Naming Observed Objects (revised)

Anthony S. Maida[*]
Steve Giambrone[†]
Heng Zhou[‡]
University of Southwestern Louisiana
Lafayette, LA 70504-4330
{maida,hz}@cacs.usl.edu, sxg8037@usl.edu

Keywords: autonomous agents, distributed AI, language and perception

Abstract

This paper sketches an operational semantics and protocol for naming observed objects. The semantics and protocol are motivated by a philosophical account of reference. We illustrate the protocol on an example in which two agents jointly explore an unknown domain and exchange information about observed objects. The protocol is examined for compatibility with two forms of default reasoning. We then raise problems for it, and discuss possible solutions.

1 Introduction

We address the following question. How can agents exchange information about unanticipated objects that they discover as they explore a novel, open environment?

This paper sketches an operational semantics for naming observed objects. We examine the problem of mapping a domain where agents acquire information through a visual modality and communicate through a linguistic modality. We shall describe a protocol where agents assign names to the objects that they find. We assume agents have advanced vision systems which partition the world into objects.

Like [Kronfeld, 1990] this paper draws on a philosphical account of reference as a source of ideas to construct a computational model for naming observed objects. More specifically, we appeal to a specific form of a causal theory of reference. We now give a brief introduction to the ideas and terminology which we use.

Following general philosophical usage, we reserve the term *denotation* for reference via description, e.g., 'the first U.S. president', 'the author of our text'. Following [Devitt, 1981], we use the term *designation* when discussing reference via proper names and the term *reference* for either.

Causal theories of reference were first independently enunciated in [Putnam, 1973] (for natural kind term) and [Kripke, 1972] (for names). Devitt and Sterelny [Devitt and Sterelny, 1987] (with much subsequent refinement) offer a general causal theory: The referent of any term is ultimately causally determined. Since we are here concerned only with reference for names, we restrict our attention to designation. Put (overly) simply, on this theory the designation of a term is the determinate cause of the perceptual experience(s) which evoke the initial introduction of the term, a "**dubbing**," and any similar subsequent causal **grounding** uses of the term. A simple example will make the main features of such an account clear.

The astronomer, Jean, discovers a new star one night through his or her telescope and names it 'StellaJ'. The term 'StellaJ' is then causally linked to this type of thing via Jean's perceptual contact with it in the naming or *dubbing* situation. It was the object that Jean had appropriately in mind for the naming.

Jean naturally tells his or her colleague, Joe, about StellaJ. Having heard the term used by Jean, Joe can and does use 'StellaJ' to refer to the same object. He **borrows** the term from Jean. Joe's uses of 'StellaJ' refer to the star, because they are causally linked to it through Joe's having heard (again, a perceptual causal event) Jean's use of it, which use itself was causally connected to the star — ultimately by the initial dubbing. Other people hear Joe and Jean talk about StellaJ (a causal perceptual event) and borrow the term from them. (There is a **causal chain** linking their subsequent uses of the term with the original referent. It is called a 'designating chain', or simply 'd-chain'.) The term spreads through the community of speakers and through generations thereof via such borrowing which establishes a multiply-branching causal chain extending from uses (tokens) of the term to its referent.

Of course, other astronomers will sight StellaJ, and say things like "So that's StellaJ, is it?" and "My, StellaJ is a red dwarf," and so forth. Such uses of the term provide further *causal groundings* of the term in its referent since they are made while the speaker is in perceptual contact

[*] Center for Advanced Computer Studies
[†] Philosophy Department
[‡] Center for Advanced Computer Studies

with StellaJ. Thus uses of the term become **multiply grounded** in the star.

2 Previous Work

Like [Fodor, 1975], [Devitt, 1981], and [Sterelny, 1990], we give priority to a language of thought. Of particular importance here is the fact that an agent's *internal object representations* (IOR's) are the primary bearers of reference [Appelt and Kronfeld, 1987]. The barn scenario appears in [Maida, 1992]. The proposal to generate symbols for each observation event appears in [Maida, 1993]. The relationship between referent identification and object recognition is discussed in [Maida and Tang, 1994]. Belief revision issues related to this problem are discussed in [Tang and Maida, 1994; 1995]. The present paper focuses on policies of assigning names to observed objects.

