# Recovery from Referent Misidentifications in Systems of Communicating Agents

*Shaohua Tang* **and** *Anthony S. Maida*
Center for Advanced Computer Studies
University of Southwestern Louisiana
Lafayette, LA 70504-4330
{sht,maida}@cacs.usl.edu
**Keywords:** autonomous agents, distributed AI

## Abstract

When interacting autonomous agents operate in open domains that contain novel objects, the agents may not be able to refer to the objects by using preassigned names. As an alternative, the agents might use descriptions. When agents refer to domain objects by using descriptions, referent misidentifications will occur. This paper models recovery from referent misidentifications in systems of such agents.

## 1 Introduction

This paper studies the problem of referent misidentification as it occurs in systems of interacting agents who refer to objects via descriptions. Figure 1 shows the situation. A finite number of agents (in this case, two) operate in a domain with a finite number of unnamed objects. Agent2 wants to send information to agent1 about one of the objects. To do this, agent2 must indicate which object it wants to talk about. Although domain objects don't have preassigned names, they do have various properties and these properties have names that are known to all agents (e.g., small, red, flat). Therefore, an agent can refer to an object by specifying a set of properties that apply to the object. A set of properties that are used for this purpose is a *description*. The agent that generates the description expects that an agent who receives the description can identify the object by using those properties.

Referent misidentification (RMI) occurs when an agent receives a description but misidentifies the object that the sending agent was trying to refer to. RMI's are inherent to description-based communication in open domains as shown below and in [11]. To implement a description-based transaction language, we need methods to either prevent, or detect and recover from, RMI's.

The following example takes place in the context of two autonomous agents exploring a farm. It shows how RMI's can happen in the simplest domains. The initial domain state is:

> `Agent1` and `Agent2` are initially outside of a barn with 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.

The following scenario describes the history and ramifications of an RMI in this domain.

**Event 1:** Agent1 enters barn at front entrance, sees tractor at that entrance, exits.

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

**Event 3:** Agent2 enters barn using rear entrance, sees broken tractor at rear entrance, assumes this is the tractor agent1 refered to, exits.

**Event 4:** Agent2 says to agent1 that *the tractor is broken.*

**Event 5:** Agent1 tells agent2 that the tractor is *definitely* not broken.

The RMI occurs during event 3. Its occurrence depends only on the fact that agents don't have complete knowledge of domain objects. Therefore, RMI's in principle cannot always be prevented. Because of this, diagnosis of and recovery from RMI's is a general problem inherent to description-based communication. This paper is concerned with modeling RMI diagnosis and recovery for this example.

We have implemented a Common Lisp simulation for this example and all of the mechanisms described in this paper have been implemented.

### 1.1 Diagnosis and Recovery

Figure 2 shows the inference process agent2 goes through as it receives a description of a tractor in the barn, discovers a tractor in the barn, computes a referent identification inference, and concludes that the tractor which it was told about is broken. Let symbol "t1" be agent2's symbol for the tractor that agent1 described. Let symbol "t2" be agent2's symbol for the tractor that agent2 observed. Ellipses indicate beliefs and rectangles indicate inference steps creating new beliefs. The particular entities in the figure are explained below.

**t1-in-barn:** Agent2's representation of t1 being in the barn.

**t2-in-barn:** Agent2's representation of t2 being in the barn.

**Ref-id:** A defeasible inference step allowing agent2 to match descriptions for t1-in-barn and t2-in-barn.

**t1-eq-t2:** This equality was derived by agent2's referent identification inference.

**t2-broken:** Agent2's representation that t2 is broken.

**Subst-eq:** The inference step of substitution of equal terms.

**t1-broken:** Agent2's representation that t1 is broken.

After agent2 learns that the tractor is not broken, it has a belief revision problem. This paper models the communication between agent1 and agent2 and agent2's subsequent belief revision process.

## 2 Agent Architecture and Domain Ontology

This section describes our assumptions about agent architecture and domain ontology.

### 2.1 Domain Assumptions

The domain consists of a finite set of objects and a finite set of basic properties. By basic property, we mean a property that is (naively speaking) inherent

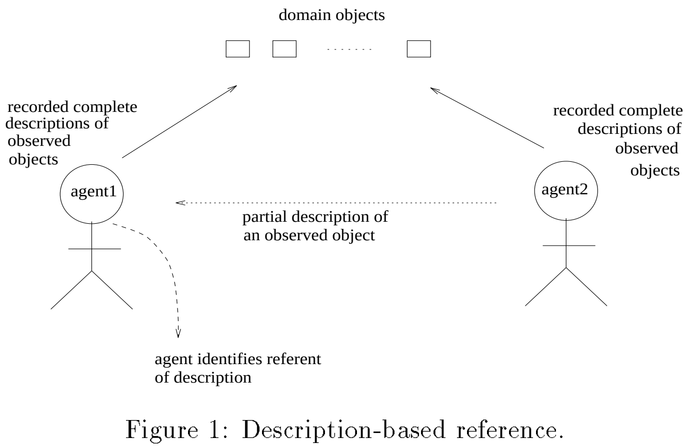


Figure 1: Description-based reference.

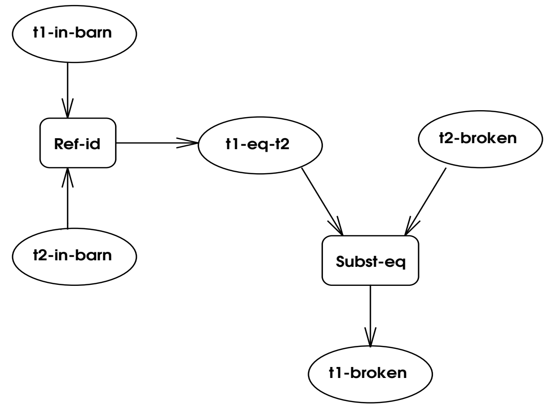


Figure 2: Trace of agent2's inference process.

to the object such as size, shape, or color. We exclude relational properties such as location. The basic properties of an object do not change over time and the names of these properties are standardized for all agents. We assume that no two objects have exactly the same set of basic properties.

### 2.2 Agent Assumptions

We model agents as being complete and accurate observers. When they observe an object they accurately notice all of the basic properties of the object. That is, if an agent observes an object, it will know with certainty whether that object has a basic property.

Agents also keep track of whether the information they have about an object is certain or plausible. The agent considers any information which is not 100 percent certain to be plausible and therefore retractable. Other than distinguishing between certain and plausible, agents do not judge the reliability of their beliefs.

Agents are honest, messages reach their destination without error, and are decoded properly by the receiving agent. When an agent sends a message to another agent, the sending agent indicates the certainty level it associates with that information. Consequently, if one agent sends to another agent a message saying, with certainty, that some object has a property, the agent who receives the message can also know, with certainty, that the object does indeed have the property.

