Advances in Cognitive Systems: Papers from the 2011 AAAI Fall Symposium (FS-11-01)

Reference-Related Memory Management in Intelligent Agents Emulating Humans

Marjorie McShane, Sergei Nirenburg and Stephen Beale
University of Maryland Baltimore County, Baltimore, MD, 21250, USA
{marge, sergei, sbeale}@umbc.edu

Abstract

For intelligent agents modeled to emulate people, reference resolution is memory management: when processing an object or event – whether it appears in language or in the simulated physical or cognitive experience of the agent – the agent must determine how that object or event correlates with known objects and events, and must store the new memory with semantically explicit links to related prior knowledge. This paper discusses eventualities for memory-based reference resolution and the modeling strategies used in the OntoAgent environment to permit agents to fully and automatically make reference decisions.

Introduction

In our view, the ultimate goal of agent modeling is to contribute to attaining the original goal of AI research: developing automatic intelligent agent systems that function in a society of human and artificial agents and are able to perform tasks that to date only people can perform. It is common knowledge that comprehensive agent systems must contain working models of core human capabilities of perception, reasoning and action. A representative subset of core capabilities of a general-purpose, simulated, embodied, language-enabled intelligent agent includes: perceiving events (including verbal ones) in the outside world and interpreting them; experiencing, interpreting and remembering its own mental and physical states; learning through perception, experience, communication and reasoning; and managing memory. Memory management – the focus of this paper – involves determining how each object and event encountered in the physical or mental reality of an agent is related to existing memories. In other words, when an agent does/thinks/observes something or processes language input, it must *thoughtfully incorporate* encountered objects and events into its memory of assertions, called the fact repository (FR). We refer to the full spectrum of memory-oriented reference procedures as **reference resolution**.

This paper presents a theory of reference resolution as memory management by intelligent agents along with an overview of the approach to agent modeling that will serve as a functional proof of the computer tractability of the theory. The paper does not pursue non-reference-related aspects of memory management, such as generalizing, forgetting, managing inconsistencies from different data sources, managing metacognitive information (such as trust), and so on. In addition, we will only touch upon our large body of work on linguistic aspects of reference resolution, available in McShane 2005, 2009, McShane et al., forthcoming-a and McShane et al., in preparation, among others.

The work reported here is being carried out as part of a program of research and development in the OntoAgent environment. OntoAgent supports the development of societies of human and artificial agents collaborating toward some goal (Nirenburg et al., forthcoming). To date, the most advanced application in the OntoAgent environment is Maryland Virtual Patient (MVP), a simulation and tutoring environment developed to support training cognitive decision making in clinical medicine (Figure 1). MVP is implemented as a society of agents, with one role – that of the trainee – played by a human and other roles played by artificial intelligent agents. At the core of this network is the virtual patient (VP), a knowledge-based model and simulation of a person suffering from one or more diseases. The virtual patient is a "double agent" in that it models and simulates both the physiological and the cognitive functionality of a human. Physiologically, it undergoes both normal and pathological processes in response to internal and simulated external stimuli (McShane et al. 2007, 2008). Cognitively, it is implemented as a collection of knowledge-based models of simulated human-like perception, reasoning and action

processes (see, e.g., Nirenburg et al. 2008 & forthcoming, and others of our group's related publications at http://www.trulysmartagents.org/). VP reasoning is carried out through modeling the VP's goals and plans.

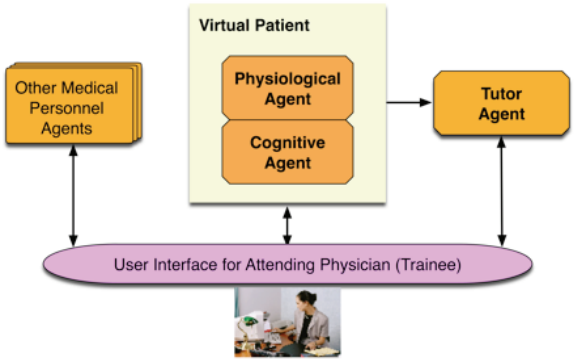

Figure 1. The network of agents in MVP.