3 The Barn Scenario

Consider the following problem. Two agents cooperate to explore a domain and generate a pooled inventory of objects that they find. The example below takes place in the context of two autonomous agents exploring a farm. The initial domain state is the following:

> `Agent1` and `Agent2` are initially outside a barn which has front and rear entrances. Inside, there is a tractor near each entrance. The barn is L-shaped and if an agent sees a tractor at one entrance, the agent will not see the other tractor. The agents have not seen the tractors. The tractor at the rear entrance of the barn is noticeably broken.

In the scenario below, agent1 tells agent2 about the existence of one of the tractors. Later, agent2 discovers a second tractor but misidentifies it as the tractor that agent1 was referring to. The question we examine is how agents can agree on names for objects under conditions in which misidentifications occur.

Event 1: Agent1 enters the barn at the front entrance, observes the tractor at that entrance, and exits.

Event 2: Agent1 tells agent2 that *there is a tractor (:= N1) in the barn.*

Event 3: Agent2 enters the barn using the rear entrance, observes the broken tractor at the rear entrance, assumes this is the tractor agent1 referred to, and exits.

Event 4: Agent2 says to agent1 that N1 is broken.

Event 5: Agent1 tells agent2 that N1 is definitely not broken.

In event2, agent1 uses a description to refer to the object that it observed during event1. Let us assume that agent1 also knows that the tractor is not broken but did not bother to mention this in its description. The notation ":= N1" means that agent1 has baptized this object with the name "N1" and that agent2 will use this name when it wants to discuss this object with agent1. During event3, agent2 has misidentified a different tractor as being the referent of N1. In event4, agent2 intends (in part) to make a statement about the tractor it observed. However, the referent of "N1" is fixed to designate a different object. This situation illustrates that speaker reference can and sometimes will differ from semantic or conventional reference.

4 Protocol for Object Naming

We now describe our first attempt to specify an operational semantics for naming observed objects. We shall discuss policies for symbol grounding, object naming, and object recognition. When observing objects, our agents abide by the following symbol grounding policy.

Observing: Whenever an agent observes an object, it generates a new internal, perceptual symbol. If the agent observes the same object more than once, it generates a new symbol for each event.

We say that the symbol designates the object that caused it to be generated. Following [Appelt and Kronfeld, 1987], these symbols are called "perceptual internal object representations (perceptual IOR's)." Note that no perceptual IOR will ever designate more than one object.

Since an agent generates a new IOR whenever it observes an object, the agent must infer equalities on these IOR's in order to recognize anything. IOR's in an agent's knowledge base form equivalence classes based on the agent's beliefs of which symbols corefer.

New equivalence classes: For any perceptual IOR, if the agent is unable to infer that the IOR corefers with an IOR already in its knowledge base, then the agent places the IOR in its own singleton class.

The term "name" refers to a symbol in an agent communication language (ACL, [Genesereth and Ketchpel, 1994]). The symbol grounding policy for names is expressed by the following two rules:

Naming: During communication, whenever an agent wishes to refer to an object denoted by an existing equivalence class, it uses one of the names associated with the class. If the class does not have a name, the agent generates a new name to associate with the class. (The agent *dubs* the object.)

Receiving names: When an agent receives a name, it generates a new discourse IOR.

- **if:** the agent receives the name for the first time, it puts the IOR in a singleton equivalence class and associates the name with the class. (The agent *borrows* the name from the sender.)

otherwise: the agent puts the IOR in the already existing equivalence class that is associated with the name — unless doing so would result in holding contradictory beliefs. Such cases are discussed below.

The discourse IOR's [Appelt and Kronfeld, 1987] are acquired as a result of receiving names.

5 Naming Dynamics

This section walks through the barn scenario explaining how representations are acquired and how names are attached to these representations. An equivalence class is described by a 4-tuple with fields for *name*, *original IOR*, *current IOR's*, and *properties*. This representation allows us to direct our attention to the task of assigning names to equivalence classes without distraction from the details of managing default beliefs. For instance, the tuple below describes agent1's knowledge state at the end of event2.

