intentionality. These theories may define words in terms of other words or word-like symbols and call these definitions concepts. While

such theories contribute much to our thinking about the possible mental structures that are concepts, here we present an approach to concepts that emphasizes the importance of intentionality and tightly couples the internal structure of concepts with their need to be about the world. Any theory that draws a clear line between concepts and the world leads to "detached" concepts that lose their intentionality. We provide a theory that avoids drawing any such line. Instead, it proposes that each element of a concept must make a prediction about the world, thus crossing over from the mind to the world. Every concept thus becomes both a property of the language using system, and of its relation to the embedding world. These structural elements are called *perceived affordances*, yielding a theory of *Affordance-Based Concepts*.

Based on the Affordance-Based Concept (ABC) theory, we introduce a computational model that employs plan recognition as a mechanism for finding and ranking the perceived affordances of a person engaged in co-operative tasks. Situated language interpretation is modeled as a process of filtering perceived affordances. In effect, the complete meaning of linguistic expressions is only understood when words are meshed with the situation in which they are used. To evaluate the model, we describe an implementation of the model that interprets situated language collected from people playing a multiplayer computer game. We designed a computer game in which two human players explore a set of interconnected rooms via avatars in order to cooperatively solve puzzles. Due to the nature of the puzzles, players must co-ordinate their actions using language. During game play, all verbal communication and situational context are recorded, providing a rich record of communication in context. The implementation of the model uses a probabilistic hierarchical plan recognizer in the form of an Earley parser to analyse the actions of human players as a basis for understanding commands produced by the players. At any point in time within a game session, the plan recognizer processes all observed player actions in order to predict the likely actions of each player based on a priori knowledge of the goals and likely strategies in the game.

These hierarchically organized and ranked predictions stem from a combination of the structure of the environment, the player's past interaction with this environment, and the goals of the player. They provide a uniform representational substrate for modeling objects, geographical spaces and actions in terms of their functional significance relative to an agent. The crucial contribution of this paper lies in treating these predictions as the building blocks of concepts underlying (grounding) word meaning. To do so, we define lexical items as filters on the space of all such structured predictions, which are computational instantiations of perceived affordances. During linguistic parsing these filters compose to form more complex concept definitions, and the concept associated with a whole utterance is the set of perceived affordances selected by its composite filter. We have evaluated the implemented model for its ability to understand freeform directives issued by players in the game environment by comparing its predictions to the actions taken by human players in response to the same commands. Quantitative results of the evaluation show that the model accurately predicts how human players respond to spoken commands issued by their human partners, demonstrating the viability of our approach for modeling the interpretation of context-dependent language on the basis of perceived affordances.

The Affordance-Based Concept addresses the intentional link between language users and the world by treating predicted interactions as the basic building block for conceptual representation. By doing so, it also yields a substrate that addresses many other demands of a theory of concepts that are often only considered individually. For example, perceived affordances are naturally ranked according to typicality and context, addressing the prototypicality effects often exhibited by human concepts. Similarly, the richly structured predictions made by Affordance-Based Concepts naturally lend themselves to conceptual composition. In fact, as we will show in the implementation presented here, conceptual composition can be cast as a filtering process on the complete set of affordances a situation yields. Finally, hierarchical sets of affordances

give an intuitive framework for performing conceptual generalization and abstraction.

The remainder of this paper is organized as follows: Section 2 motivates the need for a new theory of concepts. Section 3 introduces the Theory of Affordance-Based Concepts. Section 4 describes our implementation of an instance of ABCs, and Section 5 presents evaluation studies performed with this implementation. Finally, Section 6 provides a brief summary and suggests some useful directions for future work.

2 Background and related work

A growing trend in Cognitive Science has cast language understanding as an embodied and dynamically contextualized process (Duranti and Goodwin, 1992; Glenberg, 1997; Barsalou, 1999). Rather than traditional views of concepts, such as those summarized by Laurence and Margolis (1999) and Prinz (2002), these theories emphasize the importance of the language user's possible interactions with the world. They argue that mental representation is based on sensing and acting in the world, and that conceptualization and the ability think about acting in the world are inseparable. Evidence for the cognitively tight coupling of these processes during language comprehension in humans comes from both neurological and behavioural studies. Pulvermüller et al. (2001) shows activity in the motor regions corresponding to body parts during verb understanding, where the active regions correspond to the body part involved in the action specified by the verb (e.g. the mouth for "talk" and the leg for "walk"). These studies suggest strong links between language understanding and motor control in the brain, and may even hint at a thesis like the one presented here: that language is understood directly in terms of action planning representations. Specific support for our project here comes from Glenberg and Kaschak (2002), Zwaan (2003) and others who show that language comprehension involves the generation of embodied mental representations and simulations to make predictions and generate hypotheses about novel contexts.

So far, there have been few computational models of interactionist (Bickhard, 2001) theories, partly due to the difficulty of building a machine that is sufficiently embedded in a non-trivial world to simulate language understanding. In this paper, we introduce a specific interactionist theory of language understanding and describe an implementation that leverages the theory to understand spontaneous human commands in a dynamic virtual environment. By doing so we hope to provide a first instance of a situated system that understands language directly in terms of perception and action representations. While the type of embodiment in our system differs in that we use computer games as an easily sensed yet socially complex embedding situation, we believe that the lessons learned from attempting to perform language understanding on unconstrained human speech in terms of plan models and predicted actions apply directly to more fine-grained sensory and motor systems embedded in the real world.

While the theory introduced in the next section is general in nature, it should be seen as a proposal and outline with partial support from the implementation and studies that follow in the subsequent sections. Many of the linguistic aspects of the implementation are simple, and blatantly ignore discourse history to focus on taking into account intentional and physical history. This is a deliberate decision, because discourse history has been proposed as a way to analyse intentions and recover plans before (Allen and Perrault, 1980; Litman and Allen, 1984; Stone, 2001), whereas intentional and physical history has been left unaddressed. This decision means that treatment of anaphora, and linguistic analysis in general are simplistic in our current work in favour of emphasizing the connection to the situation captured by physical and intentional (teleological) analysis. Furthermore, while we evaluate our implementation quantitatively as compared to human performance, this is a general measure of the model's viability and does not validate the implementational details as cognitively real in humans. It does, however, suggest that the theory can be the basis for predictive computational models of situated human reasoning and language use. Our results emphasize the predictability

of the situation through plan recognition, and add language as a further filtering factor on top of the possible actions dictated by the affordances of the actors. The increase in predictability through taking into account utterance is significant, but the good performance of a hierarchical plan recognizer on the actions of players alone driver home the point that much of understanding situated language comes from modelling the situation rather than relying solely on the words.

2.1 Existing computational approaches

Winograd's SHRDLU was one of the first situated language understanding systems (Winograd, 1970). In fact, it still stands today as one of the most sophisticated ones, without much followup work to surpass it. SHRDLU uses a relatively static, symbolic representation of the situation and keeps the user's plans distinct from the physical (logical) situation. Plans in SHRDLU are only implicitly encoded in the form of procedures applied due to the language used. In the work presented here, the situation includes a noisy estimate of the language user's plans in a highly dynamic situation. The situation thus requires categorization and representation in order to be tied to language, which in turn requires interaction and prediction on the part of the language understanding system. SHRDLU, on the other hand, thus commits to the problematic assumption of the separation of linguistic concepts from the world they are about that was discussed in the previous sections.

Chapman's work describes a semi-autonomous agent in a game that follows simple linguistic instructions (Chapman, 1991). While touching on elements of interaction and planning, this work de-emphasizes the linguistic component in favour of focusing on a model for interactivity. This article expands on those ideas by introducing a strong language element to cast the elements of interactivity and prediction themselves as the conceptual basis for a linguistic system.

In our own work, we have introduced both visually situated language understanding

systems (Gorniak and Roy, 2004) as well as interactive conversational robotic systems (Hsiao et al., 2003). While this prior work focused on grounding words in visual perception and developed methods for linguistic parsing compatible with sensor-grounded language understanding, that work did not address teleological aspects of semantic grounding. All of the utterances understood by these systems consist of visually referring expressions, each uttered with the single purpose of communicating its referent, and in the case of the robotic models, performing simple manipulation actions on those referents. Here, we propose that determining the purpose behind an utterance is of prime importance to understanding its meaning. Along similar lines, our robotics work has led Roy to propose a theory for grounding linguistic concepts in physical interaction (Roy, 2005). That work complements that presented here as a proposal for linguistic meaning based on interactions with the world at a far more detailed and fine grained level of physical (sensory-motor) experience than considered here. In the future, we hope to give an account that encompasses both the level of representation discussed there as well the more abstract and broader interactions under investigation here.

Following the idea that human cognition uses scripts and plans to model the detailed and more abstract affordances of a situation and to reason about language and action (Schank and Abelson, 1977), the implementation introduced here relies on hierarchical plan recognition based on observing a sequence of actions given a generative model to perform planning. While much work and many systems exist that produce hierarchical plans given goals, especially in the popular framework of HTN (Hierarchical Transition Network) planning (Erol et al., 1994; Nau et al., 2003), there exists considerably less work on applying similarly expressive and structured models to probabilistic plan recognition. Probabilistic parsers have also been used in other plan recognition systems (Bobick and Ivanov, 1998; Pynadath and Wellman, 2000), and in addition the use of Abstract Hidden Markov models has been suggested, which does not produce the type of modularity required here (Bui et al., 2002). A promising new candidate is

Geib and Goldman's execution model based plan recognition framework, which maintains pending action sets that could be used instead of the Earley state sets on which the work here is based (Geib and Goldman, 2005). The advantage of a plan library based approach using HTN style methods would be a better parametrization of the plan library, and thus easier creation of and reasoning about possible plans without a need to generate all possible actions explicitly.

In addition to work explicitly related to planning and plan recognition, some authors have proposed other predictive representations for learning and acting. Drescher (1991) uses structural elements that assemble themselves into hierarchies while interacting with a simple world. While strongly related to the notion of affordances used here, this work does not connect to language and it is unclear how it scales to a problem of the size tackled in the studies presented in later sections. The work does contain many insights into how affordances might be learned and organized by interacting with a situation. More recently, Littman et al. (2001) have proposed a stochastic representation of an agent's state based upon predictions of the outcome of a series of actions the agent could take. These proposed representations are promising candidates for computational instantiations of affordances. However, in the implementation presented here we rely on a known plan recognition paradigm that is suitable for the complexity and structure of the scenario investigated. In other situations, for example in the robotic case where action and perception are unreliable, but plans may be less complex, these other ways of working with affordances may be more suitable.

Finally, there exists work on computationally modelling affordances more abstractly as a theoretical tool to explore linguistic mechanisms (Steedman, 2002), as well as in a non-linguistic setting to model a robot's interactions with the real world (Stoytchev, 2005). While both research areas are relevant to the work presented here, they do not address the need for a theory linking perceived affordances to linguistic concepts in an implementable fashion. They do, however, suggest other ways to encode and reason

about affordances, which could enrich the work presented here in the future.