The RMI scenario uses four kinds of actions. An action can be either *observing* an object, *receiving* a message, *sending* a message, or moving around in the domain to find objects. Actions which change location are not relevant to our analysis of communication, so we won't discuss them further.

### 2.3 Temporal Ontology

An agent is a process that operates autonomously, continuously, does not use a global system clock, and asynchronously sends messages. Therefore, our agents are a special case of Lamport's sequential pro-

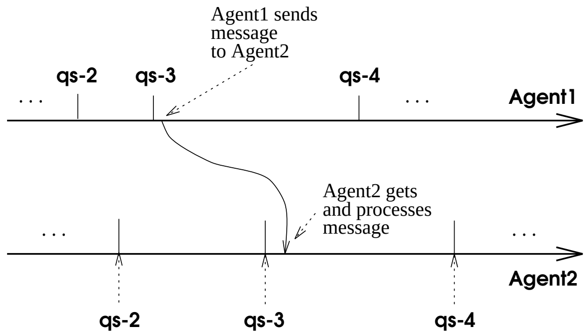


Figure 3: Comparing quiescent states of two agents.

cesses [9].

We assume that agents operate within a two-step variable-length control cycle: perform a primitive action, update mental state. Together, these two steps compose the body of the control cycle and are called the *active interval*. Some primitive actions, such as sending a message, may not require an agent to update its mental state. Other primitive actions, such as observing an object or receiving a message, do require an agent to update its mental state. Updating mental state consists of two substeps. First, the direct result of the previous action is added to the agent's knowledge base according to some constraints. Then inference processes (both sound and plausible) are applied to the agent's knowledge base. We assume the inference process terminates. We use the term *quiescent state* to refer to the boundary between two adjacent active intervals. The length of active intervals can vary because the action taken in the body of the interval and the amount of computation needed to update an agent's mental state can vary from one cycle to the next.

We describe belief change by refering to agent's beliefs at quiescent states. That is, we can state that an agent has a particular belief at, say, its $n$-th quiescent state but not its $n+1$-th quiescent state (nonmonotonicity).

Quiescent states within an agent form a totally ordered sequence in time. Quiescent states across the system of agents form a partial order. Particular quiescent states for two different agents can sometimes be temporally ordered if a certain history of message transmission events in the system can be identified. In figure 3, quiescent state 3 in agent2 must come after quiescent state 3 in agent1 because the former represents the beginning of agent1 sending a message and the latter represents the beginning of agent2 receiving the message. An agent cannot start receiving a message until after the sending process has started.

Figure 4 shows the RMI scenario. Message transmission events are subdivided into sending and receiving. For any two events in an agent's history, there must be at least one quiescent state between them. We only speak of an agent's beliefs at quiescent states because they mark situations where an agent's inference processes have stopped but the agent is not processing a new event.

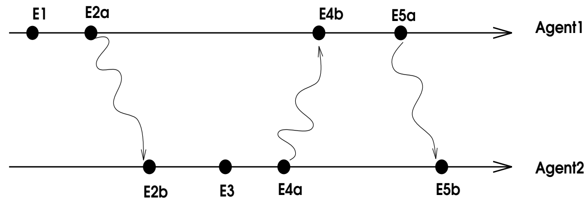


Figure 4: Event history for the RMI scenario.

### 2.4 Creation and Flow of Symbols

This section presents our model of the creation and flow of symbols in a system of agents. Following [1], we adopt the name *intensional (internal) object representation* (IOR) for these symbols.

We call a protocol which obeys the constraints below a *dynamic object naming protocol* [10].

- An agent creates a new IOR whenever it observes an object. If the agent observes the same object more than once, it generates a new IOR each time. An agent never creates the same IOR in response to more than one object.
- Except for exporting (explained below), agents never generate an IOR under any other circumstances.
- Different agents never generate the same IOR's.

With these assumptions, an IOR represents exactly one object. Thus, we can speak of the denotation of an IOR independently of the agent who generated it. This allows IOR's to be exported in messages between agents as a global name which denotes the object that caused it to be generated.

For communication purposes, we allow two operations on IOR's.

- An agent can make copies of IOR's in its knowledge base and include them in messages to other agents. The denotation of the copy is the same as the original. The creator of the copy is the same as the original. This is called *exporting*.
- An agent can assimilate an IOR contained in a received message into its own knowledge base. This is called *importing* and it does not change the creator or the denotation of the IOR.

### 2.5 Agent Reasoning

This section describes the agents' internal language and inference processes.

#### 2.5.1 Formulas

Agents can believe two kinds of formulas. These are equality formulas and beliefs about basic properties of objects. Agents do not use quantifiers or connectives. Agents also associate with each formula in its knowledge base a set of four binary flags indicating whether the formula is true, false, certain, or plausible. There are constraints on how these flags can be set. For instance, an agent's hardware will not allow it to believe that a particular formula is both true and false with certainty. Fortunately, the agent is situated in a world that is consistent. Also, the agent never confuses certain information with plausible information. Therefore, it never gets into the situation where its inference processes tell it that a formula is both true and false with certainty.

An agent will believe a formula is defeasibly true (or false) if its inference processes allow the defeasible conclusion. If there are independent lines of evidence suggesting opposite conclusions, it will believe that the formula is defeasibly both true and false. Thus an agent may have the three flags for a formula set simultaneously (i.e., true, false, plausible).

If an agent believes that a formula is plausibly true, then if the agent learns that the formula is certainly false, the true flag will be turned off and the plausible flag will be ignored.

### 2.5.2 IOR's

We classify IOR's that reside in an agent's knowledge base into three categories depending on the quality of information the IOR encodes. Below, the word "accurate" means "known with certainty to be accurate."

1. Those in which the IOR encodes complete and accurate information about the denoted object.
2. Those in which the IOR encodes some accurate but not complete information about the object. The IOR may also encode plausible information about the denoted object.
3. Those in which the IOR encodes only plausible information about the denoted object.

IOR's of the first type can be acquired only by an agent observing an object, or, by an agent receiving a complete description of an object from an agent who has complete and accurate knowledge of the object.

IOR's of the second type are acquired by receiving partial descriptions of objects that describe some basic properties with certainty. Figure 5 shows IOR's of the first and second types residing in an agent's knowledge base. The solid circles depict IOR's of the first type. The solid lines connecting the solid circles represent equality statements that are known with certainty. The absence of a solid line connecting two solid circles represent that the agent knows the objects are distinct with certainty. These certain beliefs of equality and distinctness are obtainable because of the domain assumption that no two distinct objects have exactly the same set of basic properties.