Agent1: [N1, s1, <s1>, <tractor, in-barn, ¬broken>]

At this point, agent1 models the existence of only one object so there is only one equivalence class in its knowledge base. The name "N1" is a symbol in the ACL which designates the entity that the original IOR in the equivalence class designates. In this example, "N1" designates whatever s1 designates. The variables "s1," "s2," ...refer to distinct IOR's. The agent generated the IOR s1 as a result of the observation event, event1, placed it into a new singleton designation equivalence class, and attached the properties of being a tractor inside of a barn and not broken. It then assigned the set the name "N1" during event2. This was done according to the policies of *Observing* and *Naming*.

At the end of event2 after receiving the message from agent1, agent2 has one equivalence class in its knowledge base. This class was created according to the policy of *Receiving names* and is described by the tuple below.

Agent2: [N1, s2, <s2>, <tractor, in-barn>]

During event3, agent2 observed an object and generated the perceptual IOR s3 to denote it. However, at the end of event3, agent2 still has one equivalence class in its knowledge base. This reflects the fact that agent2 has made a referent identification inference on IOR's s2 and s3. This is shown below.

Agent2: [N1, s2, <s2, s3>, <tractor, in-barn, broken>]

During event4, agent1 receives the message from agent2 that claims N1 is broken. Agent1 generates a discourse IOR, s4, for the received name and attempts to add it to the equivalence class for which "N1" is associated. This fails because agent1 already knows that the object denoted by s1 is not broken. Therefore, agent1 must create a separate equivalence class. Here agent1 faces some choices: Attach N1 to the new equivalence class also? Attach N1 to the new equivalence class and remove it from the other equivalence class? Do not attach N1 to the new equivalence class and leave it attached to the original equivalence class? Or do not attach N1 to the new equivalence class and remove it from the original equivalence class? At this point we can afford to follow Kripke [Kripke, 1972] and insist that the dubbing or naming event overides other groundings of the name. Since N1 is attached originally to s1 in agent1 and since s1 is a perceptual IOR and s4 is a discourse IOR, agent1 will leave N1 attached to s1's equivalence class and will not attach it to s4's equivalence class. So the new equivalence class does not have a name associated with it, since agent1 has not yet attempted to communicate about the object referred to by s4.

Agent1: [N1, s1, <s1>, <tractor, in-barn, ¬broken>]
[—, s4, <s4>, <tractor, in-barn, broken>]

During event5, agent2 receives a message from agent1 stating that the object denoted by "N1" is not broken. This forces agent2 to retract the equality between IOR's s2 and s3. This leaves agent2's perceptual IOR s3 unattached to an equivalence class. The *New equivalence class* policy specifies that agent2 must place s3 into a singleton equivalence class. At the end of event5, agent2 has revised its belief state to that shown below.

Agent2: [N1, s2, <s2>, <tractor, in-barn, ¬broken>]
[—, s3, <s3>, <tractor, in-barn, broken>]

6 Complications

6.1 Equivalence Classes

Consider a more complicated scenario where agent2 exhaustively searches the barn and discovers both tractors. Assume, temporarily, that agent2 does not make referent identification inferences. Let the IOR's be the same as in the previous example except that agent2 uses the additional IOR s5 to represent its observation of the tractor which is not broken. Agent2's knowledge base is shown below.

Agent2: [N1, s2, <s2>, <tractor, in-barn>]
[—, s3, <s3>, <tractor, in-barn, broken>]
[—, s5, <s5>, <tractor, in-barn, ¬broken>]

Agent2 has evidence that IOR s3 is coreferent with s2 and also that s5 is coreferent with s2. Further, the evidence is equally strong. In both cases, two out of three properties match and there are no mismatches. Finally, agent2 has the information to know with certainty that s3 is not coreferent with s5. This situation is nearly isomorphic to the Nixon diamond[1] problem in default reasoning [Touretzky, 1986].

- There is evidence that s2 is coreferent with s3.
- There is equal evidence that s2 is coreferent with s5.
- s3 is not coreferent with s5.

A reasoner can take two approaches to this problem. One involves nonmonotonic inference, such as default reasoning, which does not place particular emphasis on weighing evidence [Reiter, 1978]. The other is evidential, such as probabilistic reasoning [Pearl, 1988]. We address both approaches.