3 The ABC theory

The theory of Affordance-Based Concepts provides a solution to the problem of concept detachment outlined in the last section. The nature of its basic units, perceived affordances, ensures that it provides the linked triplet of perception, representation and prediction at the most basic level. The theory therefore produces concepts connected to the concept user's world in the strongest possible sense, doing away with problems of passive perception and lack of normativity. We describe the theory in this section, and a computational instantiation that captures many aspects of the theory in the next section.

3.1 Affordances

In the previous section we highlighted the need for mental structures that integrate aspects of perception, representation and prediction. This section introduces the notion of *perceived affordances* to fulfill this need.

3.1.1 Affordances and perceived affordances

The term *affordance* was coined by Gibson (1977). Working in the field of visual perception, Gibson was responding to what we have called correspondence theories of perception. Rather than focusing on image-like representations that are similar to, or correspond to, the light information impinging on the retina, he proposed that perception encodes what the external world affords the perceiver. Thus, extended surfaces are perceived to provide support for walking on, if the surface is of an appropriate size relative to the perceiver and sturdy enough to hold the perceiver's weight, and the perceiver is actually able to walk. However, affordances are not necessarily perceived.

They are relationships between an actor and the embedding environment that hold independently of the actor perceiving them. We therefore distinguish between affordances and perceived affordances – those that the actor perceives and thus mentally represents.

Affordances are unique in that they are primitive aspects of the physical makeup of the world that are neither objective nor subjective. They span the objective-subjective boundary. There is no sense in which a chair affords sitting on, unless we think of someone who is doing the sitting relative to the chair: the sitter must be of the right size and weight to get onto the chair and be supported by it. Thus, a human sized chair affords sitting for an adult human actor, but not for a horse. A chair might also afford picking up and throwing for adult humans, but not if it is bolted to the floor. The set of all affordances of an individual in an environment contains all possible interactions of the individual with the environment. This set is not identical to the set of perceived affordances of the individual. Neither is the set of perceived affordances a subset of the set of all affordances, because the individual may be wrong about what the environment affords it. If a person attempts (and fails) to sit on a cunningly designed object that looks like a wooden chair but is actually made out of paper, the person perceived an affordance that did not actually exist.

Perceived affordances, as we have described them here, fulfill the requirements of a representation we arrived at in the last section: they are the product of perception of the world, they encode some aspect of the structure of the world relative to the perceiver, and they predict a possible interaction between perceiver and world. By implying a prediction, they can be falsified. However, some incorrectly perceived affordances may well never be falsified. If in the preceding example the perceiver decides not to use the prediction and sit on the paper chair, the perceived affordance, though wrong, will never be falsified. The distinction between true and false perceived affordances is not necessarily a binary one. Agents may have degrees of belief in the validity of perceived affordances, and in fact the implementation presented in Section 4 maintains

exactly such degrees of belief.

3.1.2 The structure of perceived affordances

An affordance concerns possible interactions between an actor and an environment, and an interaction necessarily includes a temporal element. Given a joint state of actor and environment an affordance is a possible future interaction and thus concerns at least two points in time: the current moment, and the future point of interaction, which may also be extended in time. Recall that affordances in general are not representations, they are sets of possible interactions and thus exist simply because of the physical state of the system that includes the state of the environment and the state of the actor – in short, because of the *situation*. Here, we are more interested in perceived affordances, which are mental representations, and thus must be finitely describable without requiring a complete description of the situation. Due to what Smith calls the *flex and slop* of the world (Smith, 1996), namely the property that in the macroscopic world of everyday experience effects die off with distance, it is generally possible to produce a state description of the situation that suffices to make good predictions without describing it completely. The *Markov Assumption* of a state in a model proposes much the same thing: that it is possible to predict the future behaviour of the system given only an encoding of its current state. Perceived affordances thus include an encoding of some aspects of the current situation. There are many examples of such state encodings in current literature concerning decision making for artificial agents (Boutilier et al., 1999).

In addition to a state encoding, an affordance predicts a possible interaction. This prediction may be representationally explicit, such as a list of possible ways to pick up a cup, or it may be implicit, such as an encoding of the cup's geometry together with a model of possible hand movements and configurations. Both representational styles have their place at different levels of affordances. It seems unlikely that a list is a good

way to represent the myriad ways to pick up a cup, but it may serve well for thinking about what to have for breakfast. In general, as Minsky (1985) points out, there are many styles of representation that are amenable for different ways of thinking about different things, or thinking differently about the same thing. As long as representations encode state and serve to predict possible interactions, they are candidates for affordances.

An affordance addresses the possible action prediction problem at a single level of representation. In the previous example, the possible ways to pick up a cup and the choice of breakfast foods are on very different levels of representation. They are connected, however, in that a possible breakfast choice may include pouring a cup of milk, and thus picking up a cup. To make mental representation feasible it is important to keep these levels of affordances related yet distinct. Keeping them distinct allows one to reason on a single level, to achieve more concise yet still approximately Markovian state encodings and to employ the representation and reasoning methods that are best for that level. Keeping them loosely connected, on the other hand, allows for predictions that span levels and lets one fill in the details of high level plans, creating a hierarchy of perceived affordances.

3.2 Affordance-based concepts

3.2.1 Concepts of objects

Note that so far we have not invoked the notion of objects per se – perceived affordances are about the structure of the world that can be exploited to make predictions. This structure can be below the level of everyday objects, for example when it concerns the geometry of a graspable surface, which may or may not be part of a larger structure that we usually label "doorknob." Having replaced the notion of objects with the notion of structural elements called affordances, we can now re-introduce objects as *bundles of affordances*. A doorknob yields a set of interactions, as determined by

its physical properties and the agent's abilities. When we engage in an active process of representation to distinguish objects within the structure of the world, we carve out a set of local affordances in the world and consider it an object. This process is not arbitrary, however, as it exploits the pre-existing structure of the world, including our own abilities. Thus while concepts of objects are the product of our perception, representation and actions, and while we may decide to cut up the world into different sets of objects at different times, we are externally constrained in our object categorizations by our own structure and that of our environment.

In the following studies this unified representation of objects as bundles of perceived affordances lets us capture the situation in terms of its possible functions for the agent. For example, a door is represented by the uses an agent might have for it, such as unlocking it, opening it or walking through it. This lets us interpret language by connecting it to a representational substrate that already includes predicted actions and abstractions and thus turns understanding into a filtering process on this substrate. For example, "open the door" selects a subset of the perceived affordances of the listener in his or her present situation that involve opening available doors.

3.2.2 Concepts and composition

Concepts of objects are instances of the more general class of structures we call concepts. Each concept is a bundle of perceived affordances. In addition to representing concrete everyday objects, concepts can represent sets of structures in the world not limited to a single agent and object. Allowing arbitrary bundles of affordances gives the Affordance Based Concept theory a unique representational power, but the use of affordances imposes limits as it is constrained by the structure of subject and environment. One aspect of this power is the ability to represent abstraction. For example, the command "let me into the next room" in our studies selects a more abstract interaction of changing rooms that at lower levels expands out into the listener unlocking or de-

stroying the door to the next room, or pulling the correct lever to open it, followed by the speaker moving to the next room. This is also an example of concept composition in which the filter functions of lexical items are combined during the linguistic parsing process. Thus, while "room" selects sets of affordances available in any single room in the virtual environment, "next room" selects only those requiring exactly one room change by the speaker.

In this paper we specifically address situated language. Here *situated* is used in the sense that one cannot understand this type of language without knowledge of the speaker's immediate physical and intentional context. Other kinds of language are less immediately situated, such as the displaced language used by someone relaying a story about a recent trip, or a description of a fictional event in a book. While not immediately situated, these types of language are still embedded in a strong intentional context created by the story as well as background knowledge of speaker and listener. In that manner, the methods presented in this paper for modelling intentional context via plan recognition and mental simulation have relevance to understanding displaced language.

While not covered by the implementation presented later in the paper, theoretically ABCs also extend to non-physical concepts. Some labelled concepts have intuitively clear constraints on interaction possibilities associated with them, such as "mass" or "ease of use". But we believe there is even a story of levels of affordances to be told about a concept like "freedom". As said, we do not claim that a single type of mental representation suffices to account for all possible levels and types of affordances. The following sections introduce one type of framework to maintain hierarchical levels of affordances and to perform language understanding in terms of these affordances. Some meanings of a word like "freedom" might be representable in that framework, within the limited domain addressed - being unable to leave a room is certainly an aspect of the puzzle the players encountered and our framework captures. We will need

to develop a representationally richer framework that relies less on explicit generation of affordances to cover the full human meaning of a word in terms of affordances. Mental simulations, plans and affordances, however, are likely to still be a part of any such expanded framework, as shown by work on human understanding of abstract language (Glenberg and Kaschak, 2002).

4 An implementation of the ABC

We now turn to a computational implementation of the Affordance-Based Concept theory. The implementation described here demonstrates all of the main aspects of ABCs:

- Predictive units that capture the possible interactions at a particular level of abstraction;
- A hierarchy relating affordances at different levels of abstraction;
- A mechanism to track the current situation in terms of perceived affordances of all levels;
- A set of functions to form and combine concepts from the past and current perceived affordances;
- The necessary relationships linking words and grammatical constructions to ABCs to decode language into concepts given a situation.

As a first step in exploring the space of possible ABC models, the implementation is limited in scope. While its mechanisms are general and should be transferable to many domains, it achieves coherent treatment of hierarchical perceived affordances through uniformity: each affordance is represented in the same way, namely as a single structured prediction of a probabilistic plan recognizer. While this particular representation

is useful for a number of problems and domains, we claim in no way that perceived affordances should actually be uniformly represented.

Another caveat applies with respect to the high-level symbolic form of the input accepted by the system we are about to present (i.e. high-level game events such as player movement and object manipulations, not raw visual or other sensory input). There are many different levels of granularity at which affordances can be modeled. One such level encompasses low-level, fine-grained sensory inputs such as camera pixels that need to be aggregated and interpreted over time as well as raw motor outputs. We acknowledge the need to address this layer of granularity and have proposed models that do so elsewhere (Roy et al., 2004; Roy, 2005). The difficulties of sensing and acting in the real world, however, confine the implementation of these proposals to controlled real world scenarios and simple linguistic and social interactions. By turning to computer games as a research platform we are able to focus on more complicated linguistic and co-operative social interactions by greatly simplifying sensing and acting. We believe, however, that many of the problems of modeling affordances exist independently of the granularity of input and output. For example, a model must be able to generate affordances in a new situation, but constrain the production of affordances by known limitations of the agent in relation to its environment. At a low level, this means acknowledging the restrictions imposed by the agent's body and the properties of the physical environment, and evaluating the affordances of novel situations in the context of these restrictions. At the level we address here, this means taking into account properties of how the game world works, and how the agents can affect it during their search for a puzzle solution. The abstract affordance representation we employ here provides a framework to generate the relevant affordances for a situation, and thus applies both to lower level sensory input and motor output as well as the input and output of the game setting.