The empty circles in the figure depict IOR's of the second type. The dotted lines that connect the empty circles with other circles (either solid or empty) represent plausible equalities that are derived by referent

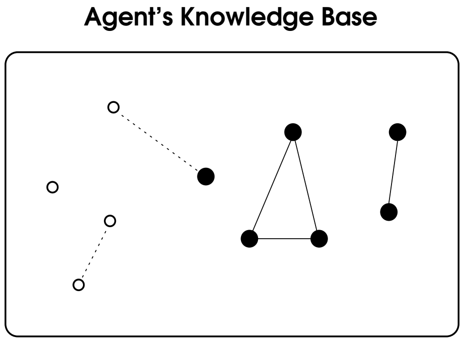


Figure 5: IOR's and equality statements in an agent's knowledge base.

identification heuristics. These equalities are never certain because it is always possible for distinct objects to spuriously have the same partial description.

We shall assume that agents ignore plausible information about basic properties when they are doing referent identification. Therefore, IOR's of the third type play no role in this model and we won't discuss them further.

### 2.5.3 Inference

Agents use two kinds of inference processes: certain and plausible.

**Certain Inference** By certain inference, we mean logically sound inference applied to premises that are known with certainty. This type of inference generates conclusions that are known with certainty. Agents use the rules of symmetry of equality, symmetry of distinctness, transitivity of equality, sound identity of indistinguishables (when complete knowledge is available), and sound distinctness. These rules are explained further in [10].

**Plausible Inference** Plausible inference is very restricted. In particular, agents have little ability to use plausible beliefs to generate new plausible beliefs.

We first discuss plausible inference in which all the premises are known with certainty. There is only one rule of inference that falls into this category and it is called *eager identity of compatibles*. This is the only referent identification heuristic that agents use and is a worst case rule which maximizes the possibility of an RMI. An agent infers default equality between two IOR's if they could denote the same object on the basis of the information the agent currently has. That is, if an agent cannot determine, with certainty, that two symbols denote distinct objects, then it assumes they denote the same object.

There are two implications of this rule. First, an agent will never believe by default that two symbols denote distinct objects. Second, an agent can believe that a particular IOR is equal by default to several other IOR's, all of which are known with certainty to be distinct.

We now turn to plausible inference in which some of the premises are plausible. There are two rules in this category. These are *symmetry of default equality* and *default substitution of equals*. Symmetry of default equality is the exact analog of ordinary symmetry of equality.

Default substitution of equals propagates basic properties that are known with certainty across default equalities to become default properties. Default properties are not allowed to propagate so any default property is at most one step away from its originator. Also, since the agent is a complete observer, a default belief about a property can only apply to an IOR that was imported. When an agent believes that an IOR is default equal to two other IOR's which are known with certainty to be distinct, the agent is able to plausibly infer both that the IOR denotes an object that has a basic property and that the object does not have the basic property.

# 3 Formal Vocabulary

This section introduces ordinary two-valued predicate calculus vocabulary that an observer could use to formally describe the domain history for the RMI scenario. Agents do not use this language; they only use the language described in the previous section.

The formalism describes specifications for the LISP implementation. We have used these specifications to prove properties of the implementation. We assume that the observer has complete knowledge of any random observation or inform events in the domain history. We also use the closed-world assumption; anything that cannot be proven true is false.

## 3.1 Predicates

All quiescent states of all agents in the domain, as well as all the agents, and all basic properties are included as abstract objects in the conceptualization. In the formulas that follow, the symbol $o$ denotes an object, $s$ denotes an IOR, and $p$ denotes a standard predicate name for a basic property. We assume all agents use the same unique standard names for basic properties.

### 3.1.1 State Predicates

There are three state predicates. These describe either the state of an object in the domain or the belief state of an agent. To recover from an RMI, an agent must retract a belief of equality. This nonmonotonicity within the agent requires us to explicitly state when an agent has a belief.

$\mathsf{hasp}(p, o)$ states that object $o$ has the basic property denoted by the property symbol $p$. Basic properties of objects do not change over time.

$\mathsf{belp}(a, t, s, p, v)$ states that at quiescent state $t$, the formula "$p(s)$" resides in the knowledge base of agent $a$ and $a$ associates the truth value $v$ with the formula.

$\mathsf{beleq}(a, t, s_1, s_2, v)$ states that at quiescent state $t$, the formula "$s_1 = s_2$" resides in the knowledge base of agent $a$ and $a$ associates the truth value $v$ with the formula.

The truth values, $v$, can be either: true with certainty ($\mathsf{t}$), false with certainty ($\mathsf{f}$), default true ($\mathsf{dt}$), or default false ($\mathsf{df}$). Default beliefs are retractable. Since the latter two predicates describe an agent's belief states at particular times, we specify persistence axioms for these formulas in section 5.1.

## 3.2 Event Predicates

There are five event predicates. The predicates all describe events which take place during the active interval following the referenced quiescent state.

$\mathsf{observe}(a, t, o, s)$ states that agent $a$ observes object $o$ during the active interval following quiescent state $t$ and generates the IOR $s$ to denote $o$.

$\mathsf{inform}(a_1, t, a_2, s, p, v)$ states that an agent $a_1$, distinct from agent $a_2$, sends a message to $a_2$ during the active interval following quiescent state $t$ of $a_1$. The message states that formula $p(s)$ has truth value $v$.

$\mathsf{informed}(a_2, t, a_1, s, p, v)$ states that $a_2$, distinct from $a_1$, receives a message from $a_1$ during the active interval following quiescent state $t$ of $a_2$. The message states that formula $p(s)$ has truth value $v$.

$\mathsf{retractp}(a, t, s, p, v)$ states that agent $a$ associated truth value $v$ with formula $p(s)$ at quiescent state $t$ but that $a$ retracted this association during the active interval following $t$.

$\mathsf{retracteq}(a, t, s_1, s_2, v)$ states that agent $a$ associated truth value $v$ with formula $s_1 = s_2$ at quiescent state $t$ but that $a$ retracted this association during the active interval following $t$.

We use the terms "inform" and "informed" instead of "send" and "receive" because the content of the message is a proposition, rather than an arbitrary data packet. Agents can inform one another of properties but they cannot inform one another of equalities.

Because of: 1) the forwarding-path theorem described in the next subsection; 2) the assumption that agents are honest; 3) the fact that agents only generate new certain beliefs by sound inference; and, 4) the basic properties of an object are persistent; it follows that an agent's certain beliefs are well justified and never have to be retracted.

## 3.3 Symbol Grounding

By using the name "George Washington," I refer to an individual that I have never seen. My reference is grounded in a causal chain [6, p. 29] to someone with firsthand knowledge of George Washington.

Theorem 1 states that if an agent, $a$, uses a symbol, then if the agent is not the creator of the symbol, there is a sequence of agents (forwarding path), starting with the creator of the symbol, and ending with $a$, such that each agent in the sequence described the symbol to the successor in the sequence. Furthermore, the events took place sequentially in time.

