Subjective Ontologies*

Devika Subramanian†
Computer Science Department
Cornell University
Ithaca, New York 14850
devika@cs.cornell.edu

John Woodfill‡
Computer Science Department
Stanford University
Stanford, California 94305
woodfill@neon.stanford.edu

Oct 1990

Abstract

This paper extends our previous analysis of the use and computational benefits of indexicals [SubWood89] to the problem of scaling reactive systems to realistic domains. We first note that perception is primarily indexical. By taking advantage of this indexicality, we drastically reduce the amount of work done by both the reasoning and the perceptual components of a reactive system. Indexical ontologies which pick out particular properties or objects in the world to direct action towards, allow the construction of concise propositional prescriptions for action. We provide a method for producing indexical theories from classical ones. First we generate a classical plan which makes the assumption that a complete world model is available and then compile it into a reactive plan which has finer grained actions and limited sensing. We demonstrate this in the context of deriving a reactive monkeys-and-bananas planner from a situation calculus axiomatization of this world.

1 Introduction

In our Kr89 paper, we showed that indexical-functional representations can be understood in terms of the well-known situation calculus formalism. We showed how a theory of action formulated in the situation calculus could be converted into an indexical one that directly related perceptual inputs to effectory outputs of an agent. This transformation required us to severely restrict our model of time by introducing the indexical *now* to refer to the current situation at any time, and by allowing guarded access to past situations via the *Before* function. In particular, only a finite subset of the infinity of situations expressible in the standard situation calculus were now allowed. These situation terms could be propositionalized: for instance, we could work only with three situations *now*, *before-now*, and *before-before-now*. This limited the ability to refer to past situations: in this case, projection can only go 1 time step ahead and two time steps back. Restricting the ability to refer to situations other than *now* and a few situations *before-now* allowed us to completely eliminate quantified reasoning in our theory of action with obvious computational benefits. The introduction of the time indexical *now*, paved the way of the introduction of other indexicals that non-rigidly referred to objects, e.g. the-block-in-front-of-me.

*This is a draft of a paper currently being written. Please do not cite or distribute without permission from the authors.
†This author is supported by NSF IRI-8902721.
‡This author was supported in part by ONR Contract N00014-81-K-0004.

This analysis therefore, exposed some of the expressiveness and efficiency trade-offs resulting from using indexicals. The main power of these subjective ontologies came from the fact that they bunched objects into equivalence classes based on functional roles; the loss of expressive power in the inability to identify uniquely members of a given class was more than adequately compensated by a corresponding gain in computational complexity because of the reduction of the ontology.

From a logical standpoint, our analysis of the indexical *now* involved extending the situation calculus to support the use of constant symbols that denote different individuals in different contexts. In this sense, our axioms for the introduction of *Before* and *Now* correspond to developing a tense logic. Very sophisticated model-theoretic accounts of indexicals have appeared both in the AI literature [Lesperance] as well as the philosophy literature [DKaplan]. Our objective in the present analysis of indexicals is neither to provide a truth-theoretic semantics for English sentences that contain indexicals nor to provide an account of one agent reasoning about another, rather it is to explore the question of how systems can embody indexicality for computational benefit or for augmentation of capabilities. These questions have to do with the indexical coupling of the world and the perceptual and motor machinery of the robot.

Our paper is organized as follows. In Section 2, we discuss the problem of linking perception to action. The indexicality of perception is the subject of Section 3 where we also elaborate on mechanisms for establishing and maintaining reference of indexical terms. In Section 4, we sketch the derivation of an indexical theory from a situation calculus one using information about goals as well as descriptions of the sensory and effectory capabilities of a robot. In Section 5, we debunk a few misconceptions about the computational benefits of indexicality and conclude with the main points of this paper.

2 The Problem: Linking Perception and Action

Consider constructing a robot to follow an object in front of it, while avoiding running into anything. A classical picture of how such a system ought to be built is as follows: the robot has a vision system which produces a labeled scene description – asserting what the objects in view are, and where they are. The ideal in this classical picture would have this description formulated in some absolute coordinate space. The control component of the robot would be a planner which processed the scene description in order to find the object to be followed, and the obstacles to avoid. The sequence of actions produced by the planner would then be passed on to an execution monitoring system that drives the robot to act in the world.