4.1 Hierarchical plans

In the implemented system, the structure of perceived affordances hinges on the notion of a hierarchical plan. A plan is a sequence of one or more steps an agent takes or considers taking. A hierarchical plan is a plan in which a top level node is expanded into sequences of lower level nodes each of which in turn may expand into yet lower level nodes. The leaves of the plan structure form a non-hierarchical plan of concrete actions the agent can actually take. Humans explicitly or implicitly maintain hierarchical plans all the time, such as when planning to buy milk, which expands into going to the store and purchasing milk, which in turn expands into walking to the car, getting in the car, driving to the store, and so on. Hierarchical plans have the advantage of making some independence assumptions: if your goal is to buy milk, how you get to the store does not matter – you could walk, drive or bike. This independence assumption is a powerful tool that buys computational tractability and an easy method to leverage substitutability of sub-plans. However, if one suspects that this independence assumption does not hold, context dependence can be achieved by providing distinct symbols that occur in distinct contexts. For example, instead of a context-independent sub-plan to get to the store, one would employ two context-specific sub-plans for getting to the store, one to do so in a covered manner (to be used if it is raining) and one to get to the store in other cases.

Plans and planning are intimately related to perceived affordances. In fact, perceived affordances are the basis for planning. The current situation must contain an affordance predicting one could go buy milk, as otherwise one would not plan for it. Similarly, someone will only consider driving to the store if that persons actually has access to a car (which includes planning to obtain access to a car). Perceived affordances are thus not the elements of a plan, but at each step they are the possible choices a planner faces when making decisions. Thus each planner must maintain sets of affordances to perform its planning, and a hierarchical planner maintains hierarchical trees

of affordances.

Planning and plan recognition are tightly coupled activities. As soon as there are two agents involved in a plan, the two activities become one and the same - to plan for two people, each individual must recognize the other individual's plan and incorporate it. In the implementation presented here, we focus on hierarchical plan recognition, because it allows us to model two human players' intertwined affordances, model their concepts and understand their language even though we cannot control their actions or perceptions directly as would be possible with an artificial agent. As we will see, however, elements of planning will be necessary to understand language as well, and when building an artificial language using machine, planning takes central stage. We will outline how to proceed to a fully autonomous language using machine after describing the computational modelling of the ABCs of human speakers via plan recognition.

4.1.1 Probabilistic context free parsing

The implemented representation of perceived affordances is based on methods of context free parsing, which we now briefly introduce. A Context Free Grammar (CFG) is described by a set of rules of the form $X \rightarrow AYZ$ where X is a single symbol called a non-terminal, and AYZ is a string of symbols. Any symbol in *AYZ* (the *tail* of the rule) that does not appear on the left side of an arrow in the set of rules (is not the *head* of a rule) is called a terminal. Rules should be interpreted as re-write rules: X can be re-written as AYZ (or AYZ as X, depending on the direction of analysis). In a context free grammar the fact that every rule can only have one non-terminal as its head enforces that when X occurs in the tail of a rule, it can be replaced with AYZ independently of what symbols occur to the left or to the right of X in the same tail, i.e. independent of X's context. Given a string of terminal symbols, the basic task in using a grammar is to apply re-write rules starting with the string of terminal symbols until a pre-specified top-level symbol, S, is produced. This process is called *parsing*

R_RETRIEVE_KEY	→	R_ROOM_1_TO_ROOM_2 R_OPEN_CHEST R_TAKE_KEY
R_ROOM_1_TO_ROOM_2	→	I_MAKE_DOOR_PASSABLE R_ROOMCHANGE_ROOM_1_TO_ROOM_2
R_ROOMCHANGE_ROOM_1_TO_ROOM_2	→	R_THROUGH_DOOR R_ENTER_ROOM_2
I_MAKE_DOOR_PASSABLE	→	I_PULL_LEVER O_OPEN_DOOR
I_MAKE_DOOR_PASSABLE	→	I_BREAK_DOOR
I_MAKE_DOOR_PASSABLE	→	I_UNLOCK_DOOR I_OPEN_DOOR
R_OPEN_CHEST	→	R_UNLOCK_CHEST R_LIFT_LID
R_OPEN_CHEST	→	R_BREAK_CHEST

Table 1: Sample Plan Recognition Grammar Fragment

and the tree of symbols produced due to rule applications is called a *parse tree*. Note that the combination of a given terminal string and a given grammar can produce many parse trees (a *forest*) due to ambiguity. There are a number of efficient parsing algorithms, which work either as described by starting with S and expanding it (*top-down*), or by starting with the given terminal symbols and applying rules by replacing the tail with the head until the top level symbol is produced (*bottom-up*), or a combination of top-down prediction and bottom-up parsing (Collins, 2003). By making the same context-free assumption in a probabilistic context, namely that rules are expanded independently from each other during the parsing process, a CFG parser can be turned into a Probabilistic Context Free Grammar (PCFG) parser by adding a probability p of rule expansion to each rule. In the context of the paper, the important gain from adding probabilities to rules consists of being able to judge the likelihood of different possible continuations of a sequence of symbols.

4.1.2 Parsing for plan recognition

We employ context free parsing both to perform plan recognition by using events from the game as an observation sequence, as well as to analyse the words in players' utterances. We focus first on the plan recognition in our examples, and later discuss linguistic parsing. The whole point of context free parsing is to recover hierarchical

structures from a sequence of non-hierarchical observations, so it is natural that context free grammars, and especially PCFGs have been suggested as ideal paradigms for performing plan recognition (Bobick and Ivanov, 1998; Pynadath and Wellman, 2000), a suggestion that originally dates back at least to 1960 (Miller et al., 1960). In this case, the symbols in the terminal string correspond to observed events in a temporal sequence, and the grammar specifies possible higher level event structures. Let us turn to a simplified example from the studies that will be described in the next section. The example involves two players, Roirry (prefix 'R') and Isania (prefix 'I'), that engage in the short sequence of events depicted in Fig. 1. Isania pulls a lever to open a door, and Roirry goes through the door and fetches a key from a chest in the next room. Table 1 shows a small grammar fragment covering this example event trace. Given the observation sequence given in Fig. 1, a context free grammar parser would recover the parse tree shown in Fig. 2.

4.1.3 Probabilistic earley parsing

There exist many different choices for parsers, some employing rather distinct parsing strategies. As we will be using the internal data structures maintained by a parser to encode possible affordances at a certain point in time, we prefer parsers that predict only those continuations of the sequence being parsed that are consistent with the higher levels of affordances already predicted, as well as with the lowest level observations

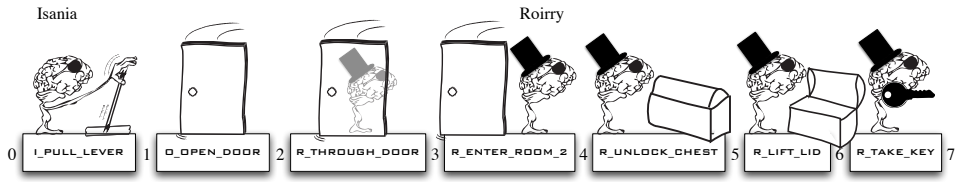

Figure 1: Sample Event Trace

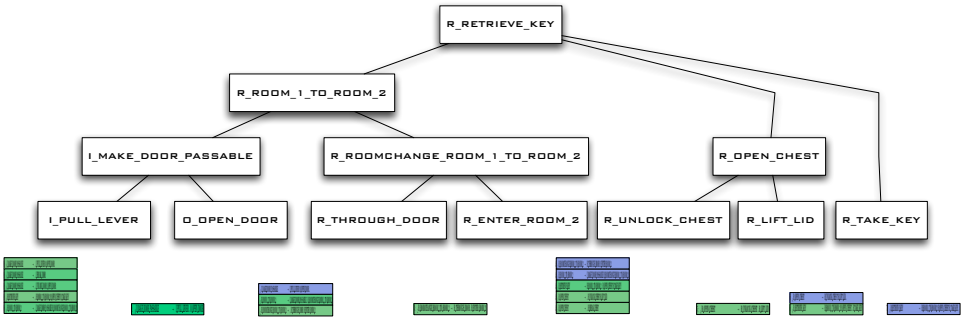

Figure 2: Sample Plan Parse Tree

Position 0:			
0 :	$_0$ I_MAKE_DOOR_PASSABLE	→	. I_PULL_LEVER O_OPEN_DOOR
0 :	$_0$ I_MAKE_DOOR_PASSABLE	→	. I_BREAK_DOOR
0 :	$_0$ I_MAKE_DOOR_PASSABLE	→	. I_UNLOCK_DOOR I_OPEN_DOOR
0 :	$_0$ R_RETRIEVE_KEY	→	. R_ROOM_1_TO_ROOM_2 R_OPEN_CHEST R_TAKE_KEY
0 :	$_0$ R_ROOM_1_TO_ROOM_2	→	. I_MAKE_DOOR_PASSABLE R_ROOMCHANGE_ROOM_1_TO_ROOM_2
Position 1, **I_PULL_LEVER**:			
1 :	$_0$ I_MAKE_DOOR_PASSABLE	→	I_PULL_LEVER . O_OPEN_DOOR
Position 2, **O_OPEN_DOOR**:			
2 :	$_0$ I_MAKE_DOOR_PASSABLE	→	I_PULL_LEVER O_OPEN_DOOR .
2 :	$_0$ R_ROOM_1_TO_ROOM_2	→	I_MAKE_DOOR_PASSABLE . R_ROOMCHANGE_ROOM_1_TO_ROOM_2
2 :	$_2$ R_ROOMCHANGE_ROOM_1_TO_ROOM_2	→	. R_THROUGH_DOOR R_ENTER_ROOM_2
Position 3, **R_ROOMCHANGE_ROOM_1_TO_ROOM_2**:			
3 :	$_2$ R_ROOMCHANGE_ROOM_1_TO_ROOM_2	→	R_THROUGH_DOOR . R_ENTER_ROOM_2
Position 4, **R_ENTER_ROOM_2**:			
4 :	$_2$ R_ROOMCHANGE_ROOM_1_TO_ROOM_2	→	R_THROUGH_DOOR R_ENTER_ROOM_2 .
4 :	$_0$ R_ROOM_1_TO_ROOM_2	→	I_MAKE_DOOR_PASSABLE R_ROOMCHANGE_ROOM_1_TO_ROOM_2 .
4 :	$_0$ R_RETRIEVE_KEY	→	R_ROOM_1_TO_ROOM_2 . R_OPEN_CHEST R_TAKE_KEY
4 :	$_4$ R_OPEN_CHEST	→	. R_UNLOCK_CHEST R_LIFT_LID
4 :	$_4$ R_OPEN_CHEST	→	. R_BREAK_CHEST
Position 5, **R_UNLOCK_CHEST**:			
5 :	$_4$ R_OPEN_CHEST	→	R_UNLOCK_CHEST . R_LIFT_LID
Position 6, **R_LIFT_LID**:			
6 :	$_4$ R_OPEN_CHEST	→	R_UNLOCK_CHEST R_LIFT_LID .
6 :	$_0$ R_RETRIEVE_KEY	→	R_ROOM_1_TO_ROOM_2 R_OPEN_CHEST . R_TAKE_KEY
Position 7, **R_TAKE_KEY**:			
7 :	$_0$ R_RETRIEVE_KEY	→	R_ROOM_1_TO_ROOM_2 R_OPEN_CHEST R_TAKE_KEY .