**Definition 1** *Define a forwarding path from agent, $a_1$, to agent, $a_n$, for some symbol, $s$, as follows. A forwarding path is a nonempty, finite sequence of agents $< a_1, \ldots, a_n >$ such that for any two consecutive agents in the sequence,* $\mathsf{described}(a_i, a_{i+1}, s)$ *holds. Furthermore, the $n-1$ described events have taken place sequentially in time.*

**Theorem 1 (Forwarding-path theorem)** *For any agent, $a$, and symbol, $s$, if $a$ uses $s$, then there is a forwarding path from the creator of $s$ to $a$.*

Given any IOR in an agent's knowledge base, there is a sequence of forwarded messages that starts with the agent who generated the IOR by observing an object and ends with the agent who has a copy of the IOR. Since the denotation of the IOR is held constant during message forwarding, the final copy of the IOR denotes the original object that caused it to be generated.

These results are proved in [10]. They depend only on the temporal properties of symbol transmission, namely, that: 1) an agent cannot use a symbol in a message until after it has acquired the symbol; and, 2) for any particular symbol, if the agent did not generate the symbol by observing an object, then the agent will not acquire the symbol until after it receives a message that uses the symbol.

#### 3.3.1 Quiescent States

The predicates below allow one to describe the temporal order between quiescent states and to say which quiescent state belongs to which agent.

$t_1 < t_2$ is true just if both $t_1$ and $t_2$ denote a quiescent state and $t_1$ temporally preceeds $t_2$.

$\mathsf{qs}(t, a)$ is true just if $t$ denotes a quiescent state of agent $a$.

For any two distinct quiescent states of an agent, one must come temporally before the other:

$$\{1\}\ \forall a \forall t_1 \forall t_2\ \mathsf{qs}(t_1, a) \wedge \mathsf{qs}(t_2, a) \wedge t_1 \neq t_2 \implies t_1 < t_2 \vee t_2 < t_1$$

No quiescent state belongs to more than one agent.

$$\{2\}\ \forall a_1 \forall a_2 \forall t\ \mathsf{qs}(t, a_1) \wedge \mathsf{qs}(t, a_2) \implies a_1 = a_2$$

The function below allows one to refer to the next quiescent state after a given quiescent state of an agent.

$\mathsf{next}$: $\{t \mid \exists a\, \mathsf{qs}(t, a)\} \rightarrow \{t \mid \exists a\, \mathsf{qs}(t, a)\}$ such that $\mathsf{next}(t)$ denotes the next quiescent state after $t$.

We abbreviate $\mathsf{next}(t)$ as $t + 1$. A quiescent state always comes before its next one. This may be expressed as:

$$\{3\}\ \forall t \forall a\ \mathsf{qs}(t, a) \implies t < t + 1$$

hasp(tractor,tr1)
¬hasp(broken,tr1)
hasp(tractor,tr2)
hasp(broken,tr2)
tr1 $\neq$ tr2

Table 1: Initial state.

## 4 System History

Table 1 shows the domain's initial state. Table 2 describes the RMI scenario using the predicates defined above and our simulation generates this output. The event labels (column 1) correspond to the labels in figure 4. The first line states that, as a result of an observation event, agent1 generates symbol "s1" to denote tractor1 during the active interval following quiescent state $t_0^{a1}$. The third column holds the justification label. In this example, we do not try to explain why agent1 observed tractor1, and just attribute it to random behavior (ran).

The notation $t_0^{a1}$ means some arbitrary quiescent state for agent1 which we happen to call $t_0$. The notation $t_1^{a1}$ abbreviates $t_0^{a1} + 1$. Finally, the notation $t_{1+m}^{a1}$ means the $m$-th successor to agent1's $t_1$ quiescent state.

A horizontal line which spans only column 2 signifies a quiescent state boundary within an agent. A horizontal line which spans all three columns temporally orders quiescent states for two different agents. Lines 2 and 3 of the table say that, because of complete and accurate observation, agent1, at quiescent state $t_1^{a1}$, believes with certainty, that the object denoted by symbol "s1" is a tractor that is not broken.

Line E3 in the table also shows an observation event. Agent2 generates symbol "s2" to denote tractor2 during the interval following quiescent state $t_1^{a2}$. At the next quiescent state, agent2 believes with certainty that the object is a broken tractor. Also during this quiescent state, the agent has the two inaccurate default beliefs that the symbols "s1" and "s2" denote the same object and that the object denoted by "s1" is broken. An asterisk indicates that the agent being described has an erroneous belief.

Finally, in the last lines of the table, agent2 has retracted both of these inaccurate beliefs as a result of being informed that tractor1 is definitely not broken. The remainder of this paper explains in detail how these retractions take place.

The justification labels are explained below.

**ran** Explanation of most of an agent's actions are beyond the scope of this paper. We say that such actions are random.

**cao** This stands for complete and accurate observation.

**mtr** This stands for message transmission.

**inf** This stands for informing.

| Label | | State or Event | Just |
|---|---|---|---|
| 1.E1 | | observe(a1,$t_0^{a1}$,tr1,"s1") | ran |
| 2. | | belp(a1,$t_1^{a1}$,"s1",tractor,t) | cao |
| 3. | | belp(a1,$t_1^{a1}$,"s1",broken,f) | cao |
| 4.E2a | | inform(a1,$t_1^{a1}$,a2,"s1",tractor,t) | ran |
| 5.E2b | | informed(a2,$t_0^{a2}$,a1,"s1",tractor,t) | mtr |
| 6. | | belp(a2,$t_1^{a2}$,"s1",tractor,t) | inf |
| 7.E3 | | observe(a2,$t_1^{a2}$,tr2,"s2") | ran |
| 8. | | belp(a2,$t_2^{a2}$,"s2",tractor,t) | cao |
| 9. | | belp(a2,$t_2^{a2}$,"s2",broken,t) | cao |
| 10. | * | beleq(a2,$t_2^{a2}$,"s1","s2",dt) | ri |
| 11. | * | belp(a2,$t_2^{a2}$,"s1",broken,dt) | sub |
| 12.E4a | | inform(a2,$t_2^{a2}$,a1,"s1",broken,dt) | ran |
| 13.E4b | | informed(a1,$t_{1+m}^{a1}$,a2,"s1",broken,dt) | mtr |
| 14.E5a | | inform(a1,$t_{2+m}^{a1}$,a2,"s1",broken,f) | cor |
| 15.E5b | | informed(a2,$t_{3+n}^{a2}$,a1,"s1",broken,f) | mtr |
| 16. | * | beleq(a2,$t_{3+n}^{a2}$,"s1","s2",dt) | prs |
| 17. | * | belp(a2,$t_{3+n}^{a2}$,"s1",broken,dt) | prs |
| 18. | | retractp(a2,$t_{3+n}^{a2}$,"s1",broken,dt) | ccp |
| 19. | | retracteq(a2,$t_{3+n}^{a2}$,"s1","s2",dt) | ini |
| 20. | | belp(a2,$t_{4+n}^{a2}$,"s1",broken,f) | inf |
| 21. | | beleq(a2,$t_{4+n}^{a2}$,"s1","s2",f) | dst |

Table 2: System history for RMI scenario.