The problems with the classical picture are well known. First, it assumes that the world had been abstracted into an objective description consisting of a set of well-formed formulas. The effects of actions in the world are then formulated as sentences in the situation calculus and the planning problem is: what sequence of actions achieves the goal description? Chapman[Chapman] showed that the problem as formulated above is NP hard even in the propositional case. This standpoint on planning leaves open the questions of how exactly the referents of symbols in these theories map to the world (the reference problem), and introduces the additional problem of monitoring in which the planner makes sure that the world is indeed in the state specified by the descriptions.

Horswill's[Horswill] robot is a reactive planner that solves the problem posed at the start of this section. His robot keeps the object it is currently interested in centered in its visual field. The robot gets an image, once every .2 seconds, segments it to find an interesting object, tracks the object from frame to frame by latching the frame at the previous time tick and uses it to compute a relative direction of motion of the object. Key to his design is the short path between sensors and effectors, which allows his system to be reactive and avoid obstacles in real time. Thus reactive planners make a more careful analysis of the requirements of planning. Situation calculus allows one to make too many distinctions in the world. Planners that need to work in changing environments cannot be as wasteful – they need to simplify their models of the world and of their decision-making. In our earlier paper [SubWood89] we demonstrated that reactive planners access the world via terms in a agent-centered ontology (e.g. the object-I-am-chasing) as opposed to terms in an objective ontology (block A). This allowed us to rewrite theories of the world in propositional form. Designers of reactive planners develop elaborate perceptual components for acting on the referents of these subjective or indexical terms: these embody a pragmatic solution to the reference problem. The next simplification reactive planners perform is to formulate the planning problem as: what action to take *next*. This simplification is in accord with the nature of the world – it is pointless to calculate action two or more time steps from *now*, because the world may cease to be in a state that allows for the execution of these action sequences. Our main claims in this paper are as follows.

1. Perception is primarily indexical. An agent perceives objects and properties in the world relative to itself. A bicycle in front of an agent is perceived as *the bicycle-in-front-of-me* and not as a bicycle in some absolute spatio-temporal coordinate frame. Further it is seen as some bicycle, as opposed to a previously known bicycle because it takes effort to determine which bicycle it is.

2. When the rules of the world change slowly relative to the actual state of the world at any given time, this approach of reasoning with indexical theories is a computational win. The Horswill robot is a good example: the method of chasing objects in a world is hardwired into his robot, yet the specific object being chased can vary dynamically. Indexical theories do not make as many distinctions as an objective situation calculus theory of the world would permit. This is exactly the reason why the frivolous fruitcake [Ginsberg] problem is easy for reactive planners and not so for classical ones. A classical planner first discovers that a tower is to be built bottom up, this is hardwired into a reactive agent. A classical planner is also able to distinguish (uniquely name) every block in the world, a reactive planner focuses on exactly the relevant blocks.

3. Under what conditions can a theory be indexicalized? Indexical ontologies which pick out properties or objects in the world to direct action towards, can allow the construction of concise propositional prescriptions for action. The problem of automatically generating appropriate indexicals is a difficult problem, and this is clearly revealed in our attempts to build large-scale reactive systems. The hardest part of designing these agents is to pick appropriate distinctions (indexicals) in the world that determine the agent's next action.

4. But this use of indexicals only really makes sense in domains where perception plays a key role in determining action. Abstracting away from perception as in [Hendler] does not give us the same computational benefits. One could indexicalize the N-queens theory, however we still will not be able to get a constant-time connection between perception and action. It appears that successful indexicalization relies on there being room in the domain for streamlining perception and action such that a significant part of the work could be offloaded to the perceptual system.