Table 2: Earley State Sets for the Plan Parsing Example

encountered so far. The ideal candidate for an efficient parser along these lines is an Earley parser, which performs a combination of top-down prediction and bottom-up completion of parse trees (Earley, 1970).

An Earley parser is based on the notion of an Earley state, a structure that concisely summarizes the state of the parser at a particular point in the observation sequence (a sequence of game events in the plan recognition case), and at one hierarchy level of the current parse. An Earley state consists of one rule from the parser's grammar, annotated with three extra pieces of information to encode how and where this rules applies during the parse. Symbols in the sequence are numbered from 0 onwards, where index 0 corresponds to no symbol having occurred yet, index 1 corresponds to the first symbol having occurred, and similarly for the rest of the sequence. For each state, the parser stores: 1) at which index in the sequence this state was created, 2) where in the sequence this rule application started, and 3) how many symbols in the tail of the rule have already been used in the parse so far.

We now step through a plan parsing example in some detail to convey the meaning of Earley states and the workings of an Earley parser. Table 2 shows the state sets an Earley parser would produce while producing the parse tree in Fig. 2. At position 0 in the observation sequence (as indicated by the number before the colon in the states), before any symbols have been observed, the parser predicts states, starting with the top level rule that has $R\_RETRIEVE\_KEY$ as a head. All of these states have a dot in the beginning position of the tail, because no symbols have actually been parsed yet (the dot always signifies which part of the rule has been used so far), and all start at position 0, as indicated by the subscript preceding the rule. Each state can be seen as *predicting* the symbol to the right of the dot - the symbol's occurrence would be consistent with the grammar and the symbols encountered so far. At position one, one of the symbols predicted in the state set at index 0, namely I_PULL_LEVER has been observed, and thus the state that predicted it is copied into the state set at position 1 and

its dot advanced by one step. In state set 2 not only the symbol O_OPEN_DOOR is used, but also the higher level symbol $I\_MAKE\_DOOR\_PASSABLE$ which is the head of a completed lower level rule. Thus in state set 2 we find a state that starts at position 0, meaning that the parser has successfully parsed the first 2 symbols. Parsing continues in this manner until in position 7 the state with the starting symbol $R\_RETRIEVE\_KEY$ is completed, indicating a successful parse of the whole symbol string. The state sets are also visually represented as colour coded stacks below the leafs of the parse tree in Fig. 2. Each state that has the dot to the right of the rule, meaning that it has successfully completed the rule, is coloured in blue, whereas states that still have predictions pending are coloured in green. The same colour scheme will be used to visualize more complex plan parses in the next section. In short, at any given position i in the parse, the Earley parser is predicting a set of next symbols, namely the symbols to the right of a dot in the set of states at i (from those states coloured in green). In a probabilistic Earley parser (Stolcke, 1995), the states that are created during a parse are ranked by probabilities indicating how likely they are to occur. In turn, this lets us rank the symbol predictions made by these states. At each step along the sequence the Earley parser thus generates a set of hierarchically ordered states based on its grammar and the symbols encountered so far. These states predict future symbols, where predictions are ranked by their probabilities. However, the parser does not produce all top-down parse trees, because it uses already present states to predict future states. Thus, a non-terminal will only be expanded at a given position if it occurs to the right of a dot, and each possible symbol will be only expanded once at a given position because the Earley parser re-uses produced sub-trees. The parser thus does not generate all possible predictions, but only those consistent with the grammar and the symbols parsed so far.

4.2 Earley states as perceived affordances

An Earley state used for plan recognition is an ideal candidate for a computational manifestation of a perceived affordance. Assuming that the parser is used to recognize the plans of a particular agent, it

- predicts possible future interactions with the world at a particular point in time (the symbols to the right of the dot in the state);
- ranks the likelihood of possible future interactions given the interaction seen so far through its probability;
- applies to a particular level of abstraction, but is related to other levels due to the hierarchical nature of the grammar;
- restricts the predicted interactions to those consistent with the past and with the parser's grammar.

As an Earley parser progresses, it maintains complete state sets for each point in time, thus providing a complete history of past actions and predictions in addition to currently relevant predictions. We call the grammar used by this Earley parser an *affordance grammar*. This grammar is a predictive model of the structure of the world, representing one agent's predictions about and possible interactions with the world.

4.2.1 Concise environment descriptions

While the representation for affordances presented in the preceding sections is amenable to learning, in the current implementation they have been manually designed. The many rules for the affordance grammar used to derive Earley states are specified concisely via a rule generation system. The rule generation system produces a full set of rules capturing the hierarchical structure in possible event sequences, so that events and sub-events can be recognized and predicted at varying levels of description. The

generation system works from a set of meta-rules that concisely specify 1) the essential events of interest and the sequence in which they must be observed to form higher level events, 2) the hierarchical relationships between these events, 3) the times and types of possible extraneous event structures within other events (note that what is extraneous to recognizing one event sequence may be the core of another), 4) the physical structure of the space (e.g. room connectivity) and 5) the parameterization of event structure (e.g. which actors can be involved in which events). These aspects of the plan recognition problem are interrelated; for example, the physical space structure determines possible temporal event structures. However, specifying these constraints in relative isolation in a meta-language lets the designer work in terms of intuitive constraints on the events being modelled, and leaves the generation of the large space of detailed grammar rules from this specification to the machine.

4.3 Language grounding via affordance filtering

So far, we have discussed parsing as a method for plan recognition. When an utterance occurs during a game session, another parse occurs, namely a linguistic parse using an English grammar. Note that during this linguistic parse the plan recognizer is stopped - it has processed the game events leading up to this utterance, and its current set of Earley states is that used for understanding the utterance during linguistic parsing. This linguistic parsing step uses the same type of Earley parser as described earlier, this time parsing a string of words. Whenever this parser produces a complete state, that is, whenever it successfully applies a full grammatical rule and thus completes a linguistic constituent, it attempts to ground this constituent in terms of ABCs by connecting words to the Earley states (perceived affordances) currently present in the plan recognizer. We us a method of incremental composition driven by language syntax, akin to other work that associates grammatical rules with lambda calculus expressions (Schuler, 2003) and our own work that performs compositional grounding according

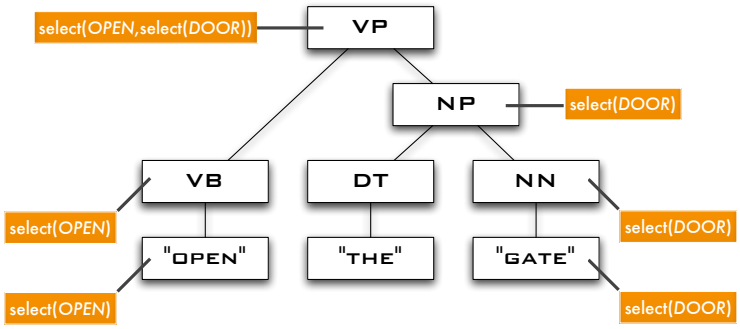

Figure 3: Simple parse tree example and affordance filters

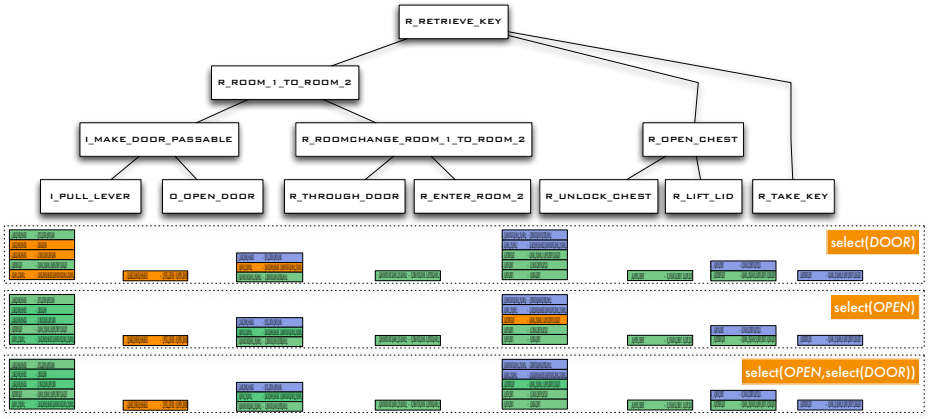

Figure 4: Filter functions applied to affordance example

to explicit composition rules in the grammar (Gorniak and Roy, 2004). Here, we augment the linguistic parser's lexicon with affordance filters that select a subset of the affordances predicted by the plan recognizer.

While parsing the utterance, a noun like "gate" might select all plan recognition Earley states involving opening, unlocking, breaking and walking through at all present and past points in time, whereas a verb like "open" might filter these to only include the possible and actual interactions of opening doors. At higher levels of the utterance parse tree hierarchy, the selections from lower level words and grammatical con-

stituents are combined to produce more complex selection criteria. This simple example is shown in Fig. 3. Fig. 4, on the other hand, shows its application to the previous affordance example. In sequence, the selected affordances for $select(DOOR)$, $select(OPEN)$ and $select(OPEN, select(DOOR))$ are highlighted in orange. This example is highly simplified: even in the restricted scenario presented in the next section, there can be tens of thousands of affordances to be considered, and hundreds of constituents completed during a single parse. We give examples of more complex selection criteria necessary for words such as "that" when describing the studies performed with this implementation in the next section.

In principle, the affordance grammar and thus the plan recognizer should include all possible interactions including verbal ones. Giving a command or asking a question is certainly an interaction with the world. For example, if a player commands another to "pull the right lever" this should not only predict the listener's next action, but also influence the estimate of the speaker's and listener's joint plan. In the affordance grammar and the studies presented in the next section, however, we face somewhat of a chicken-and-egg problem: using the affordance grammar for plan recognition provides a substrate for language understanding, but we need to understand language to write an affordance grammar that can include verbal actions. Once the initial analysis using an affordance grammar without utterances is complete, however, it should be possible to extend this grammar with possible utterance actions and treat utterances identically to other interactions with the world. The work presented here does not include this last step, and thus treats utterances as events external to the affordance grammar. This in turn means that while the meaning of utterances can be resolved in terms of how they express interaction with the physical world, the meaning cannot include linguistic interactions such as commands or descriptions. These are therefore handled externally to the affordance parsing process in the current implementation. It should also be noted that other work exists that deals with the effect of past utterances on the understand-

ing of future utterance (Litman and Allen, 1984), in fact, past utterance are often the only type of situation taken into account by other language understanding systems. We therefore intentionally focus the work here on taking into account the extra-linguistic situation first and foremost, rather than the linguistic one.