**ri** This stands for referent identification. It is a label for the event of an agent acquiring a defeasible belief of equality.

**sub** This stands for substitution of (defeasible) equal terms.

**cor** This stands for correction. Whenever an agent receives a default message which contradicts what it already knows for certain, then it sends a correction back to the agent who sent the incorrect information.

**prs** This stands for default persistence of default belief.

**ccp** This stands for belief revision based on the consistency constraint for properties.

**ini** This stands for belief revision based on the property of indiscernable of identicals.

**dst** This stands for distinguishable implies not equal.

## 4.1 Relationships between Predicates

This section walks through the system history in table 2 and formally describes most of the justifications above.

### 4.1.1 Complete and Accurate Observation

This first group of formulas states that when an agent observes an object, it notices what properties the object does and does not have (cao). The agent acquires this information with complete certainty and accuracy.

{4} $\forall a \forall t \forall o \forall s \forall p\ \mathsf{observe}(a,t,o,s) \wedge \mathsf{hasp}(p,o) \Longrightarrow \mathsf{belp}(a,t+1,s,p,\mathsf{t})$

{5} $\forall a \forall t \forall o \forall s \forall p\ \mathsf{observe}(a,t,o,s) \wedge \neg\mathsf{hasp}(p,o) \Longrightarrow \mathsf{belp}(a,t+1,s,p,\mathsf{f})$

Formulas {4} and {5} generate lines 2 and 3, respectively, of table 2. Formula {4} also generates lines 8 and 9.

### 4.1.2 Message Transmission

The formulas below describe message transmission (mtr). They state that all messages reach their destination after they were sent and that everything received must have been sent beforehand by some agent. Formula {6} captures the assumption that messages don't get lost and formula {7} captures the assumption that messages don't generate themselves. They also provide the means to partially order quiescent states of different agents. Formula {6} generates line E2b in table 2.

{6} $\forall a_1 \forall t_1 \forall a_2 \forall s \forall p \forall v\ \mathsf{inform}(a_1,t_1,a_2,s,p,v) \Longrightarrow \exists t_2\ t_1 < t_2 \wedge \mathsf{informed}(a_2,t_2,a_1,s,p,v)$

{7} $\forall a_1 \forall t_1 \forall a_2 \forall s \forall p \forall v\ \mathsf{informed}(a_1,t_1,a_2,s,p,v) \Longrightarrow \exists t_2\ t_2 < t_1 \wedge \mathsf{inform}(a_2,t_2,a_1,s,p,v)$

There are two elements of randomness to message transmission. First, the time it takes for a message to arrive after it was sent is a random variable. Second, once a message reaches an agent, the agent has to decide to read the message. This is also a random event.

### 4.1.3 Being Informed

This next set of formulas describes the effects of being informed (inf). An agent believes what it is told unless it knows otherwise. Expression {8} generates line 6 of table 2 and expression {9} generates line 20 of table 2.

{8} $\forall a \forall t \forall a_2 \forall s \forall p\ \mathsf{informed}(a,t,a_2,s,p,\mathsf{t}) \Longrightarrow \mathsf{belp}(a_2,t+1,s,p,\mathsf{t})$

{9} $\forall a \forall t \forall a_2 \forall s \forall p\ \mathsf{informed}(a,t,a_2,s,p,\mathsf{f}) \Longrightarrow \mathsf{belp}(a_2,t+1,s,p,\mathsf{f})$

{10} $\forall a \forall t \forall a_2 \forall s \forall p\ \mathsf{informed}(a,t,a_2,s,p,\mathsf{dt}) \wedge \neg\mathsf{belp}(a_2,t,s,p,\mathsf{f}) \Longrightarrow \mathsf{belp}(a_2,t+1,s,p,\mathsf{dt})$

{11} $\forall a \forall t \forall a_2 \forall s \forall p\ \mathsf{informed}(a,t,a_2,s,p,\mathsf{df}) \wedge \neg\mathsf{belp}(a_2,t,s,p,\mathsf{t}) \Longrightarrow \mathsf{belp}(a_2,t+1,s,p,\mathsf{df})$

A necessary condition for it to be safe for an agent to believe what it is told with absolute certainty (formulas {8} and {9}) is that agents are honest when they send messages.

$\{12\}$ $\forall a_1 \forall t \forall a_2 \forall s \forall p \forall v$ $\mathsf{inform}(a_1, t, a_2, s, p, v)$
$\Longrightarrow \mathsf{belp}(a_1, t, s, p, v)$

#### 4.1.4 Referent Identification

We model agent2's referent identification inference by an eager identity of compatibles heuristic. First, we formalize subjective distinguishability. If an agent believes with certainty that the referent of one symbol has a property but the referent of another symbol does not have the property, then the agent will infer that the two symbols denote distinct objects and therefore acquire a certain belief about inequality between the symbols.

$$\{13\}\ \forall a \forall t \forall s_1 \forall s_2 \exists p\ \mathsf{belp}(a,t,s_1,p,\mathsf{t}) \wedge \mathsf{belp}(a,t,s_2,p,\mathsf{f}) \vee \mathsf{belp}(a,t,s_2,p,\mathsf{t}) \wedge \mathsf{belp}(a,t,s_1,p,\mathsf{f}) \Leftrightarrow \mathsf{beleq}(a,t,s_1,s_2,\mathsf{f})$$

With this definition, we model eager identity of compatibiles as subjective indistinguishability relative to properties that are known with certainty. This rule generates line 10 of table 2.

$$\{14\}\ \forall a \forall t \forall s_1 \forall s_2\ \mathsf{beleq}(a,t,s_1,s_2,\mathsf{dt}) \Leftrightarrow \neg \mathsf{beleq}(a,t,s_1,s_2,\mathsf{f})$$

#### 4.1.5 Substitution

This next group of formulas describes substitution of equal terms (sub) for defeasible belief. This is how certain beliefs about properties are propagated to defeasible beliefs about properties. Agents do not propagate already existing defeasible beliefs of properties across defeasible equalities. Rule {15} generates line 11 of table 2.

$$\{15\}\ \forall a \forall t \forall s_1 \forall s_2\ \mathsf{beleq}(a,t,s_1,s_2,\mathsf{dt}) \wedge \mathsf{belp}(a,t,s_1,p,\mathsf{t}) \Longrightarrow \mathsf{belp}(a,t,s_2,p,\mathsf{dt})$$

$$\{16\}\ \forall a \forall t \forall s_1 \forall s_2\ \mathsf{beleq}(a,t,s_1,s_2,\mathsf{dt}) \wedge \mathsf{belp}(a,t,s_1,p,\mathsf{f}) \Longrightarrow \mathsf{belp}(a,t,s_2,p,\mathsf{df})$$