Our approach to building a cognitive model is ideologically close to (though methodologically not identical with) the belief-desire-intention (BDI) model of agency (Bratman 1999). However, unlike the classical BDI implementations (e.g., Wooldridge 2000), our approach centrally involves comprehension and production of natural language and the incorporation of physiological simulations.

All physiological, general cognitive and language processing capabilities of all intelligent agents in OntoAgent rely on the same ontological substrate, the same organization of the fact repository (agent memory of assertions) and the same approach to knowledge representation (McShane et al., forthcoming-b). Although all OntoAgents are modeled similarly, they have different knowledge bases (ontology, lexicon, fact repository), personality traits, physiological and cognitive features, reasoning capabilities, personality traits, preferences, etc., which affords great diversity in agent behavior.

This paper presents not only a theoretically motivated classification of reference-related memory-management needs, but also a description (necessarily brief) of how those needs are fulfilled in OntoAgent and our best attempt to summarize the key technical points needed the make the latter understandable to uninitiated readers. The three sections of the paper – Memory-management Eventualities, OntoAgent Modeling, and Innovations – can essentially be read in any order, and we encourage readers to look ahead if they so choose.

Memory-Management Eventualities

All memory modification starts with perception of an object or event by an agent, with perceptive inputs including natural language input, interoception and simulated action. In this section, we describe the four main outcomes for memory modification, citing both linguistic and non-linguistic examples for each. Then in the next section we turn to the approach to cognitive modeling that permits agents to select the appropriate memory-modification outcome in a given context.

Outcome 1. The agent knows about the newly perceived object or event instance and therefore has an "anchor" in memory to which this object or event instance can link. In the realm of language processing, many traditionally delineated classes of referring expressions (REs) are covered by this reference outcome, including:

A. Pronouns, definite descriptions, etc., that have a coreferential textual sponsor, in which case the RE links to the same fact repository (FR) anchor as its textual sponsor (1). (In the examples below, the referring expression being resolved is in boldface and the sponsor is underlined)

(1) The car pulled up to the gate then **it** abruptly stopped.

Coreferential textual sponsors can be objects, events or propositions (e.g., *I trained her dog. **That** made her happy*). OntoSem methods for establishing textual coreference are discussed in McShane et al., forthcoming-a and in preparation.

B. There is no coreferential text sponsor but the entity is a proper name expected to have a corresponding anchor in every agent's FR: *John F. Kennedy*; *World War II*.

C. There is no textual sponsor but the entity is considered a "universally known" phenomenon that is expected to have a corresponding anchor in every agent's FR: *the solar system*.

D. There is a textual sponsor that is not coreferential, but whose semantic representation – stored in the FR – contains the needed coreferent. This is most readily illustrated using set-element correlations, as in (2).

(2) A couple walked in the hospital and **the man** was carrying a cane.

Here, *the man* is coreferential with one of the elements of the set introduced by *a couple*. The methods by which this coreference is established in the FR are described in the section Natural Language Processing, below.

An example of a non-linguistic context in which a processed object is directly linked to an available anchor occurs when our virtual patient makes multiple, simulated visits to the same doctor: each time it encounters the doctor in a simulation run it anchors all doctor-related information to the same FR anchor for that doctor.

Outcome 2. The agent does not know about this entity or anything related to it and must establish a new anchor for it, with no additional memory modifications. When an ontological OBJECT instance is introduced into the linguistic context as new, as by an indefinite description, the agent simply creates a new anchor for it in memory, as in (3).

(3) **Some lady** called asking for a donation.

A particularly interesting example of this class requires the agent to resolve semantic ellipsis, as in (4).

(4) Mary has a fast car and John has **one** too.

Here, the agent must interpret *one* as being in a type-coreference relationship with its sponsor, *a fast car*, and must create a new FR anchor of that type to account for the meaning of *one*.