5 ABC studies

We have evaluated our implementation of the ABC theory by employing it to interpret situated language recorded from human-human communication during co-operative game play. Specifically, we here provide the results of a study that uses our implementation to understand commands players give to each other, and quantitatively compare our system's predictions with the subsequent actions actually performed by the player being commanded. To perform this study, it is not only necessary to record and analyse human language, but also to apply the machinery introduced in the last section to model the situation in which the language occurs. Studying real world human to human communication presents difficult sensing and action problems unless one severely limits the extent and detail of the physical space, the type of social relationships possible, and the ways in which participants can affect the world. Here, we turn to multi-user graphical online role playing games to provide a rich and easily sensed world to support and capture human interaction.

In short, our study runs as follows. Two players play a computer game in which they control characters in a world of rooms, doors and levers. They explore this world in order to make their way to a final destination. On their way they encounter a series of simple puzzles - levers that open various doors, locked doors that need keys, etc. The puzzles are designed such that the players have to use language co-operate in solving them. For example, one player might have to ask another to pull a certain lever.

Our implementation performs plan recognition on the player's actions as the game develops. Whenever one of the players issues a linguistic command to the other player,

the plan recognizer has taken into account the players' actions up to this point in time, and has produced a set of predictions. At this point, the linguistic parser analyzes the player's utterance to produce an affordance filter that it then runs on the current affordance predictions from the plan recognizer. We now compare the most likely prediction that passes the linguistic filter to the action actually taken by the human listener. The combination of plan recognizer and linguistic parser thus models the human listener's understanding according to the ABC theory.

5.1 Choice of research platform

Current day multi-user graphical role playing games provide a rich interaction environment that includes rooms and exterior areas, everyday objects like chairs, doors and chests, possessions, character traits and other players' avatars. All of these can be acted upon by a player, be it through taking direct action on the world or through speaking with other players. Here, we describe a set of studies using a commercial game, Neverwinter Nights[1], that includes an editor allowing the creation of custom game worlds. A sample in-game view from the player's perspective in this game is shown in Fig. 5. As pointed out before, we acknowledge that using games abstracts away from many of the perception and action problems faced when considering affordances in the real world. However, the generative, hierarchical interaction structures we apply at a higher level here should apply to real world affordances as well, and using games lets us address more socially and spatially complex situations and language.

We have instrumented the game's software environment to record complete transcripts of events in the game world, including player locations, actions such as pulling levers or opening doors, as well as all in-game text messaging between players. Fig. 6 shows the map used for the study presented here. Dependencies between objects in the map are indicated with dotted arrows. The two players start at the South end of the

[1] http://nwn.bioware.com

Figure 5: The in-game perspective of a player in Neverwinter Nights.

map. There are two pre-designed in-game characters available for them to play. One of the characters is a rogue, with the ability to pick locks, whereas the other is a monk, who has the ability to destroy doors with her bare fists. However, the rogue can only unlock the doors and chests marked as unlockable on the map, whereas the monk can only break the doors marked as breakable. The levers each open one door for a short period of time, too short for the same character to pull the lever and run through the door him- or herself. Finally, the chests contain a key each, the first unlocking the other chest, the second unlocking the door behind the first chest. The only objective of the puzzle is to reach the goal indicated on the map. When they start the puzzle, players only know that there is a goal they need to step on somewhere in the module.

One possible puzzle solution plays out as follows: The rogue picks the lock on the South-West door. The monk opens the next door for him with the South-East lever, whereupon he picks the lock on the chest, obtains the key in it, and returns to the start with help from the monk. The monk now opens the South-East door for him, and he uses the key to open the chest here and obtains another key. Once more with help from the monk opening doors, he makes his way back to the room with the first chest and uses the key in the door leading from it (which also opens the center door in the East.) Opening doors for each other, the two characters now switch places and then reach the goal by unlocking or breaking their respective doors.

This puzzle is designed for players to separate and communicate their instructions and goals by using language. As an added restriction, one of the players is randomly chosen in the beginning and forced to only use one of the following phrases instead of being able to speak freely: “Yes”, “No”, “I Can’t”, “Done”, “Now”, “What’s going on?”, “OK.” The other player is free to use unrestricted language. By limiting one of the player’s language repertoire, we exclude dialogue phenomena, which are not the focus of this study.

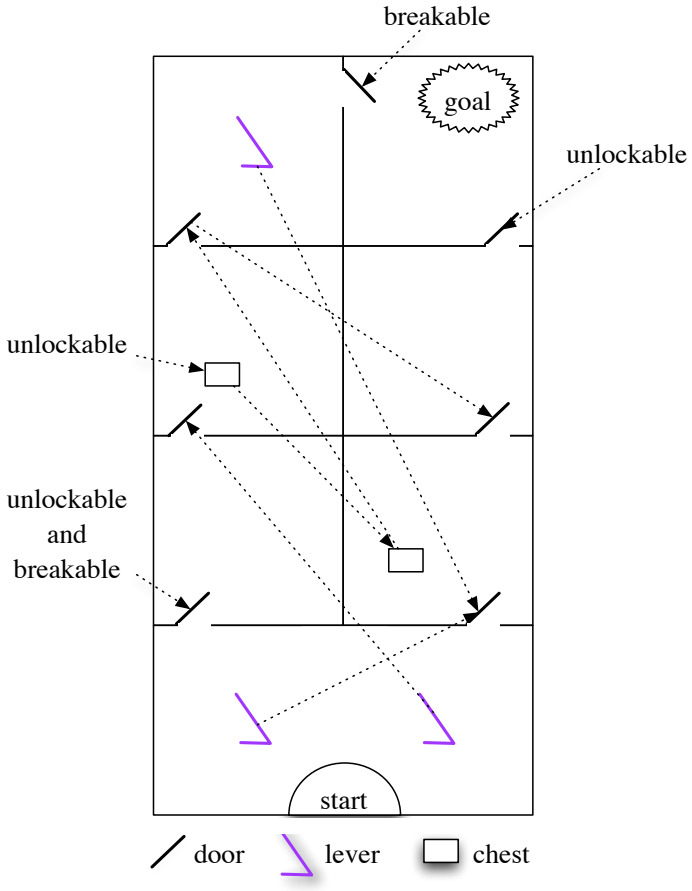

Figure 6: The map of the module used in studies.

5.2 Data collection and annotation

The study included 26 players who played in 13 dyads after responding to ads on the bulletin boards on the Neverwinter Nights website. Eleven of these dyads completed the puzzle in times ranging from 25 minutes to one hour, whereas the others gave up after one hour. Even the two incomplete sessions completed most of the puzzle, except for both players entering the last room. While previous studies showed that the framework handles speech (Gorniak and Roy, 2005a,b), this study only collected typed

text to focus on the semantic problems at hand. Nine sessions served for development purposes, such as writing the affordance grammar and estimating rule probabilities for the linguistic parser, and a group of four sessions formed an unbiased evaluation set. We first annotated the development data, built the system and estimated grammar rule probabilities, then annotated the evaluation data and tested the implementation on this previously unseen data. To generate linguistic parse trees, we first parse utterances with the Stanford Parser (Klein and Manning, 2003) using a standard grammar for written English, and then correct the parse trees by hand.

For plan recognition, the detailed event trace yielded by the game can be abstracted into a simpler trace noting only the relevant changes in world state including

- object interactions (lever pulls, chest use, door interactions)
- room changes
- key acquisitions and exchange
- attempted actions such as attempted unlocks

Table 3 shows a sample event trace segment from one session. In this segment, one of the players (player 'R' for 'Roirry', the player character's name) unlocks the Southwest door (door 4), then attempts to unlock the next door (door 7) and fails. Player 'I' (for 'Isania') now first mistakenly pulls the Southwest lever (opening the Southeast door), but then opens the correct door for Roirry by pulling the Southeast lever (lever 9). Roirry enters the next room, lockpicks the chest in it and acquires the key from the chest. Event traces from the study sessions range between 450 and 2000 events in length.

5.3 Language and situation modeling

The linguistic parser uses a grammar estimated by counting the rules used in the corrected parse trees of the sessions' utterances. The concept specification for the lexical

entries will be further described below.

A set of 90 meta-rules specify the affordance grammar, which captures

- the physical makeup of the puzzle, including room and door connectivity, effects of levers, locations of chests
- the possible actions in every room, including moving to other rooms, pulling levers, unlocking doors, etc.
- planning patterns for players, such as opening a door for the other player to enter a room
- the current state of the world, including which rooms the players are currently in and how much of the puzzle they have solved
- the distinction between actions that further the state of the puzzle solution and actions that do not, such as opening doors without walking through them

The 90 rules expand to a full affordance grammar of about 6500 rules with 1300 non-terminal and terminal symbols. In essence, the meta rules parameterize entities like actors and rooms, whereas the full rule set produces a unique rule for each parameter setting. The lack of parameterization in the actual plan recognition mechanism is one of the shortcomings of using a pure context free grammar parser. However, the parser is efficient enough to run on the large rule set produced by the precomputed parameter expansion employed here. As already pointed out previously, it is desirable to move to a plan recognizer that employs a more concise description of the situation, but none of the existing paradigms near the efficiency and high quality algorithms that exist for parsing. Note that a minimal puzzle solution consists of less than 50 events, thus most of the hundreds of events players engage in constitute player experimentation which we label noise. To predict player actions, it is essential to capture this exploratory behaviour in the affordance grammar. We do this by allowing rules to contain symbols labeled NOISE, which expand to patterns of player experimentation that

do not contribute to the completion of the rule. For example, if a player pulls a lever three times in a row, the first pull (which opens a door) is important to select rules that capture room changing behaviours, whereas the two redundant pulls would be classified as noise. Note that labeling actions as noise still lets us predict them, which is our main goal. For example, we might predict several lever pulls if this is a common player pattern.