#### 4.1.6 Correction Protocol

When an agent receives a message informing it of information that is "true (or false) by default", and this directly contradicts information that the receiving agent is certain of, then the original receiving agent informs the sending agent of this error (cor). Rule {17} generates line 14 of table 2.

$$\{17\}\ \forall a \forall t \forall a_2 \forall s \forall p\ \mathsf{informed}(a_2,t,a,s,p,\mathsf{dt}) \wedge \mathsf{belp}(a_2,t,s,p,\mathsf{f}) \Longrightarrow \mathsf{inform}(a_2,t+1,a,s,p,\mathsf{f})$$

$$\{18\}\ \forall a \forall t \forall a_2 \forall s \forall p\ \mathsf{informed}(a_2,t,a,s,p,\mathsf{df}) \wedge \mathsf{belp}(a_2,t,s,p,\mathsf{t}) \Longrightarrow \mathsf{inform}(a_2,t+1,a,s,p,\mathsf{t})$$

These formulas specify part of a communication protocol.

## 5 Persistence and Revision

A certain belief persists for the life of the agent and a default belief persists until the agent retracts it. In order to tell when a belief persists, we need to know when retractions occur.

### 5.1 Persistence Axioms

For predicates that describe mental state (i.e., belp and beleq), we must specify persistence axioms (prs). Formulas {19} – {22} say that if an agent believes a formula is true (or false) with certainty at a quiescent state, then the agent will retain the belief until its next quiescent state. Although table 2 does not show it explicitly, agent1 continues to believe that the object denoted by "s1" is a tractor for all of its quiescent states after $t_1^{a1}$.

$$\{19\}\ \forall a \forall t \forall s \forall p\ \mathsf{belp}(a,t,s,p,\mathsf{t}) \Longrightarrow \mathsf{belp}(a,t+1,s,p,\mathsf{t})$$

$$\{20\}\ \forall a \forall t \forall s \forall p\ \mathsf{belp}(a,t,s,p,\mathsf{f}) \Longrightarrow \mathsf{belp}(a,t+1,s,p,\mathsf{f})$$

$$\{21\}\ \forall a \forall t \forall s_1 \forall s_2\ \mathsf{beleq}(a,t,s_1,s_2,\mathsf{t}) \Longrightarrow \mathsf{beleq}(a,t+1,s_1,s_2,\mathsf{t})$$

$$\{22\}\ \forall a \forall t \forall s_1 \forall s_2\ \mathsf{beleq}(a,t,s_1,s_2,\mathsf{f}) \Longrightarrow \mathsf{beleq}(a,t+1,s_1,s_2,\mathsf{f})$$

Formulas {23} – {25} say that an agent's default beliefs at a particular quiescent state persist to its next quiescent state unless they are retracted. Formula {23} combined with line 11 generates line 17 provided we show that the belief in question is not retracted during any of agent2's active intervals between $t_2^{a2}$ and $t_{2+n}^{a2}$.

$$\{23\}\ \forall a \forall t \forall s \forall p\ \mathsf{belp}(a,t,s,p,\mathsf{dt}) \wedge \neg \mathsf{retractp}(a,t,s,p,\mathsf{dt}) \Longrightarrow \mathsf{belp}(a,t+1,s,p,\mathsf{dt})$$

$$\{24\}\ \forall a \forall t \forall s \forall p\ \mathsf{belp}(a,t,s,p,\mathsf{df}) \wedge \neg \mathsf{retractp}(a,t,s,p,\mathsf{df}) \Longrightarrow \mathsf{belp}(a,t+1,s,p,\mathsf{df})$$

$$\{25\}\ \forall a \forall t \forall s_1 \forall s_2\ \mathsf{beleq}(a,t,s_1,s_2,\mathsf{dt}) \wedge \neg \mathsf{retracteq}(a,t,s_1,s_2,\mathsf{dt}) \Longrightarrow \mathsf{beleq}(a,t+1,s_1,s_2,\mathsf{dt})$$

**Consistency Constraints** An agent will not believe a formula and its default negation (ccp).

$$\{26\}\ \forall a \forall t \forall p \forall s\ \neg(\mathsf{belp}(a,t,s,p,\mathsf{f}) \wedge \mathsf{belp}(a,t,s,p,\mathsf{dt}))$$

$$\{27\}\ \forall a \forall t \forall p \forall s\ \neg(\mathsf{belp}(a,t,s,p,\mathsf{t}) \wedge \mathsf{belp}(a,t,s,p,\mathsf{df}))$$

$$\{28\}\ \forall a \forall t \forall s_1 \forall s_2\ \neg(\mathsf{beleq}(a,t,s_1,s_2,\mathsf{f}) \wedge \mathsf{beleq}(a,t,s_1,s_2,\mathsf{dt}))$$

From these formulas, it can be proven that an agent must retract a default belief if it is informed, with certainty, that the opposite is true. The result is captured in the reverse direction of formulas {33} and {34}.

### 5.2 Skepticism

To show that a default belief persists, we need to characterize the conditions when a belief is not retracted. An agent will not change its existing beliefs without justification and an agent will not acquire new beliefs without a reason. Formulas {29} – {30} describe all the ways that an agent can acquire certain beliefs. They state that if an agent has a certain belief that the referent of a symbol has a property at a particular quiescent state, then at least one of the following is true: the agent observed the object and the object has the property, another agent informed it that the object has the property with certainty, it already knew that the object has the property.

$$\{29\}\ \forall a \forall t \forall s \forall p\ \ \mathsf{belp}(a, t+1, s, p, \mathsf{t}) \iff (\exists o\ \mathsf{observe}(a, t, o, s) \wedge \mathsf{hasp}(p, o)) \vee \exists a_2\ \mathsf{informed}(a, t, a_2, s, p, \mathsf{t}) \vee \mathsf{belp}(a, t, s, p, \mathsf{t})$$

$$\{30\}\ \forall a \forall t \forall s \forall p\ \ \mathsf{belp}(a, t+1, s, p, \mathsf{f}) \iff (\exists o\ \mathsf{observe}(a, t, o, s) \wedge \neg\mathsf{hasp}(p, o)) \vee \exists a_2\ \mathsf{informed}(a, t, a_2, s, p, \mathsf{f}) \vee \mathsf{belp}(a, t, s, p, \mathsf{f})$$

Formulas {19} and {20} are corollaries of formulas {29} and {30}, respectively.