When an ontological EVENT instance is introduced into the context as new, it can either (a) *only* require the creation of a new FR anchor, or (b) require the creation of a new FR anchor along with additional memory augmentation. The only time when it *only* creates a new FR anchor is if the event is non-agentive and is not a subevent of any script known to the agent.[1] An event that our agent would currently process in this way is *hiccup*, since hiccupping is a non-agentive act and it happens not to be part of any script in the agent's ontology. Note that whether someone's act of hiccupping is read about in text, observed through simulated vision, or experienced by the agent itself through interoception, the event will be interpreted and remembered in the same way by our intelligent agent.

Outcome 3. The agent does not know about this entity but does know about something that stands in a specific referential relationship to it; as such, the agent must create a new anchor for the entity and also represent the necessary semantic correlation between the new and related known entities. In certain language contexts, entities expressed using a definite description are in a merynomic (5) or bridging (6) relationship with previously introduced entities.[2] The use of the definite article provides a linguistic clue that this additional semantic relationship must be established in memory.

(5) I walked in the kitchen and **the window** was open.

(6) Our flight was late because **the pilot** got caught in traffic.

Specifically, when processing (5), the agent must create a new anchor in memory for WINDOW and append to it the property PART-OF-OBJECT, filled by the anchor for the given instance of KITCHEN. Similarly, when processing (6) the agent must create a new anchor for PILOT and append to it the reified property (AGENT-OF FLY-AIRPLANE (PART-OF-EVENT AIR-TRAVEL-#)), in which AIR-TRAVEL-# (instantiated by semantically analyzing *our flight*) will have the actual index of the appropriate FR instance. In both of these cases, the agent's decision about memory modification leverages linguistic heuristics (the use of a definite description) and ontological knowledge about the entities in the context.

In other cases, there are no explicit linguistic clues to suggest the need for additional memory modifications. For example, every time an agent processes an event instance, it must determine whether that event is a subevent of a known script instance (7) and/or is a plan seeking to fulfill a known goal (8).

(7) Last week, another act of piracy occurred off the shores of Somalia. Pirates **hijacked** a German ship with a crew of 50 and **stole** much of the cargo.

(8) [A doctor says to a patient] I'm wondering if your illness might be related to travel. Have you traveled anywhere lately that could have made you sick?

When processing (7), the agent must recognize that the new instances of HIJACK and STEAL are subevents of the known instance of PIRACY (introduced by the string *piracy*), and must link these new event instances to the memory of PIRACY using the property SUBEVENT-OF-SCRIPT. Similarly, when processing (8), the agent must (a) recognize and remember the doctor's goal (finding out if the patient's illness might be related to travel), (b) recognize and remember the content of the question and (c) recognize and remember that the question is the plan used to fulfill the goal. All of this information will be used when the agent is deciding how to answer the question.

All of the above examples involved linguistic input. However, our agent environment shows corresponding non-linguistic examples of this type of memory augmentation as well. For example, all of the new event instances that occur during a given simulated doctor's visit are linked via the relation PART-OF-EVENT to the FR anchor for that visit, which is initiated when the agent, in its virtual life, shows up at the doctor's office.

*Outcome 4. The agent does not know about the given event instance or have any directly related remembered instances in memory. However, the event itself suggests the script to which it belongs or the goal that it pursues, and these associations must be remembered explicitly***.** Assume, for example, that (9) is a text-initial sentence.

(9) Last week Somali pirates **hijacked** a German ship with a crew of 50.

The fact that the HIJACK event was carried out by pirates is sufficient to conclude (for reasons explained in the Ontology section below) that the hijacking was part of a PIRACY script. As a result, the memory augmentation that

[1] Agentive events have associated goals that must be remembered. Subevents of scripts must be remembered in conjunction with their scripts. See below for details.

[2] A definite description in an NP with the article *the*. Meronymy is the "part of" relation. Bridging is a non-coreferential reference relation between elements of the same script (for more on bridging see, e.g., Poesio et al. 1997).

should occur because of this input includes creating a new anchor for the HIJACK event, creating a new anchor for the implied PIRACY script that it belongs to, and linking these using the property SUBEVENT-OF-SCRIPT.