Fig. 7 shows 4 sample rules from the full grammar. Symbols consist of parts separated by underscores. These rules can be read as follows: The initial part of each symbol, if it is **I** or **R** indicates the player performing the action (the character names in the modules are *Isania* the monk and *Roirry* the rogue.) These four rules describe actions assigned to Isania, because their head symbols start with **I**. The heads further tell us that in this action Isania moves from the South-West room (rooms are encoded in Cartesian coordinates, thus this is room 0,0) and moves to the second room on the East side. The last part of the head indicates that while this happens, the other player is in room 0,0. To perform this action, the other player (Roirry) must first open the door leading into room 1,1 (door 6) while being in room 0,0 (this action expands to pulling the South-East lever and the door opening) while Isania must then walk to room 1,0 and then to room 1,1. The last symbol is a roomchange sequence rather than a simple room change because players can step back out of the target room and into it again before the door closes. By having a symbol for any sequence like this, the whole episode can be classified as a single room change event. The other three versions of this rule displayed here add room specific noise rules in all possible positions. These rules are marked as **NM** to indicate that they do not produce motion (room changes). The rule itself appears, amongst other places, in the tail of NOISE2_R_ROOM_0_0_I_ROOM_0_0 → I_2_ROOM_0_0_TO_ROOM_1_1_O_ROOM_0_0 NOISE2_R_ROOM_0_0_I_ROOM_1_1, showing how room noise rules transition between each other via movement rules.

The probabilities for the rules stem from counting the number of rule applications

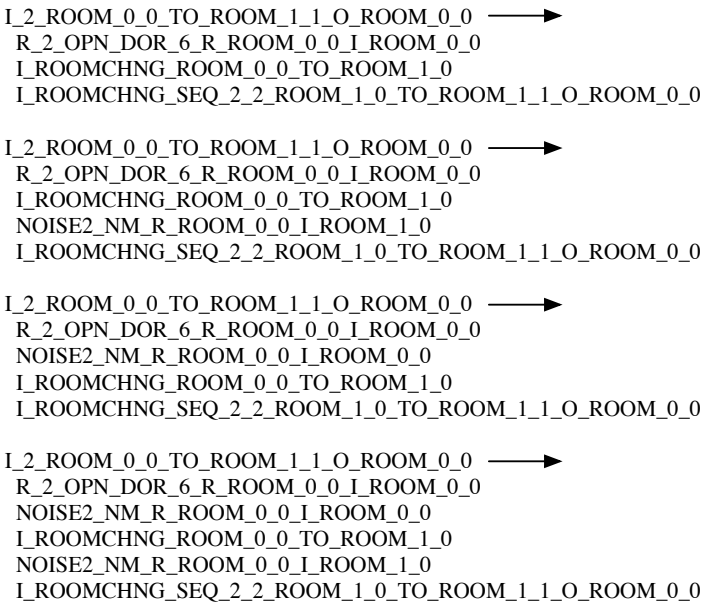

Figure 7: A sample of 4 rules from the expanded affordance grammar.

in the most likely parse trees for the development sessions (the probabilities of rules for the linguistic parser are estimated in the same way). Not all of the rules produced by the meta-rules are actually used in the development sessions (remember that rules are produced for all possible parameter settings), therefore two forms of discounting are needed to produce probability estimates for the remaining rules. Witten-Bell discounting assigns probabilities to rules whose heads have occurred, but whose tails have not, by estimating how likely a new rule with this head is to be seen (Witten and Bell, 1991). This smoothing method uses the number of types of rules with a given head to estimate how likely one is to see another new rule with this head, and divides this probability amongst all the rules with this head that were not seen in the development data. This works for rules whose heads were seen in the training data, but leaves those rules with heads that were not seen. Absolute discounting reserves a fixed probability mass for these rules, and subtracts the mass proportionally from all the rules that were

seen or received a probability via Witten-Bell discounting.

5.4 Communication strategies

Players employ many different types of language acts to communicate with each other about the puzzle, and each type further subdivides into different strategies for expressing intentions. Broadly, these strategies can be broken down into 3 types of language acts,

directives "pull the east lever", "open", "go into the room with the chest and the locked door"

descriptions "there's a lever here", "my switch opens your door", "none of these doors can be lockpicked", "I'm in the entry room"

questions "you're not trapped in the west room are you?", "does it open?", "where have you been?"

Players also produce utterances that have little to do with the actual puzzle solution, such as "it's cold and dark in here", "mutter" or "KILL THE PROGRAMMER!" The current evaluation focuses on directives because their effect on the second human player is relatively easy to measure. Furthermore, as pointed out in the last section, it is a limitation of the current implementation that the affordance grammar does not include possible interactions via language, because it is used to interpret these interactions in the first place. To distinguish between language acts within the framework presented here it is necessary to add them as possible interactions into the affordance grammar itself, so that the system can reason about them. By dealing mainly with directives we avoid this problem for now and interpret the produced grounding for an utterance as a directive by selecting those affordances selected that pertain to the listener (i.e. those the listener could take advantage of at the point in time the utterance occurs) and

considering them as likely actions. We do, however, sketch possible ways to interpret descriptions and questions below, after presenting the results on directives.

Players typed a total of 1742 utterances in the development sessions, and 689 utterances in the test sessions. We annotated 1320 of the development session utterances as being on-topic, that is, relevant to solving the puzzle. Of these, 302 can be considered directives, whereas the remaining utterances are evenly split between questions and descriptions - a distribution to be expected in a puzzle designed to separate players while solving a puzzle. Similarly, the test sessions contain 69 directives out of 427 utterances.

5.5 Affordance filters

As described in Section 4.3, the final result of linguistic interpretation is an affordance filter specification in the form of a nested function call. The affordance filtering process has two stages. First, the final concept specification is interpreted as a filtering function on the current set of affordances, producing another set of affordances that is the interpretation of the utterance at hand in terms of possible physical actions and their abstractions. Second, the utterance is interpreted as a language act, which involves deciding on the type of utterance and taking any measures to treat it as such, which may involve planning to get the character into a situation in which he or she can perform the action predicted.

5.5.1 Filter functions

In addition to the affordance set arguments they take as described in Section 4.3, filters are further parameterized with static parameters specified in the lexicon to re-use the same filter for different words (for example “east” uses the same filter function as “west” with different parameters). Many words have multiple meanings, of course, even in the limited world of these studies. Some examples of several meanings (for

example for “that”) occur below, but not all meanings are covered by the system. We discuss failures due to missing meanings in Section 5.6.1.

Simple Selection The simplest filtering function, *select*, selects affordances by substrings in their predicted next symbols. Thus, a word like “open” selects all affordances involving opening of chests or doors.

Actor Selection The *actor_selection* filter can select either the speaker (“I”), the listening character (“you”), or both characters (“us”,“’s”) by filtering affordances for the initial actor string in their predicted symbols.

Indexicality The *expand_set* filter uses the currently predicted set of affordances for the speaker as a source set, and selects a target set selecting either all affordances that specify the same interaction but for any actor. This is the filter associated with the word “this”, selecting, for example, all the possible interactions with a lever next to the speaker for the fragment “this lever.” For the word “other”, the same filter selects affordances of either actor of the same type (e.g. opening doors or pulling levers) that are not currently available to the speaker (that are, for example, not in the current room).

The *select_distant* filter, on the other hand, collects affordances that were encountered by the speaker at some point in the past and are not available in the speaker’s current state. It grounds, for example, one use of “that” as in “What about that lever?” where the speaker is standing next to one lever, but referring to another one with this utterance.

Movement Planning The *plan_path* filter plans a path from the current set of affordances to another by assuming that location changes are enough to bring about the target set. This is largely a valid assumption in the puzzle discussed here: players can usually interact with the things around them, though some plans produced this way may be invalid because the players have not yet advanced far

enough in the puzzle. For example, they may not have managed to open a door yet that is necessary to enter a target room. Movement planning takes into account the rules of the puzzle, such that players have to open doors for each other to get into certain rooms. This filter is used for words like "go" (as in "can you go stand by the other lever") or "run." The same planning functionality is also used when interpreting an utterance as a directive, which is discussed below.

Discourse Reference For every utterance, the parser stores the affordance set of the last filter call that filters by neither actor or planning. A back reference filter (*back_ref*) simply re-activates this set of affordance for words like "it."

Past Interactions The *select_past* filter finds those perceived affordances that were actually taken advantage of by the agent in the past. This yields another use of the word "that" as in "Let's try that again."

Location Reference The *select_location* filter selects affordance sets by the possible room changes they predict. This is used, for example, to ground "left" and "West" by selecting for those sets of affordances that predict a room change interaction in which the target room has an x value of 1. Note that this means that locations are defined by how one leaves them (i.e. "west" is a location from which one can walk East.) Again, this is obviously not the most general and only meaning of location references, but it works very well in the scenario discussed here.

Possession Players tend to use "my" and "your" to refer to objects they interacted with recently, thus the *select_recent* filter selects the most recently used affordances in the current set.

5.5.2 Interpreting directives

For a directive, the system first applies the concept specification provided by the linguistic parser to produce a set of affordances grounding the utterance. It then translates the resulting set of affordances into a predicted next action by finding the most recent affordances in the set and checking whether any are also available for the listener in the currently predicted set. If they are, they are turned into the basic actions they predict (that is, actions the player can actually take), by walking down the affordance grammar until a lexical item is reached. If they are not currently available, but are known to be available in other situations, the system will plan a path to the room in which such an affordance would be available, and make the first action in this plan its immediate prediction. Note that such a plan not only includes movement steps, but also the steps necessary to gain passage such as pulling levers to open doors for other players. If no predictions are produced in this way, it might be due to the fact that the next action predicted is not the listener's to take, for example in the case where the speaker must open the door for a listener to walk through. Thus, the implementation now proceeds with a depth first search for the next action of the listener starting with the currently predicted symbols in the rules contained in the selected affordance states. If any of these steps produce multiple predictions, they are ranked by the sum of the forward probabilities in the Earley states producing them, and the most probable action is used as the prediction.

Whenever one player gives the other a directive, the utterance is parsed by the language parser to produce an affordance filter specification. The plan recognizer then runs this filter specification on the complete set of affordances produced up to this point in the game, which yields a filtered set of affordances. These are then interpreted as described above to yield a single best prediction. To measure performance, this prediction is compared to the next action the player in question actually takes, and counted as correct if it matches.

5.6 Results

Table 5.6 shows the overall results of language understanding using this method. All results are split between the development and the test set to show generalization to unseen data. The first row (*All Directives (AD)*) shows the performance on the complete set of 302 directives in the development sessions and 69 directives in the testing sessions. However, players do not always follow instructions, so the second row (*Followed Directives (FD)*) shows performance only on the 281 cases where the player actually performs an action that matches the directive as determined by the annotator (64 in the testing session). Half of the directives players used and followed correctly are what we will call *action markers*: single word utterances that do not significantly restrict the nature of the action to be performed, but rather mark the time at which the obvious action should be performed. Such utterances include "now", "go", "lever" and "open." While the high frequency of such action markers supports the claim made here that the interactive situation determines much of the meaning of language (sometime so much that language becomes unnecessary), the performance of the linguistic component of the system is not evaluated in these utterances. *Followed Long Directives (FLD)* in Table 5.6 therefore shows performance on the half of the directives that contain more than one word. The average length of the total set of directives lies at 3.6 words, but rises to 6.2 words when restricted to the set of development directives employing more than one word (4.5 vs. 6.5 in the test set). Performance on this set of linguistically interesting directives is generally lower because the language groundings used in this study do not cover all of the meanings that occur (omissions and problems are discussed further below). However, the gap to the pure plan recognition baseline widens significantly on this utterance set, showing that the system can understand more complex language and produce the correct prediction for many of these directives.