Formulas {31} – {32} describe all the ways that an agent can acquire default beliefs about object properties. These formulas state that if an agent has a defeasible belief then it acquired it from defeasible substitution of equal terms or it was directly informed of the defeasible belief.

$$\{31\}\ \forall a_1 \forall t_1 \forall p \forall s_1\ \ \mathsf{belp}(a_1, t_1, p, s_1, \mathsf{dt}) \Longrightarrow (\exists t_2 \exists s_2\ t_2 < t_1 \wedge \mathsf{beleq}(a_1, t_2, s_1, s_2, \mathsf{dt}) \wedge \mathsf{belp}(a_1, t_2, s_2, p, \mathsf{t})) \vee (\exists t_2 \exists a_2\ t_2 < t_1 \wedge \mathsf{informed}(a_1, t_2, a_2, s_1, p, \mathsf{dt}))$$

$$\{32\}\ \forall a_1 \forall t_1 \forall p \forall s_1\ \ \mathsf{belp}(a_1, t_1, p, s_1, \mathsf{df}) \Longrightarrow (\exists t_2 \exists s_2\ t_2 < t_1 \wedge \mathsf{beleq}(a_1, t_2, s_1, s_2, \mathsf{dt}) \wedge \mathsf{belp}(a_1, t_2, s_2, p, \mathsf{f})) \vee (\exists t_2 \exists a_2\ t_2 < t_1 \wedge \mathsf{informed}(a_1, t_2, a_2, s_1, p, \mathsf{df}))$$

During an active interval following a quiescent state, an agent retracts a defeasible belief that an object has or does not have a property if and only if it is informed at that quiescent state that the opposite is true with certainty.

$$\{33\}\ \forall a \forall t \forall p \forall s\ \ \mathsf{retractp}(a, t, s, p, \mathsf{dt}) \iff \mathsf{belp}(a, t, s, p, \mathsf{dt}) \wedge \exists a_2\ \mathsf{informed}(a, t, a_2, s, p, \mathsf{f})$$

$$\{34\}\ \forall a \forall t \forall p \forall s\ \ \mathsf{retractp}(a, t, s, p, \mathsf{df}) \iff \mathsf{belp}(a, t, s, p, \mathsf{df}) \wedge \exists a_2\ \mathsf{informed}(a, t, a_2, s, p, \mathsf{t})$$

$$\{35\}\ \forall a \forall t \forall s_1 \forall s_2\ \ \mathsf{retracteq}(a, t, s_1, s_2, \mathsf{dt}) \iff \mathsf{beleq}(a, t, s_1, s_2, \mathsf{dt}) \wedge (\exists a_2\ \mathsf{informed}(a_2, t, a, s_1, p, \mathsf{f}) \wedge \mathsf{belp}(a, t, s_2, p, \mathsf{t}) \vee \mathsf{informed}(a_2, t, a, s_1, p, \mathsf{t}) \wedge \mathsf{belp}(a, t, s_2, p, \mathsf{f}) \vee \mathsf{informed}(a_2, t, a, s_2, p, \mathsf{f}) \wedge \mathsf{belp}(a, t, s_1, p, \mathsf{t}) \vee \mathsf{informed}(a_2, t, a, s_2, p, \mathsf{t}) \wedge \mathsf{belp}(a, t, s_1, p, \mathsf{f}))$$

#### 5.2.1 Persistence of Default Belief

Formula {36} states the conditions when a belief of default true about a property of an object persists from one quiescent state to the next. This formula is derived from formulas {23} and {33}. A belief of default true persists until the next quiescent state unless that agent has been informed, during the intervening active interval, that the belief is certainly false. A belief of default false persists under complementary conditions.

$$\{36\}\ \forall a \forall t \forall p \forall s\ \ \mathsf{belp}(a, t, s, p, \mathsf{dt}) \Rightarrow (\forall a_2\ \neg\mathsf{informed}(a, t, a_2, s, p, \mathsf{f}) \Rightarrow \mathsf{belp}(a, t+1, s, p, \mathsf{dt}))$$

$$\{37\}\ \forall a \forall t \forall p \forall s\ \ \mathsf{belp}(a, t, s, p, \mathsf{df}) \Rightarrow (\forall a_2\ \neg\mathsf{informed}(a, t, a_2, s, p, \mathsf{t}) \Rightarrow \mathsf{belp}(a, t+1, s, p, \mathsf{df}))$$

Assuming complete knowledge of the informed events in the domain, line 17 of table 2 follows from line 11 and formula {36}.

A belief of default true for an equality persists from one quiescent state to the next as long as the agent is not informed of contradicting information. Since agents do not include equality statements in messages, this information must take the following form.

$$\{38\}\ \forall a \forall t \forall s_1 \forall s_2\ \ \mathsf{beleq}(a, t, s_1, s_2, \mathsf{dt}) \Rightarrow (\forall p\ (\forall a_2\ \neg\mathsf{informed}(a, t, a_2, p, s_1, \mathsf{f}) \vee \neg\mathsf{belp}(a, t, s_2, p, \mathsf{t})) \wedge (\forall a_2\ \neg\mathsf{informed}(a, t, a_2, p, s_1, \mathsf{t}) \vee \neg\mathsf{belp}(a, t, s_2, p, \mathsf{f})) \wedge (\forall a_2\ \neg\mathsf{informed}(a, t, a_2, p, s_2, \mathsf{f}) \vee \neg\mathsf{belp}(a, t, s_1, p, \mathsf{t})) \wedge (\forall a_2\ \neg\mathsf{informed}(a, t, a_2, p, s_2, \mathsf{t}) \vee \neg\mathsf{belp}(a, t, s_1, p, \mathsf{f}))) \Longrightarrow \mathsf{beleq}(a, t+1, s_1, s_2, \mathsf{dt})$$

Line 16 of table 2 follows from line 10 and formula {38}. In summary, we can tell what defeasible beliefs persist by knowing what messages were received.

**Belief Revision** We can now derive the remaining lines of table 2. Line 18 follows from lines 15 and 17, and formula {33}. Line 19 follows from lines 15 and 16, and formula {35}. Line 21 follows from lines 9 and 20 and formula {13}.

## 6 Discussion and Related Work

### 6.1 Related Work

RMI's are a type of *referent identification failure* [5, 8]. Our active intervals correspond to Lamport's events, and our quiescent states correspond to points between two adjacent events in Lamport [9].

Chapman [2] implemented a situated agent that plays a video game in (simulated) real time. The agent can accept instructions (containing descriptions) from a human advisor. Although RMI's occur in Chapman's system, it is the responsibility of the human to diagnose them and initiate recovery.

Although RMI's cannot always be prevented, they can be made rare if agents choose descriptions carefully. Relatively short descriptions can be unique with high probability. Therefore, we have a good chance of doing accurate RMI diagnosis using either a truth-maintenance system [13, 3, 7] or a framework of reasoning under uncertainty [14, 4].