Default Reasoning

Default reasoning in the style of Reiter [Reiter, 1978] is compatible with the object naming protocol we described. Consider the default inference schema:

$$\frac{C(x = y)}{x = y}$$

The rule states that if it is consistent to conclude x equals y, then conclude that x equals y. Agent2 could use this rule to conclude that s2 and s3 are coreferent. Once this happens, the conclusion that s2 and s5 are coreferent would be blocked until agent2 retracted the previous conclusion. The naming protocol fits with this process.

To illustrate, let us revisit the barn scenario and change event3 to read as follows.

> **Event 3:** Agent2 enters the barn using the rear entrance, observes the broken tractor at the rear entrance and assumes this is the tractor agent1 referred to. After this, agent2 sees the tractor at the front of the barn and then exits.

Agent2's knowledge state at the end of event3 is the following.

Agent2: [N1, s2, <s2, s3>, <tractor, in-barn, broken>]
[—,s5, <s5>, <tractor, in-barn, ¬broken>]

During event5, agent2 receives the information that N1 is not broken. This causes agent2 to retract its default conclusion that s2 and s3 are coreferent. At this point, agent2 is free to draw the default conclusion that s2 and s5 are coreferent. At the end of event5, agent2 now has the revised knowledge base shown below.

[1]Nixon is a Quaker; Nixon is a republican; Quakers are pacifists; republicans are not pacifists. Is Nixon a pacifist?

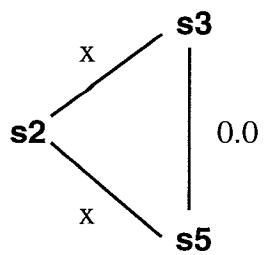

Figure 1: Triangle of equalities.

Agent2: [N1, s2, <s2, s5>, <tractor, in-barn, ¬broken>]
[—, s3, <s3>, <tractor, in-barn, broken>]

Evidential Reasoning

Under some circumstances, evidential reasoning is compatible with the naming protocol. Consider the graph of IOR's shown in figure 1. The edges indicate the agent's equality beliefs and are labeled with certainty-levels.

Agent2 knows with certainty that the equality connecting s3 and s5 is false, so the edge is labeled 0.0. There is equal positive evidence that each of the equalities connecting s2 and s3, and, s2 and s5, is true. Therefore, whatever the certainty-level is, it must be the same for both edges. Finally, agent2 knows that at most one of the edges can be true. Therefore, any rational scheme for evaluating evidence must assign the value of x to be between 0 and 0.5. Further a threshold for believing a default statement should be greater than 0.5. Otherwise, an agent might believe an explicit inconsistency.

Given these considerations, agent2 would not believe any positive equality statements between s2, s3, and s5 at the end of event3. Therefore, event4 in the barn scenario would not take place. Suppose, however, that agent1 happened to mention to agent2 that N1 is not broken. Agent2 would then have the information to set the certainty-level of the equality connecting s2 and s3 to 0. The certainty-level of the equality connecting s2 and s5 would then increase. If it went above agent2's threshold for acquiring a default belief, then agent2's final equivalence classes would match those in the previous subsection. In this case, the naming protocol is compatible with evidential reasoning.

Unfortunately, it seems possible to construct examples where the certainty-level of more than one equality statement goes above threshold for saying that it is a default belief. Consider a situation where it is known that exactly $n-1$ out of n distinct IOR's are coreferent with s2 but it is not known with certainty which of these IOR's are coreferent with s2. If no other information is available then the certainty-level of each of these equality statements must be assigned the same value of $\frac{n-1}{n}$. As n gets large, the certainty-value gets arbitrarily close to 1. Eventually, all n of these statements will exceed the threshold for belief. In this situation, agent2's knowl-

edge base cannot be partitioned into equivalence classes because s2 would fall into two different classes. Under these conditions, the protocol needs to be generalized to operate in a reasonable fashion.