Table 5.6 shows a number of prediction baseline results for the same data sets. The *Hierarchical Plan Recognition* value shows the performance if language is ignored -

that is, if we simply pick the most probable prediction of the plan recognizer at the point an utterance occurs, without paying attention to the words in the utterance. As above, *FD* and *FLD* restrict the pure plan recognition baseline to those directives that were correctly acted upon by the listener (FD), and then further to those that use more than one word (FLD), respectively. *State Based Maximum* counts the actions players took when they were in a specific combination of two rooms, and in response to a directive predicts the action taken most often in this combination. Finally, *State Based Random* randomly picks amongst all the actions players were ever observed to perform in a room combination.

When interpreting these results, it is important to keep in mind that perfect prediction cannot and should not be achieved in any of these cases. The puzzle naturally causes much exploration by the players, and, as will be discussed further below, situations and directives often do not limit players to a single next action. Some amount of variability is thus inherent in the scenario.

The best overall performance of the complete system was 72%. Given the complexity of the problem and the leeway players appear to give each other in following their own utterance, this figure indicates that the theory and implementation presented in previous sections make for an effective substrate for language understanding systems.

It is clear from these results that the hierarchical plan recognizer captures important aspects of the puzzle solution: it shows over 20% improvement in predictions compared to a simple predictor baseline. Prediction is also no simple task, as the low random baseline shows (even this baseline does not pick amongst all possible actions, but only those players performed in the development data). Language understanding heavily relies on plan recognition - often the meaning of an utterance is highly constrained by the player's states and plans. Taking the words into account, however, improves again on the pure plan recognition performance. The best measure of this improvement is the 11% gain (8% in the test set) seen when considering the set of cor-

rectly followed directives longer than one word. The percentage performance gain is smaller when considering all utterances because performance is dominated by action markers, for which linguistic content plays little role, and thus yields no improvement in performance. Not all action markers are acknowledged by the simple rule of considering one word utterances to be action markers: "go for it", "go go go", and other multi-word action markers occur in the data, but they occur rarely.

Performance on the test utterances is entirely comparable to that on the development utterances, showing that the plan recognition grammar and linguistic parser, while restricted in their coverage, generalize well to unseen data. As already discussed, individual sessions differ greatly in playing and communication style. In fact, there is a single session in the test set that contains very repetitive and easily predicted player behaviour. When it is omitted, the test set performance baselines are equal to or lower than the development set baselines.

5.6.1 Detailed performance and mistakes

Examining the utterances in detail yields clues as to the benefits and shortcoming of the implementation presented.

Action Markers We call utterances that impose next to no restrictions on the action to be performed via their words *action markers*. The most common ones (about half the data) are "go", "now", "open." There is an external bias imposed favouring "now" because it was one of the only action markers available to the non-speaking character. For this class of utterances, performance of the utterance understanding algorithm can only be as good as predictions made by the plan recognizer. However, the performance figure here also underestimates the performance of the language understanding system: it seems that in many cases players do not have an exact action in mind. For example "open" might really be taken to mean "open anything and everything you can" or "open something" in

several cases, especially when players cannot see each others' characters. Sometimes players even explicitly indicate this as in "try something else." We will discuss performance of the plan recognizer further below.

Simple Selection Almost every utterance that is not simply an action marker uses at least one content word involving simple selection of affordances (and even an action marker like "lever" or "open" does). The overall performance speaks to the usefulness of the affordance filtering approach in understanding directives in a plan recognition context.

Location Reference These include utterances like "throw the one to the west" and "now head to the east lever." These occur a significant amount in the data (35 utterances in the development data) and are correctly understood if in combination with a simple request. 4 of the 35 are incorrectly understood because they involve constructions or commands not covered by the affordance filters, such as "can you try thief' [sic] picking either the chest or north lock."

Discourse Reference 7 out of 11 uses of "it" (as in "I need you to pull it" in the development data were correctly understood via the *back_ref* filter. The remaining suggest that there are influences on the use of "it" in this context beyond the discourse one.

Indexicality Indexicals including "this", "that" and "other" were understood correctly in half of the cases (14 out of 28). In the 4 (out of 9) misunderstood cases of "this" the mistakes are due to problems with actor attribution, not with indexicality, as they are all of the form "throw it and i'll throw this one" or "let me go down this way once more ... not saying it'll help." "That" is correctly interpreted in 5/7 cases and "other" in 5/12. This only partially indicates problems with their current groundings, as some of the mistakes are due to other words in the utterance such as in "can you try to open from the other side somehow?", which

lacks groundings for “side” and “from” at minimum.

Movement Planning Is not only used for phrases like “go to” and “stand by”, but also to interpret any utterance that produces affordances not available to the listener in his or her current location. As such, it is involved in understanding most utterances and performs extremely well.

Other communication strategies occurred too rarely to allow for meaningful analysis. There are a few overarching problems and omissions with the implementation presented here:

Missing Meanings There are a few classes of meanings that occur in the data for directives that the implementation currently does not handle at all. There are a number of idioms like “go for it” and “come back” that perhaps should be handled as idioms and not analysed word by word. Sometimes complicated linguistic structures occur, often expressing temporal dependencies and causality. These can even be intermixed with descriptions such as in “I need you to pull it when I open the door for you ... I think it opens the door on the other side.” However, constructions this complex are rare.

Spatial Coarseness Spatial locations in the structural grammar are purely room based, and thus relatively coarse. For distance based directives, for example those including “that”, utterances can be misunderstood because the player considers him- or herself distant from an object and uses “that”, but is still considered to be in the same room as the object by the affordance grammar.

Multiple Interpretations The particular implementation discussed here uses the best interpretation of an utterance exclusively. In previous work we have shown ways to consider multiple weighted interpretations simultaneously by probabilistically mixing the linguistic elements from the language parser with the affordances produced by the structural grammar (Gorniak and Roy, 2005a). It would clearly

be beneficial to adapt those methods to the system described here to consider multiple word and constituent meanings and their interpretations simultaneously.

Learning The paradigm presented here lends itself to supporting learning by a synthetic character. Possible learning targets include the weights and rules of the structural grammar, the function bindings for words, and the interpretation of words in terms of affordances. Especially together with a coherent framework for considering multiple interpretations such a learning framework would likely improve robustness of the understanding system over the partially handcrafted approach taken here.

Omniscience vs. Player Modelling The plan recognizer used here models both players simultaneously and is informed of the structure of the puzzle. This eases recognition of interdependent actions by the players (such as pulling a lever to let the other person through a door), and increases prediction accuracy by taking into account the actual puzzle structure. However, when interpreted as perceived affordances, the plan states should correspond to those maintained by an individual player attempting to solve the puzzle, not to an omniscient planner for both players. For many directives this is not a problem, because "pull the east lever" can be understood in either model. Problems arise when players are mistaken about how to solve the puzzle, for example when they assume that levers act differently when pulled simultaneously. This presents two problems, one for directives and one for descriptions, discussed below. An utterance like "let's try that again" might refer to the joint action of the characters pulling their respective levers, which is not modelled in the plan used. In the particular puzzle there are few directives of this sort, but the effect on performance of the plan recognizer, which does not acknowledge these falsely perceived structures, may be degrading performance.

Descriptions The second problem with an omniscient plan recognizer is that it makes

it hard to interpret descriptions. A player utters a description to inform the other player of the physical makeup of the puzzle ("there's a chest and a locked door in this room"), his or her mental model of how the puzzle works ("they both open opposite doors"), or the effects of actions ("both door and chest remain locked"). Intuitively, each should produce a change in the listener's mental model of the situation: he or she might consider new affordances or discard ones previously thought to be available. As only the correct affordances are available in the omniscient plan recognizer, it is impossible to model this effect. However, the filtering mechanisms proposed here lend themselves to exactly this type of effect when run on a different type of plan recognizer – one that is uninformed about the puzzle structure and has limited perception of the other player's actions.

Questions Questions are in content very much like descriptions in the data collected for these studies, because the listener could respond only with primitive utterances. Thus, they usually read like a description in question form, for example "is the door back there locked?", in effect filling in the questioner's model of the puzzle workings and world state via the response.

Plan Recognition Beside the problem of whether to use an omniscient or several player-specific plan recognizers (or both in tandem), there are other problems with the plan recognizer used here. As Pynadath and Wellman point out, while successful in estimating hierarchical plans of agents, grammar based plan recognizers are not naturally parameterized in an intuitive or useful way. For example, many of the thousands of rules used in the plan recognizer here are due to the fact that they are largely conditioned on the rooms the players find themselves in. Rather than being parameters, these rooms are part of the symbols used in the grammar rules, and are explicitly produced by the meta-rules. The meta-rules are in essence a parameterization of the grammar, but they are not used during the actual plan recognition. To more easily derive and estimate affordance gram-

mars, and also to reason directly about the underlying state variables, it seems advisable to go to a combined model of a grammar and an underlying state model that are linked but represented separately (Pynadath and Wellman, 2000).

Many of the limitations we have mentioned are due to our particular choice of plan recognizer. As we have pointed out, alternative models of plan recognition exist that utilize more intensional and less omniscient representations. The ABC theory of casting language understanding as a filtering process on possible affordances transfers directly to these alternative computational approaches and in the future we hope to show that a such revised implementation scales to larger problems, handles questions and descriptions and does not require a complete model of the entire problem a priori.

6 Conclusion

We hope to have convinced the reader at this point of four things, namely

- that language understanding depends on a mental representation designed for interaction with and prediction of the world
- that the notion of an *affordance* captures the crucial element of a theory of concepts that from the ground up acknowledges the need for interaction with the world
- that affordances make for powerful computational instantiations based on planning and plan recognition and lead to a new method for truly grounded computational language understanding
- and that, by example, this new method can feasibly be implemented and performs well in understanding spontaneous human language in a complex situation.

The implementation presented in this article provides a convenient framework for probabilistic hierarchical reasoning about affordances while understanding situated

language. As it stands, this provides one possible interpretation of the theory presented for the case of language situated in the current physical and intentional context. The performance of the implementation when compared to human decisions shows the viability of the theory in leveraging intentions and affordances to understand language of this type. In doing so, it also lends further support to work on affordances as an aspect of human cognition by providing a working synthetic model understanding human language that employs affordances (Glenberg and Kaschak, 2002). It will be important to integrate this framework with other approaches and views on affordances (Steedman, 2002; Roy, 2005) and to re-phrase existing approaches dealing with other aspects of grounded language understanding in an affordance-based framework.