Similarly, in (10), the fact that a question is asked triggers the agent (a) to try to determine what the asker's goal was in asking the question, (b) to remember *both* the goal *and* the question *and* their relationship, and then (c) to attempt to respond to the question taking into consideration the asker's goal.

(10) [A doctor says to a patient] Have you been traveling lately?

Non-linguistic examples of this eventuality will occur when we provide our agents with simulated vision, which will permit them to "view" other agents' non-verbal actions and reason about their script-related and goal-related associations.

This concludes the overview of eventualities that can occur when an agent processes an object or event. Now we turn to the knowledge bases and modeling strategies that permit the agent to select the appropriate memory modification for a given context.

OntoAgent Modeling

Six aspects of OntoAgent that are important for understanding memory management in are discussed below in turn. In each section, the specific reference-oriented import of the module being described is highlighted in boldface.

The Ontology (Semantic Memory of Types)

The OntoAgent ontology is a formal model of the world that provides a metalanguage for describing meaning derived from any source, be it language, intelligent agent perception, intelligent agent reasoning or simulation. The metalanguage of description is unambiguous (i.e., concept names are not English words, though they look like English words), permitting automatic reasoning about language and the world to be carried out without the interference of lexical and morphosyntactic ambiguities. The ontology currently contains around 9,500 concepts, with more concept "bunching" than is found in most word nets and other hierarchies that have been called ontologies (see McShane et al. 2005 for discussion). Each object and event is described by an average of 16 properties, whose fillers can be locally defined or inherited. The ontology contains both basic (slot-facet-filler) frames and scripts, which are complex events.

Basic ontological frames support reasoning about object meronymy: e.g., an agent knows that a window can be part of a room from the knowledge (ROOM (HAS-OBJECT-AS-PART (DEFAULT WINDOW))) (cf. example (5) above).

Ontological scripts support two kinds of reference-related reasoning. First, they support reasoning about bridging relationships, which are correlations between objects and events that typically play a role in known types of situations. For example, an agent can understand the reference link in (6) using the excerpt from the AIR-TRAVEL script: (AIR-TRAVEL (HAS-EVENT-AS-PART (FLY-AIRPLANE (AGENT PILOT))). The second type of reference relation supported by ontologically recorded scripts is the SUBEVENT-OF-SCRIPT relation, as is needed to process examples (7) and (9). A *short excerpt* from the OntoAgent PIRACY script is shown below. Indices are used for cross-referencing ontological instances[3] within the script. In the full script, each ontological instance is opened up into its own frame; here we show this process only on the example of the subevent HIJACK-#1.

PIRACY-SCRIPT

AGENT	PIRATE-#1 ; a set of pirates
THEME	SHIP-#1
LOCATION	BODY-OF-WATER-#1
LEGALITY	NO
HAS-EVENT-AS-PART	HIJACK-#1, STEAL-#1, RAID-MILITARY-#1, ...

HIJACK-#1

HAS-EVENT-AS-PART	
APPROACH-#1	(THEME: BOAT-#1, DESTINATION: SHIP-#1)
SIEGE-#1	(AGENT: SET-#1, THEME: SHIP-#1)
CATCH-#1	(AGENT: SET-#1, THEME: SHIP-#1)
BOARD-#1	(AGENT: SET-#1, DESTINATION: SHIP-#1)
CONTROL-EVENT-#1	(AGENT: SET#1, THEME-SHIP#1)
ATTACK-#1	(AGENT: SET-#1, THEME: CREW-#1)
...	

If an agent encounters one or more subevents of a script, it needs to link them all to the same remembered script instance as a record of the fact that these events are all interrelated. In other words, if a report includes the fact that a group of pirates approached a ship, boarded it, captured the crew (a CONTROL-EVENT) and stole the cargo, it is essential for the agent to not only remember each event, but to remember that they were all part of a single PIRACY incident. Importantly, the agent must link all of these events to the same remembered instance of PIRACY whether or not the text included an explicit mention of piracy.