6.2 Erroneous Property Attributions

Recall that in the original scenario agent1's knowledge state after event3 and before event4 is described by the expression:

Agent1: [N1, s1, <s1>, <tractor, in-barn,¬broken>]

When event4 occurs, agent1 is informed by agent2 that N1 is broken. However, agent1 knows with certainty that N1 is not broken. In the following, we analyze agent1's reaction to this erroneous property attribution. When agent1 receives the message that N1 is broken, agent1 has the information to know that agent2 is referring to some object but not the object named N1. Since agent2 is misapplying the name N1 to this other object, agent1 knows that agent2 misidentified N1. Further agent1 plausibly infers that the misidentified object has the same properties as those of N1 except for the erroneous property. So, at the end of event4, agent1 reaches the knowledge state described below.

Agent1: [N1, s1, <s1>, <tractor, in-barn, ¬broken>]
[—, s4, <s4>, <tractor, in-barn, broken>]

The symbol s4 designates the object misidentified as N1 by agent2.

One may ask whether this procedure works in general? To answer this, we revise the barn scenario as follows. We add a property, color, to the domain, and assume that the tractor at the front entrance is red and the tractor at the rear entrance is green. In the context of the revised scenario, the knowledge states of agent1 and agent2 at the end of event3 are the following.

Agent1: [N1, s1, <s1>, <tractor, in-barn, ¬broken, red>]

Agent2: [N1, s2, <s2, s3>, <tractor, in-barn, broken, green>]

After event4, if agent1 processes the erroneous property using the method mentioned above, then, at the end of event4, agent1's knowledge state is the following.

Agent1: [N1, s1, <s1>, <tractor, in-barn, ¬broken, red>]
[—, s4, <s4>, <tractor, in-barn, broken, red>]

Unfortunately, there is no object in the barn whose properties match those of s4. The conclusion is inaccurate.

Let us review the scenario. At event2, agent1 informs agent2 of two properties of N1, tractor and in-barn. So agent1 knows that agent2 knows the two properties of N1. During event3, agent2 plausibly infers that s2 (named N1) equals s3. At event4, because agent1 knows that agent2 knows the two properties of N1, and he knows the procedure for inferring indentities and the protocol for assigning names, agent1 can infer that the object misidentified by agent2 has properties: tractor, in-barn, and broken. Thus, at the end of event4, agent1's knowledge state should be described as below.

Agent1: [N1, s1, <s1>, <tractor, in-barn, ¬broken, red>]
[—, s4, <s4>, <tractor, in-barn, broken>]

Now let us summarize this procedure. First, agent1 must remember what he said to agent2 before. After agent1 receives the erroneous information from agent2, he infers that there is an object whose properties consist of the erroneous property and those properties told to agent2 before. An advantage of the new method is that an agent can infer some properties of the object misidentified by another agent in the most extent. But the method requires that an agent remember what messages he has sent and to whom. This is called a history-based method.

6.3 Certainty Levels

Let us consider another revision of the original scenario. In the new scenario, there is an agent3. The initial states of agent1 and agent2 remain unchanged. In order to discuss certainty levels, we divide a communication event into two events: informing and being informed. The new scenario is given below is shown in figure 2.

Event 1: Agent1 enters the barn at the front entrance, observes the tractor at that entrance, and exits.

Event 2: Agent1 tells agent2 that there is a tractor (:=N1) in the barn.

Event 3: Agent2 hears from agent1 that there is a tractor (:=N1) in the barn.

Event 4: Agent2 enters the barn using the rear entrance, observes the broken tractor at the rear entrance, assumes this is the tractor agent1 referred to, and exits.

Event 5: Agent2 says to agent1 that N1 is broken.

***Event 6:** Agent2 says to agent3 that N1 is broken.

Event 7: Agent1 hears from agent2 that N1 is broken.