In short, reactive planners work because their designers carefully and explicitly separate the changing aspects of the world from the stable ones, at design time. The rules of the world are compiled into propositional theories with indexicals which allow them to access the dynamic aspects of the world via sensors. So there are two parts to the generation of a reactive agent given the rules of the world and goals of this agent: the choice of appropriate indexicals and methods of achieving reference to them, and making the specification of action that is implicit in the rules, into a direct prescription for linking percepts to effectory output.

3 Indexicals and Perception

Logic with the objective world as a model makes distinctions that are neither sustainable nor desirable. Take the following situation: we have a pile of diamond-like stones and an expensive mechanism for determining whether or not a particular stone is a diamond. We could formulate a plan to obtain a diamond from this heap as follows: Pick up x such that x is a diamond. Classical objective reference would require us to construct a complete world model that determines the nature of each stone, and maintain a correspondence between the world model and the individual stones in order to sustain the plan above. Doing the tests on the stones counts as perception, and here we see that perception is doing far more work than necessary, A reactive agent with the ability to refer indexically could embody the following plan: pick up a stone, test if it is a diamond, if it is, hang on to it. The indexical reference machinery will maintain the correspondence between object held and diamond. This example demonstrates that objective reference has two costs: one, building the world model, and two, maintaining the correspondence between model and world.

We come upon a new variant of the frame problem while addressing the issue of acquiring and maintaining reference to objects across time. If at a given time t, the system establishes a reference to a certain object. If that object moves in the interval t to t+1, we need to track it across frames to make sure that the reference that was established in the previous time point carries through to the next. Which reference achievements at time t can we copy to time t+1? This problem becomes worse when we think of bee swarms or ant swarms that constitute and reconstitute over successive time ticks. The reason this is an issue for us is because: suppose we have a rule that says

$\exists x.red(x) \wedge apple(x) \Rightarrow Pickup(x)$

as one of our stimulus-response rules. It is difficult to guarantee that the robot will pickup the red apple bound to the x on the LHS of the rule because the evaluation of the LHS will take time and before the pickup action can reach out for the apple, the apple in that position may be gone or may not be the same red apple that we had originally picked out of the visual field. If we picked up an arbitrary red apple then we would not be faithful to the logic. For instance, to ask Robbie to pick up never before seen objects is contradictory. The red apple example is a case of a statement with unsustainable reference properties. In objective ontologies, there is need to maintain referential continuity across snapshots in time. An indexical variant of the above would not impose these referential requirements. A good choice of indexicals will take into account the fact that reference is not free – there is a cost of acquiring reference to objects and maintaining that connection for the duration of an action.

4 The Derivation of Indexical Theories

Our enterprise is to explore the automatic construction of agents given a declarative description of the world and what needs to be achieved in the world and a description of the perceptual capabilities of the agent. The creatures generated by Brooks are existence proofs of the fact that such agents can be generated from specifications (in particular, that humans can generate them). By analogy with VLSI, it is clear that the only way we can scale this method of agent creation is to derive large scale theories that link perception to action in a world by composing small ones in a systematic, hierarchical manner. This composition cannot be entirely described at the level of the implementation medium, be it FSMs or combinational circuitry. The abstractions and tools developed so far for generating large scale mappings are little more than programming languages to aid in the synthesis of finite state machines. We believe that indexicals have a primary role in this synthesis hierarchy. We describe below our plan to attack this problem.

We will first synthesize a plan for our goal using a situation calculus description of the world. The plan constructed makes the assumption that it has access to a complete world model and also that it has actions that are guaranteed to succeed. We will treat this plan as a functional specification of what needs to be implemented on a concrete robot with specific sensors and effectors. This synthesis problem is very similar to the problem of generating VLSI circuits where we go from an abstract description of behaviour to a physical piece of hardware that achieves that behaviour.