Meaning Representations

The ontological metalanguage is the language of all OntoAgent meaning representations, whose content can derive from language processing, agent thinking/reasoning/observing or simulated agent experience. The format of a meaning representation is illustrated below and described in detail in Nirenburg and

[3] These are called proto-instances in the Knowledge Machine environment: http://www.cs.utexas.edu/users/mfkb/RKF/km.html).

Raskin 2004 and McShane et al., in preparation. The example whose meaning is represented is the sentence *Pirates hijacked a ship on March 3, 2009.* Indices are used to indicate instances of ontological concepts.

HIJACK-1
  AGENT PIRATE-1
  THEME SHIP-1
  ABSOLUTE-MONTH MARCH
  ABSOLUTE-DAY 3
  ABSOLUTE-YEAR 2009
PIRATE-1
  AGENT-OF HIJACK-1
SHIP-1
  THEME-OF HIJACK-1

No matter what kind of physical or mental action an agent carries out, it renders the meaning in representations of this type, such that the format of input to memory modification functions is compatible in all cases.

The Fact Repository (Episodic Memory)

The fact repository (FR) represents agent memory of tokens – in contrast to the ontology, which represents agent knowledge of types. When initially populated (i.e., before memory merging, forgetting, etc.) the FR is very similar to the meaning representations that provide its content. The main difference is that the FR *reflects the results of reference-oriented reasoning.* For example, if the same pirates referred to in our example above were already known by our agent to have been the agent of 2 other ship hijackings, 2 kill events, and so on, the FR anchor for those pirates would list those event instances as fillers of the AGENT-OF slot. For example, the HIJACK-1 event described by the meaning representation above might be remembered in the FR as HIJACK-FR69, and the ship that was seized might be SHIP-FR203 (these instance numbers reflect the real-world situation in which this set of pirates is the 78th one encountered by our agent, these hijack events were among many other hijack events encountered by our agent, etc.).

PIRATE-FR78
  AGENT-OF HIJACK-FR45, HIJACK-FR47, HIJACK-FR69, KILL-FR120, KILL-FR-231, ...
SHIP-FR-203
  THEME-OF HIJACK-FR69
HIJACK-FR69
  AGENT PIRATE-FR78
  THEME SHIP-FR203
  ABSOLUTE-MONTH MARCH
  ABSOLUTE-DAY 3
  ABSOLUTE-YEAR 2009

The importance of the FR to the current discussion should be clear: **reference resolution, as well as all reference-related memory modifications, are memory-oriented processes that augment the contents of the FR.**

Natural Language Processing

OntoAgent represents an approach to cognitive modeling that grew out of the OntoSem approach to processing natural language (Nirenburg and Raskin 2004; McShane et al., in preparation). The OntoSem text analysis system takes as input unrestricted text and outputs meaning representations of the type illustrated above. So, before an agent tries to carry out any reasoning based on textual input, it first interprets that input – rendering it as unambiguous, ontologically-grounded meaning representations – and stores it in memory; then, all subsequent reasoning is carried out on the basis of stored memories. Language analysis relies on the ontology, the fact repository, an ontologically linked lexicon, and rule sets that cover preprocessing, syntactic analysis, semantic analysis and discourse/pragmatic analysis.

As an example of the benefits of deep semantic processing for reference resolution, let us return to example (2), repeated here:

(2) A couple walked in the hospital and **the man** was carrying a cane
[*instance coreference: the man is a member of the set introduced by 'the couple'*]

In this context, *the man* is coreferential with one of the elements of the set introduced by "a couple". The FR state after this sentence is processed is shown below:

SET-FR40 ; a couple
  ELEMENTS HUMAN-FR345, HUMAN-FR346
  AGENT-OF WALK-FR10
HUMAN-FR345
  AGENT-OF CARRY-FR4

The key to understanding this example is the fact that the OntoSem lexicon describes the appropriate sense of *couple* as a set containing two people – formally, (SET-1 (ELEMENTS HUMAN-1, HUMAN-2)). During text processing, when that lexical sense is selected as the interpretation of the word *couple,* the resulting meaning representation includes (SET-1 (ELEMENTS HUMAN-1, HUMAN-2)). When the system seeks a sponsor for *the husband*, the elements of the set are available to serve that role. It is namely the fine-grained lexical description of the meaning of "couple" that facilitates the entity linking needed for our example McShane et al. 2005).