***Event 8:** Agent1 tells agent3 that N1 is not broken.

***Event 9:** Agent3 hears from agent1 that N1 is not broken.

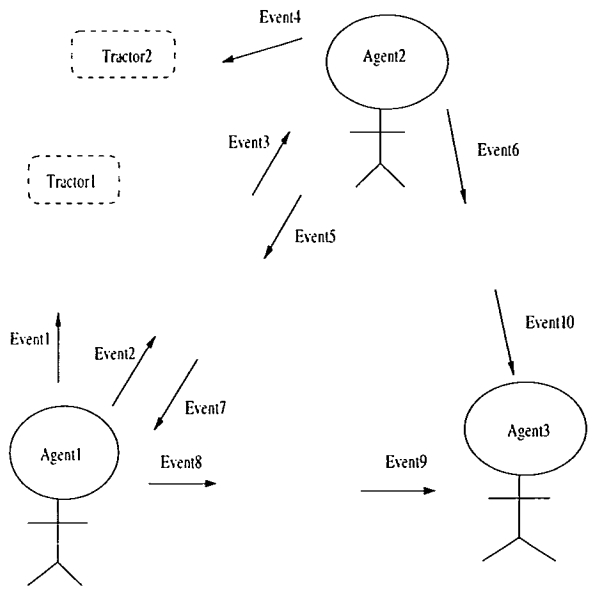

Figure 2: Scenario of three agents.

***Event 10:** Agent3 hears from agent2 that N1 is broken.

Let us focus on the events noted by asterisks. During event6, agent2 says to agent3 that N1 is broken, while during event8 agent1 says to agent3 that N1 is not broken. Due to uncertain communication delays between agents, the order in which agent3 receives the messages from agent1 and agent2 is not known. If agent3 hears from agent1 first (event9 occurs before event10), then at the end of event9, agent3 thinks that N1 is not broken. Then when event10 occurs, agent3 cannot decide whether or not N1 is broken because none of agent3's beliefs about N1 are based on his own observation, but rather from messages sent by others. Let us fix the problem.

During event1 agent1 observes the tractor, sees that it is not broken, and refers to it as N1. So, agent1 knows that N1 is not broken. But after event4, agent2 has only partial justification for believing that N1 is broken. Agent1's evidence that the tractor he had observed was in fact N1 was only plausible. Clearly, the desired outcome is that agent3 believe agent1 and not agent2. One way of arriving at that result would be: 1) that agents assign a modality — either plausible or certain, symbolised as $\Diamond$ and $\Box$ respectively — to the propositions that they believe and communicate; and 2) agents resolve conflicting information in favor of certainty where possible. So, let us re-examine the 3-agent scenario again with this emendation.

Let the IOR's be the same as in the previous examples for agent1 and agent2. Assume that agent3 produces s5 for event9 and s6 for event10. At the end of event3, the knowledge states of agent1 and agent2 are the following.

Agent1: [N1, s1, <s1>, <$\Box$tractor, $\Box$in-barn, $\Box\neg$broken>]

Agent2: [N1, s2, <s2>, <$\Box$tractor, $\Box$in-barn>]

During event4, after agent2 observes the broken tractor before making any inference, agent2's knowledge state is below.

Agent2: [N1, s2, <s2>, <$\Box$tractor, $\Box$in-barn>]
[—, s3, <s3>, <$\Box$tractor, $\Box$in-barn, $\Box$broken>]

When agent2 makes the inference that s2 is equal to s3, the conclusion is only plausible, and hence so is the further conclusion that N1 (i.e., s2) is broken. So at the end of event4, agent2's knowledge state is the following.

Agent2: [N1, s2, <s2, s3>, <$\Box$tractor, $\Box$in-barn, $\Diamond$broken>]

Let us skip event 5, 6, and 7, and concentrate on event9 and event10, looking particularly at agent3's knowledge state. We consider two cases; event9 occurs before event10, or event10 occurs before event9. In the first case, at the end of event9 agent3's knowledge state is the following.

Agent3: [N1, s5, <s5>, <$\Box\neg$broken>]

At the end of event10, after agent3 hears from agent2 and before it makes any inference, agent3's knowledge state is as below:

Agent3: [N1, s5, <s5>, <$\Box\neg$broken>]
[N1, s6, <s6>, <$\Diamond$broken>]

Because $\Box\neg$broken is stronger than $\Diamond$broken, agent3 believes that N1 is not broken. Thus at the end of event10, agent3's knowledge state is below.

Agent3: [N1, s5, <s5>, <$\Box\neg$broken>]
[—, s6, <s6>, <$\Diamond$broken>]

It is straightforward to see that agent3's final belief state will be the same for the second case, i.e., when event10 occurs before event9.

Agent3: [—, s6, <s6>, <$\Diamond$broken>]
[N1, s5, <s5>, <$\Diamond\neg$broken>]

Evidently no matter what case occurs, agent3 will always believe that N1 is not broken.