To do this, we need to assess the demands that the plan steps make on the perceptual and motor systems. This is a compilation problem: we need to compile an abstract sequence of actions into a particular sensor and effector architecture to have a realizable plan. Take for instance, the following plan (read from the bottom-up) synthesized using Plaisted's axiomatization of the monkeys-and-bananas problem.

```
(holds (has monkey bananas)
 (do (grab monkey bananas l2 ceiling)
           (do (climb monkey ladder l2)
       (do (drop monkey ladder l2 floor)
                  (do (carry monkey ladder l0 l2)
              (do (grab monkey ladder l0 floor)
                  (do (go monkey l1 l0) s0))))))))
```

From the plan above, we would like to synthesize the following situation action rule set that will solve this problem most effectively. Items in sans-serif are actions that can be executed by the robot. Items in italics are given by the sensors on the robot.

¬ *at-m-ladder*	⇒	move-toward-ladder
¬ *at-ladder-bananas* ∧ *at-ladder-m*	⇒	carry-ladder-toward-bananas
below-m-bananas ∧ *at-m-ladder* ∧ ¬ *up-m-ladder*	⇒	climb-one-step-up-ladder
below-m-bananas ∧ *up-m-ladder* ∧ ¬ *have-m-bananas*	⇒	grab-m-bananas

We can illustrate some of the issues in this compilation by analogy with a relational database – the situation calculus plan reads in the entire database (describing the world) at every step along the way. The situation-action rule set only queries specific aspects of this large database depending on the current goal to be achieved. So the large database describing the world is not instantiated at every time tick, but relevant aspects of it are generated on demand in response to finding the referents (or truth-values) of indexical terms. XS

Take the first step of the classical plan which involves the monkey getting to the ladder. The postcondition for this step is the monkey being at the ladder. The monkey needs to sense this condition to determine that this step of the plan has been achieved. Since the monkey does not have a single action that will move it from its current location to the ladder, it iterates an indexical action: move-in-direction-of-ladder, until the condition at-monkey-ladder is achieved. We can calculate whether or not this condition can be sensed given a description of the sensory apparatus on the monkey. We can also determine whether or not the action can be achieved given information about whether the monkey can move in an arbitrary direction along the floor. We can construct an algorithm that can go from the classical plan to the reactive rule set.

In sum, we will let the classical planner vectorize (or convert to Markov form) the world, so we can compute the next action based purely on the situation now. The classical plan above has given us the general sequence of steps to be achieved. The reactive plan achieves this specification for us on an actual robot given descriptions of the sensors and effectors at hand.

5 Debunking of Myths

1. Indexicality *is* variable binding.
   There is a type clash here. There is no way we can have a *cons* cell in the head of an agent, whose *cdr* is the object in the world. All we can hope for are correspondence accounts, and meta-theoretically prove that the correspondence between tokens in the agent's head and objects in the world is maintained by the reference machinery of the robot. Indexical reference can be *implemented* as variable binding only if we have a complete world model in the agent's head. When that is not the case, the achievement of indexical reference requires active computation on the visual field of an agent and machinery to track these referents across time.

2. Providing model theoretic semantics for indexical knowledge is enough.
   The mechanics of reference is very important to the enterprise of designing agents that act in the world. Model theoretic accounts of indexicals as in [Lesperance,Kaplan] are important from the perspective of assessing the logical content of indexical theories. Unfortunately, we cannot express concerns about reference to objects within the context of model theory. The need for generating action requires us to deal with reference and give a more pragmatic semantics to indexicals in terms of the properties of perceptual procedures that extract the referent of an indexical in the visual field and maintain that reference across time.

3. Simply adding indexical terms to the ontology is enough
   In our Kr89 paper, we showed that the addition of indexical terms into the situation calculus ontology allowed us to propositionalize action computation. In fact, we took a detailed theory of the blocks world and made it into a Markov theory using indexicals – the reason we were able to derive a very efficient reasoner from the computationally intractable situation calculus description was because we made the computation of next action be based entirely on the situation at the current time, and we erased distinctions that an objective ontology would have forced us to make thus reducing the burden on the reference machinery. So merely adding indexical terms does not buy us much: it was what the indexicals allowed us to do that contributed to their effectiveness.

We demonstrate that by taking advantage of the indexicality of perception and with our scheme for synthesizing indexical theories using information about the goals to be achieved and the perceptual capabilities of an agent, we can build large scale reactive systems.