The particular framework of hierarchical, probabilistic plan recognition using context-free grammars is one possible choice, and it works well for the high level events that constitute the input in the computer game worlds studied here. Other choices are certainly possible, the schemas introduced by Roy (2005) being a different one that is more applicable to directly modeling low-level sensory input and motor action. However, each of the possible instantiations of the ABC theory must share important features: they must all dynamically generate the affordances for a situation particular to the agent and its environment. To predict or generate an agent's choices, they must take into account the agent's goals and respect how the agent's goals and abilities interact with the environment and other agents. They are likely to be hierarchical to capture different levels of granularity, though a complete solution is unlikely to employ a uniform encoding of affordances like the Earley states used here. Particularly, while a context-free grammar is generative it is not concisely parameterized, more intensional models are more likely to be successful at levels and in situations where an extensional listing of affordances is simply prohibitive due to their numbers. That said, we see no reason that the model introduced when appropriately parameterized (as is done in many hierarchical planners such as HTN planners) should not scale to cover a larger range from

sensory inputs to high level planning, and should not be able to deal with more complex environments. In fact, due to its hierarchical, abstract nature, our framework may be an ideal candidate for tying together more modality and task specific representations into a coherent affordance modeling framework.

We believe the ABC theory to be a useful new view of mental representation of concepts. It is unique in its computational interpretation of Gibsonian affordances based on plan recognition, and its successful realization in a language understanding task dealing with spontaneous, situated human language. We hope that this pairing of theory and implementation speaks to those studying human mental representation as well as those building artificial language processing systems.

6.1 Future work

To handle questions and descriptions in addition to commands, the implementation needs to be address partial observations and lack of knowledge. As a first step, one might replace the symbol string representing events in the game world with a confusion network Mangu et al. (1999). When players are in the same room, the confusion sets of this network contain a single member because players can see each others' actions. When they are in different rooms, however, each confusion set representing an action by the other player contains all possible actions currently available to that player. Using confusion networks would spread the probability assigned to the current world state over many possible states as players take actions without seeing each other act. This directly leads to the ability to interpret a subset of descriptions and questions such as "I'm in the Northwest room" or "Did you make it into the next room?" The descriptions would have the effect of narrowing the probability distribution over possible world states by raising the probability of the described state. Beyond uncertainty about the current state of the world, future extensions might include an explicit treatment of lack of knowledge about the structure of the world, such that when a player encounters

a lever for the first time, he or she might be modelled as generating predictions about possible effects of this lever that might then be verbally described or experimentally explored. The omniscient plan recognizer employed by the current implementation has access to too much a priori knowledge of the game's structure to model such thought processes directly, or to be deployed in an environment that is not modelled ahead of time. More generally, we hope to apply the ABC theory to other research platforms such as communicating robots, which have other requirements such as sharing the physical world with their human communication partners. They will therefore impose different demands on future interpretations of the ABC theory, but their concepts will also be designed for interaction from the ground up.

7 Acknowledgements

We thank Josie Hughes for letting us use her brain drawing for illustration purposes. This material is based upon work supported by the National Science Foundation under Grant No. 0083032..

References

Allen, J. and Perrault, R. (1980). Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178.

Barsalou, L. (1999). Perceptual symbol systems. *Behavioural and Brain Sciences*, 22(4):577–609.

Bickhard, M. H. (2001). Function, anticipation and representation. In Dubois, D. M., editor, *Computing Anticipatory Systems. CASYS 2000 - Fourth International Conference*, pages 459–469, Melville, NY. American Institute of Physics.

Bobick, A. F. and Ivanov, Y. A. (1998). Action recognition using probabilistic parsing.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Boutilier, C., Dean, T., and Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of AI Research*, 11:1–94.

Bui, H. H., Venkatesh, S., and West, G. (2002). Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research*, 17:451–499.

Chapman, D. (1991). *Vision, Instruction and Action*. MIT Press, Cambridge, MA.

Clark, A. (1998). *Being There: Putting Brain, Body and World Together Again*. MIT Press.

Collins, M. (2003). Head-driven statistical models for natural language parsing. *Computational Linguistics*.

Drescher, G. (1991). *Made-up minds*. MIT Press, Cambridge, MA.

Duranti, A. and Goodwin, C. (1992). *Rethinking Context: Language as an Interactive Phenomenon*. Cambridge University Press.

Earley, J. (1970). An efficient context-free parsing algorithm. *Communications of the ACM*, 6(8):451–455.

Erol, K., Hendler, J., and Nau, D. (1994). Htn planning: Complexity and expressivity. In *Proceedings of the American Association for Artificial Intelligence*.

Geib, C. and Goldman, R. (2005). Partial observability and probabilistic plan/goal recognition. In *IJCAI-05 workshop on Modeling Others from Observations*.

Gibson, J. (1977). The theory of affordances. In Shaw, R. and Bransford, J., editors, *Perceiving, Acting and Knowing*, pages 67–82. Wiley, New York.

Glenberg, A. M. (1997). What memory is for. *Behavioural and Brain Sciences*, 20:1–55.

Glenberg, A. M. and Kaschak, M. P. (2002). Grounding language in action. *Psychonomic Bulletin and Review*, 9(3):558–565.

Gorniak, P. and Roy, D. (2005a). Probabilistic grounding of situated speech using plan recognition and reference resolution. In *Proceedings of the International Conference on Multimodal Interfaces*.

Gorniak, P. and Roy, D. (2005b). Speaking with your sidekick: Understanding situated speech in computer role playing games. In *Proceedings of Artificial Intelligence and Digital Entertainment*.

Gorniak, P. J. and Roy, D. (2004). Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470.

Hsiao, K., Mavridis, N., and Roy, D. (2003). Coupling perception and simulation: Steps towards conversational robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of teh 41st Meeting of the Association of Computational Linguistics*.

Laurence, S. and Margolis, E. (1999). Concepts and cognitive science. In Margolis, E. and Laurence, S., editors, *Concepts: Core Readings*, chapter 1, pages 3–81. MIT Press.

Litman, D. J. and Allen, J. F. (1984). A plan recognition model for clarification subdialogues. In *COLING*, pages 302–311.

Littman, M., Sutton, R., and Singh, S. (2001). Predictive representations of state. In *NIPS*.

Mangu, L., Brill, E., and Stolcke, A. (1999). Finding consensus among words: Lattice-based word error minimization. In *Proceedings of EUROSPEECH'99*, volume 1, pages 495–498, Budapest.

Miller, G. A., Galanter, E., and Pribram, K. H. (1960). *Plans and the Structure of Behavior*. Adams, Bannister, Cox, New York.

Minsky, M. (1985). *Society of Mind*. Simon and Schuster, New York.

Nau, D., Au, T., Ilghami, O., Kuter, U., Murdock, W., and Wu, D. (2003). Shop2: An HTN planning system. *Journal of Artificial Intelligence Research*.

Prinz, J. (2002). *Furnishing the Mind: Concepts and their Perceptual Basis*. MIT Press, Cambridge, MA, USA.

Pulvermüller, F., Härle, M., and Hummel, F. (2001). Walking or talking?: Behavioral and neurophysiological correlates of action verb processing. *Brain and Language*, 78:143–168.

Putnam, H. (1975). The meaning of 'meaning'. In *Philosophical Papers, Vol. 2: Mind, Language and Reality*. Cambridge University Press.

Pynadath, D. V. and Wellman, M. P. (2000). Probabilistic state-dependent grammars for plan recognition. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence, UAI2000*. Morgan Kaufmann Publishers.

Roy, D. (2005). Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*.

Roy, D., Hsiao, K.-Y., and Mavridis, N. (2004). Mental imagery for a conversational robot. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(3):1374–1383.

Schank, R. C. and Abelson, R. P. (1977). *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates.

Schuler, W. (2003). Using model-theoretic semantic interpretation to guide statistical parsing and word recognition in a spoken language interface. In *Proceedings of the Association for Computational Linguistics*.

Smith, B. C. (1996). *On the Origin of Objects*. MIT Press, Cambridge, MA, USA.

Steedman, M. (2002). Formalizing affordance. In *roceedings of the 24th Annual Meeting of the Cognitive Science Society*, pages 834–839.

Stolcke, A. (1995). An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201.

Stone, M. (2001). Representing communicative intentions in collaborative conversational agents. In *AAAI Fall Symposium on Intent Inference for Collaborative Tasks*.

Stoytchev, A. (2005). Behavior-grounded representation of tool affordances. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, page ??

Winograd, T. (1970). *Procedures as a representation for data in a computer program for understanding natural language*. PhD thesis, Massachusetts Institute of Technology.

Witten, I. H. and Bell, T. C. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. Information Theory*, 37(4):1085–1094.

Zwaan, R. A. (2003). The immersed experiencer: Toward an embodied theory of language comprehension. *The Psychology of Learning and Motivation*, 44.

High Level Events
R_ATTEMPT_UNLOCK_DOOR_4
R_UNLOCK_DOOR_4
R_ATTEMPT_UNLOCK_DOOR_4
R_OPENDOOR_DOOR_4
I_THROUGH_DOOR_4
I_ROOMCHANGE_ROOM_0_0_TO_ROOM_0_1
R_THROUGH_DOOR_4
R_ROOMCHANGE_ROOM_0_0_TO_ROOM_0_1
R_ATTEMPT_UNLOCK_DOOR_7
I_THROUGH_DOOR_4
I_ROOMCHANGE_ROOM_0_1_TO_ROOM_0_0
I_ACTIVATE_LEVER_10
O_OPENDOOR_DOOR_6
I_ROOMCHANGE_ROOM_0_0_TO_ROOM_1_0
O_CLOSEDOOR_DOOR_6
O_DEACTIVATE_LEVER_10
I_ACTIVATE_LEVER_9
O_OPENDOOR_DOOR_7
R_THROUGH_DOOR_7
R_ROOMCHANGE_ROOM_0_1_TO_ROOM_0_2
I_ROOMCHANGE_ROOM_1_0_TO_ROOM_0_0
O_CLOSEDOOR_DOOR_7
O_DEACTIVATE_LEVER_9
R_ATTEMPT_UNLOCK_CHEST_13
I_THROUGH_DOOR_4
I_ROOMCHANGE_ROOM_0_0_TO_ROOM_0_1
R_UNLOCK_CHEST_13
R_OPENPLACEABLE_CHEST_13
O_INVENTORY_CHEST_KEY_14

Table 3: A Sample Event Trace Segment from a Study Session

Selected Utterances	**Accuracy - Development**	**Accuracy - Test**
All Directives (AD)	70%	68%
Followed Directives (FD)	72%	70%
Followed Long Directives (FLD)	61%	68%

Table 4: Results of Understanding Directives in the Neverwinter Nights Puzzle Scenario

Prediction Type	**Accuracy - Development**	**Accuracy - Test**
Hierarchical Plan Recognition (AD)	65%	63%
Hierarchical Plan Recognition (FD)	66%	64%
Hierarchical Plan Recognition (FLD)	50%	60%
State Based Maximum (AD)	42%	48%
State Based Random (AD)	15%	17%

Table 5: Prediction Baselines for the Neverwinter Nights Puzzle Scenario