### 6.2 Answers to Symposium Questions

#### 6.2.1 Communication

The research in this paper was motivated by the question: How can agents exchange information about unanticipated objects that they discover as they explore a novel, open environment? This seems to be a basic problem. Newly discovered objects will not have preassigned names and no exchange of information can take place unless agents can signal which objects they want to talk about. Let us call this the *object naming problem*.

The paper explores the approach, inspired by analogy to natural language, of using descriptions. Using descriptions seemed like a natural way to approach this problem. It also seemed straightforward to add a description syntax and semantics to an agent communication language.

Before studying descriptions we considered two other alternatives. One was to dynamically generate canonical names for objects as they were discovered. For instance, an agent might baptize a newly discovered object with the name "the object that I saw at location x, y, z at time t." Unfortunately, this approach didn't seem to really simplify the original description-based communication problem. This is because the name used above is a description and it is not canonical (objects can change location, different agents will baptise the object with a different description depending on when and where they encounter the object, more than one object can reside at the same location).

The other approach we considered was that of refering to objects by directly pointing to them. At first this approach did not seem general because it appeared to require copresence between the agents who were exchanging the information. However, agents could be built to send raw images to each other. This would give them a shared view of a scene, regardless of their locations, and they could then point to unnamed entities by using an electronic cursor. The main disadvantage of this approach was that it required agents to possess very advanced vision systems and to be connected by high-bandwith communication links.

### 6.2.2 Real verses Software Environments

There have been recent arguments in the literature stating that a software environment could be used as the habitat for a software agent and it would present many of the scientific and engineering challenges of an actual physical environment for a physical agent. With respect to the object naming problem, I believe that software environments do not present the same challenges as physical environments. In a software environment, everything that an agent would ever want to refer to can be algorithmically given an unambiguous description. For instance, there are scripting languages that allow descriptions such as "the second character of the seventh word of the third paragraph of the file named 'guru' on the host named 'unicorn'."

Agents will not be able to use this method when operating in physical environments.

### 6.2.3 Performance

Just as in natural language communication, agents who communicate using artificial descriptions may be unable to identify or misidentify the object that the sender of the description intended to refer to. I believe that this is the key performance obstacle to make description-based communication practical. The communication scenario studied in our paper presents a difficult challenge case.

The first reason for expecting this particular example to be difficult is that an agent misidentifies a referent but has no direct evidence that such an event has occurred. How is the agent to recover from a referent misidentification that it does not know it made? The second reason for expecting this example to be difficult is that it presents a direct challenge to extreme forms of the propositional knowledge representation hypothesis, for instance, as put forth by Nilsson. Nilsson claims that symbols in a knowledge base or expert system application *denote* objects in the application's domain in exactly the same way that symbols in a mathematical first-order language *denote* objects in a model. In this scenario, using reasonable cognitive modeling assumptions we cause an agent to generate one symbol to represent more than one object. We have previously called this situation *compression* where an agent unwittingly compresses the information for two different domain objects onto one internal representation. Since the majority of traditional AI methods are based on the denotational semantics of formal logic, deciding how to apply these methods to this problem is not obvious.

### 6.2.4 Structure/Psychology

Although we used intuitions from natural language and cognitive modeling to select our approach to the object naming problem, the research in this paper represents a step away from psychology. Earlier research on this topic by the second author has more of a cognitive modeling flavor [12]. However, the approach described in [12] is more complicated than it needs to be in order to deal with this particular example. In terms of short-term limiting factors, the excess cognitive modeling baggage is not cost-effective. We believe that the system described in the present paper is the leanest system which can solve this problem. Tractability of analysis ultimately dictated our choice of agent architecture. We made the following structural decisions:

1. Basic domain properties are standardized across agents and hardwired into their perceptual systems.

   Although it is impossible to give all objects preassigned names, it seems practical to give preassigned names to properties. This design choice allows agents to build descriptions (by chosing a set of primitive properties that are descriptive of the object) and to expect the receiving agent to understand the description.

2. Object identification occurs after raw perception and does not interact with raw perception.

Whenever an agent observes an object, a symbol is generated, even if the agent has already generated a symbol for that object. This is the critical step away from cognitive modeling that guarantees that no symbol will ever represent more than one object. It allows us to speak of the denotation of a symbol and to model object identification as the action of inferring an equality between a newly acquired symbol and a pre-existing symbol in the agent's knowledge base.

Recovery from misidentification can then be modeled as belief revision where false equality statements are retracted.

3. No two agents generate the same symbols.

   This allows agents to create copies of symbols and send them to other agents without creating name conflicts in the databases of the other agents.

The main weakness of the research is that we made some very idealized assumptions. We assumed that agents are complete and accurate observers of basic properties of objects, that agents are always honest, and that there are no errors in message transmission. We certainly intend to weaken these assumptions in our future work. Critics may be tempted to warn that "researchers must discipline themselves to build physically instantiated agents in order to resist the temptation to make unrealistic assumptions." I do not think this criticism applies in our case. If we attempted to build physical agents first, we would have been trapped in a web of hardware specific obstacles which would have prevented us from even looking at the object naming problem.

#### 6.2.5 Significance

The object naming problem has received little attention in the literature. I believe this can be explained by the fact that, at least in part, AI is driven by limiting factors. Nobody's agent-program fails to perform because it can't name the object it wants to talk about.

However, in the future, we may have underwater submarine agents that explore the ocean bottom and the agent may encounter, an as yet, undiscovered species of animal. The agent's ability to tell us about the animal will then be a performance factor.

## References


[1] 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.

[2] David Chapman. *Vision, Instruction, and Action.* MIT Press, Cambridge, MA, 1991.

[3] E. Charniak, C. Riesbeck, D. McDermott, and J. Meehan. *Artificial Intelligence Programming (2nd Ed.).* Lawrence Erlbaum, Hillsdale, NJ, 1987.

[4] Eugene Charniak. Bayesian networks without tears. *AI Magazine*, 12(4):50–63, 1991.

[5] Phillip R. Cohen. The pragmatics of refering and the modality of communication. *Computational Linguistics*, 10(2):97–146, 1984.

[6] Michael Devitt. *Designation.* Columbia University Press, New York, NY, 1981.

[7] Kenneth D. Forbus and Johan de Kleer. *Building Problem Solvers.* MIT Press, Cambridge, MA, 1993.

[8] Bradley A. Goodman. Reference identification and reference identification failures. *Computational Linguistics*, 12(4):273–305, 1986.

[9] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

[10] 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.

[11] 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.

[12] Anthony S. Maida. Propositionally representing incomplete knowledge about existence. In *Proceedings of the AAAI-92 Spring Symposium at Stanford University*, March 1992.

[13] David McAllester. Reasoning utility package user's manual. Technical Report AI-TR-667, MIT, 1982. 545 Technology Square, Cambridge, Massachusetts 02139.

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