It is noteworthy that the theory of Ontological Semantics that served as the substrate for the OntoSem system required no fundamental extensions as it migrated from its strictly language-processing origins to its broader agent modeling applications.

Although there are many aspects of reference resolution

specific to language processing (as detailed in McShane 2005 & 2009 and McShane et al. forthcoming-a & in preparation), one of the main points here is that the inventory of memory modifications that can be triggered by a new input is the same whether that input is from language, agent cognition or agent action. So, **although language inputs provide specific types of heuristics to help the agent decide what kinds of memory modifications are needed** (e.g., an indefinite description typically suggests that the entity is new to the context), **reference resolution is always carried out on interpreted semantic structures and is always, ultimately, about memory, not language.**

Goal-Oriented Agent Modeling

Agent action in OntoAgent is modeled using agenda-style control, so that agent behavior is determined by which goal (i.e., desired state) is currently at the top of its agenda and which plan (i.e., event) toward that goal is being pursued.

Each agent has a repository of goals and associated plans, the latter being ontological events. Goals are associated with triggering conditions implemented through "listener" daemons. Thus, the goal "HAVE RESPONDED TO REQUEST-INFO" (i.e., have discharged the responsibility of answering someone's question) is triggered by the agent's perception of another agent's question. The plans for this goal – at least for some of OntoAgent agents – include RESPOND-TO-INTENTION, DO-NOT-RESPOND and RESPOND-DIRECTLY, as shown in Figure 2, which is a trace of the agent's agenda in the MVP system.

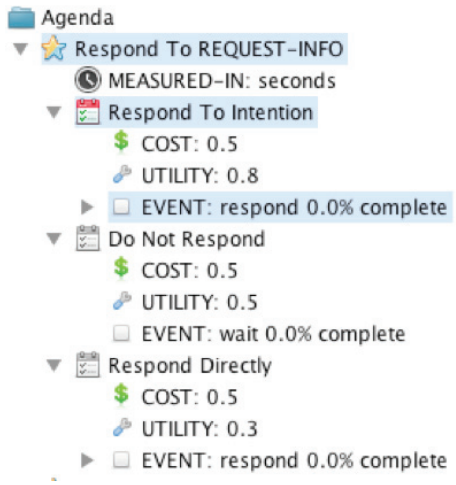

Figure 2. Excerpt from agent agenda.

When an agent interacts with another human or artificial agent, it keeps track of its own goals and associated plans and also creates a dynamically changing model of its understanding of the other agent's goals and associated plans. Accordingly, when an agent remembers its own or another's goal-directed actions, it must remember the actions *in conjunction with* the plans that they pursues: i.e., **memory modification for goal-directed actions includes the memory of the action linked to the memory of the associated goal.**

For example, it not sufficient for a virtual patient to remember that the doctor asked *Have you travelled lately?,* the patient must also remember how it interpreted the goal of that question. If it interpreted the goal of the question as simply knowing if the patient traveled at all (RESPOND-DIRECTLY), then it might answer, *Yes, I drove to New York to visit my mom*. However, if it interpreted the goal of the question as knowing if the virtual patient did any travel that might have negatively impacted its health (RESPOND-TO-INTENTION), the patient might answer *No, I haven't been anywhere that might have made me sick.* In either case, in order for the patient to have a true and complete record of this dialog turn, it must remember (a) what the doctor asked, (b) what goal it attributed to the question, (c) what it answered, (d) what goal it was pursuing when answering. The necessity of such a detailed memory is clear if the dialog takes an unexpected turn: for example, if the patient responds to what it believes to be the doctor's intention by saying, *No, I haven't been anywhere that might have made me sick*, and the doctor follows up by saying, *Oh, I wasn't asking that – I just wondered if you had had any time to relax with your family*, then the agent may realize its mistake as deriving from an incorrect initial goal attribution.