In this subsection, we have assumed that the grainsize of information between agent communication is the same as that of knowlege in their knowledge bases, that is, the same certainty levels are applied both to messages between agents and to agent's knowledge bases.

7 Discussion

The outlines of some features of language and perception mediation are suggested by the results we have. For instance, language processors must be able to accept the symbols formed by perception, or else there must be mediated communication between the two modules with the intermediate processor being able to access and intertranslate the symbols of both. Also, with agents having multiple perceptual interactions with the same objects, one must sooner or later provide for shedding IOR's from the the current IOR store. One issue would soon have to be adressed, namely having more than one name for the same object.

However, a few general comments can be made. Like Appelt and Kronfeld [Appelt and Kronfeld, 1987], we rely heavily on epistemic features for working out the pragmatics of naming and name use, as we did in section 6.3. This is not surprising. But we also take modes of representation to be a central epistemic matter and will be investigating alternative modes for efficient — and we think, realistic — solution to many of the problems to be faced.

References

[Appelt and Kronfeld, 1987] Douglas E. Appelt and Amichai Kronfeld. A computational model of referring. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 640–647, Milan, Italy, August 23–28, 1987.

[Brachman and (Eds), 1985] Ronald Brachman and Hector Levesque (Eds). *Readings in Knowledge Representation.* Morgan Kaufmann Publishers, Los Altos, CA, 1985.

[Devitt and Sterelny, 1987] Michael Devitt and Kim Sterelny. *Language and Reality.* Blackwell/MIT Press, Oxford, MA, 1987.

[Devitt, 1981] Michael Devitt. *Designation.* Columbia U Press, New York, New York, 1981.

[Fodor, 1975] Jerry Fodor. *The Language of Thought.* Thomas L Crowell, New York, New York, 1975.

[Genesereth and Ketchpel, 1994] Michael R. Genesereth and Steven P. Ketchpel. Software agents. *Communications of the ACM*, 37(7):48–53, 1994.

[Kripke, 1972] Saul Kripke. Naming and necessity. In Donald Davidson and Gilbert Harman, editors, *Semantics of Natural Language.* Reidel, Dordrecht, 1972.

[Kronfeld, 1990] Amichai Kronfeld. *Reference and Computation: An essay in applied philosophy of language.* Cambridge University Press, New York, 1990.

[Maida and Tang, 1994] Anthony Maida and Shaohua Tang. Description-based communication for autonomous agents under ideal conditions. Technical Report TR–94–2–3, Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, LA 70504–4330, April 1994.

[Maida, 1992] Anthony S. Maida. Knowledge representation requirements for description-based communication. In B. Nebel, C. Rich, and W. Swartout, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, pages 232–243. Morgan Kaufmann, San Mateo, CA, 1992.

[Maida, 1993] Anthony S. Maida. Propositionally representing incomplete knowledge about existence. *Journal of Experimental and Theoretical Artificial Intelligence*, 5(1):185–197, 1993.

[Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[Putnam, 1973] Hilary Putnam. Meaning and reference. *Journal of Philosophy*, 70:699–711, 1973.

[Reiter, 1978] Raymond Reiter. On reasoning by default. In *Proc. TINLAP-2, Theoretical Issues in Natural Language Processing-2*, pages 210–218, 1978. reprinted in [Brachman and (Eds), 1985].

[Sterelny, 1990] Kim Sterelny. *The Representation Theroy of Mind.* Basil Blackwell, Cambridge, MA, 1990.

[Tang and Maida, 1994] Shaohua Tang and Anthony S. Maida. Temporal aspects of description-based communication for autonomous agents. In Frank D. Anger and Rasiah Loganantharaj, editors, *AAAI94 Workshop on Spatial and Temporal Reasoning*, pages 69–75, 1994.

[Tang and Maida, 1995] Shaohua Tang and Anthony S. Maida. Recovery from referent misidentifications in systems of communicating agents. In Henry Hexmoor and David Kortenkamp, editors, *Working notes of Spring 1995 AAAI Symposium: Lessons Learned from Implemented Software Architectures for Physical Agents*, pages 213–223, 1995.

[Touretzky, 1986] David S. Touretzky. *The Mathematics of Inheritance Systems.* Morgan Kaufmann, Los Altos, 1986.