The Agent's Decision Theory

All decision making in OntoAgent – which covers everything from language processing to selecting plans and goals to managing memory – can be viewed as a type of voting environment. Building decision functions involves: (i) selecting the relevant **set of parameters** (features) and their domains and ranges; (ii) determining the relative **weights** (**importance**) of the parameters in every decision function, both in cases of complete knowledge and in cases of incomplete knowledge;[4] (iii) determining the **cost** of calculating actual values of parameters for a specific input (e.g., creating static knowledge manually or using machine learning methods; creating knowledge offline or on the fly; and so on); and (iv) determining the **confidence** in the fact that the values for parameters returned by a procedure are "true". Feature, weight, confidence and cost information

[4] Classical decision theory is not applicable to our agent modeling because it presupposes that the agent possesses *all the knowledge* necessary (or desired) for making a decision, operates with *optimum decision procedures* and is fully *rational* in a narrow terminological sense, for example in the light of rational choice theory (e.g., Becker 1976). Bounded rationality was introduced by Simon (e.g., 1991) to remove the necessity of having complete knowledge and the best algorithm, and, instead, accepting a *satisficing* decision – roughly, making do with the first decision for which utility exceeds costs even though there may be any number of better decisions available.

enrich the notion of *utility*[5], as shown in the example in Figure 1, where the utility of responding to intention is greater than the utility of responding directly, which is in turn greater than the utility of not responding at all. The parameters that go into this particular decision function include the meaning of the input request, the agent's own goals, the agent's understanding of the interlocutor's goals, the relationship between the agent and the interlocutor (if it is hostile, the agent might not want to respond), and so on.[6]

Many of the decision functions used by agents involve variables that permit different agents to have different decision-making outcomes, including in the real of memory augmentation. For example, one agent might require little evidence to corefer a newly encountered *John Smith* with a human named *John Smith* already in memory, whereas another agent might require property-based corroboration (age, profession, etc.) before assuming a coreference link. Modeling such differences across agents lends verisimilitude to the environment, since people regularly show differences of this kind in memory management, as illustrated by (11):

(11) [Husband whispers to wife at dinner party:] You realize the "scuzzy neighbor" she's talking about is the same one who dinged our car last week? [Wife:] No! Really??

An agent's decision function for resolving a newly encountered object or event involves parameters that include, non-exhaustively:

1. For language input, the available linguistic heuristics, from "surfacy" (results of preprocessing, part of speech tagging, syntactic analysis) to deep (results of semantic analysis, pragmatic analysis, speech act recognition).
2. The cost of arriving at a single, confident resolution: e.g., full semantic analysis of an input text is more costly than more "surfacy" processing (part of speech tagging, syntactic analysis).
3. The importance of arriving at a single, confident, resolution: e.g., a physician agent will definitely need to disambiguate between different patient instances, but it might be willing to leave residual referential ambiguity in its interpretation of a patient's overly long description of his recent problems at work.
4. The difference among the confidence scores of competing reference outcomes.
5. The possibility and cost (measured in user annoyance) of asking clarification questions about referential ambiguity.
6. The possibility and cost of rerunning prior processing in the hope of achieving a single best resolution result: e.g., the agent might be reparameterized to pass more candidate syntactic analyses to the semantic analyzer for evaluation, which might lead to a better overall semantic analysis and therefore to better inputs to the reference decision function.
7. The possibility and cost of postponing evaluation (implemented in OntoAgent using listeners) until more information that might aid in disambiguation becomes available.

These parameters centrally incorporate the notion that agent's job is not necessarily to resolve every instance of every referring expression to exactly one answer no matter the cost (the same, of course, applies to tasks such as word sense disambiguation, interpreting unexpected input, and the like). Managing residual ambiguities, underspecification, incongruities and all manner of approximation are a normal part of communication among people and, accordingly, must be in the repertoire of intelligent agents as well.[7] In short, agents must be able to buffer such hard edges without breaking. The job of an agent is to learn what it needs to in order to carry out the plans that will help it to fulfill its goals. **Viewing reference resolution in terms of its utility for an agent**, rather than a goal in itself, thus **leads to quite a different definition of success than is met with in the pervasive "competition style" methods of evaluation**. Rigid yardsticks for success and failure of individual subtasks fall away, leaving room for gradations of correctness and a focus on overall utility to the goal-oriented functioning of the agent.

Innovations

The innovations of the reported work, outside of the innovations of the OntoAgent environment on the whole, include the following.

1. Reference resolution is interpreted as a memory management task, not an exercise in linking up uninterpreted strings in a text (the latter encapsulates the widely-pursed in NLP "textual coreference resolution" task; cf. Hirshman and Chinchor 1998).
2. An object or event to be processed can derive from language input, agent perception, agent interoception (perception of bodily signals), or agent cognition. In

[5] Cf. Neumann and Morgenstern's (1944) notion of utility, defined as the cost effectiveness of a means to achieve a specific goal.

[6] We follow the spirit of Tversky and Kahneman's prospect theory (e.g., Kahneman and Tversky 1979) and its descendants, such as cumulative prospect theory, which augment the inventory of decision parameters for a decision (utility) function by stressing psychological influences on decision-making, such as risk aversion and utility relative to perceived utility for others.

[7] See Artstein and Poesio 2008 for a proposal about how to manage the possibility of having more than one "correct" textual coreference answer in the realm of corpus annotation.

all cases, the available eventualities for memory modification due to reference resolution are the same.

3. When the input is natural language, all types of referring expressions are treated, not only the narrow subset typically defined for the textual coreference resolution task (ibid).
4. No matter the source of the input, it is *interpreted* by the agent and rendered into a formal, ontologically grounded semantic representation that is compatible with the agent's existing memories; as such, when comparing new information to existing memories, apples are being compared with apples.
5. When comparing newly processed objects and events with the current state of memory, the outcome is not simply "match or not match"; instead, numerous other memory modifications can be required to *fully render* the reference-oriented meaning that a human would glean from the input.
6. Reference resolution is modeled as a decision function whose input parameters include not only a large inventory of features, but also a reckoning about the *utility* of the decision to the agent. Utility calculations involve the agent's *confidence* at many levels of the decision-making process, the *importance* of the decision to the agent's goals, and the *cost* of making better decisions, including the cost of gathering information to support decision-making.

Implementation of the work reported here has begun and is being carried out in parallel with other lines of work devoted to language processing, agent reasoning and agent learning (e.g., Nirenburg et al. 2007).

References

Artstein, R. and M. Poesio. 2008. Inter-coder agreement for computational linguistics (survey article). *Computational Linguistics* 34(4): 555-596.

Becker, Gary S. 1976. *The Economic Approach to Human Behavior*. University of Chicago Press.

Bratman, M. *Faces of Intention*. Cambridge University Press. 1999.

Hirshman, L. and N. Chinchor. 1998. MUC-7 coreference task definition. Version 3.0. *Proceedings of the Seventh Message Understanding Conference* (MUC-7). Applications International Corporation.

Kahneman, D. and A. Tversky. 1979. Prospect Theory: An analysis of decision under risk. *Econometrica*, XLVII, 263-291.

McShane, M. 2005. *A Theory of Ellipsis*. Oxford University Press.

McShane, M. 2009. Reference resolution challenges for an intelligent agent: The need for knowledge. *IEEE Intelligent Systems* 24(4): 47-58.

McShane, M., Nirenburg, S., and Beale, S. 2005. An NLP lexicon as a largely language independent resource. *Machine Translation* 19(2): 139-173.

McShane, M, Fantry, G, Beale, S, Nirenburg, S, Jarrell, B. 2007. Disease interaction in cognitive simulations for medical training. *Proceedings of MODSIM World Conference, Medical Track*, 2007, Virginia Beach, Sept. 11-13 2007.

McShane, M., Jarrell, B., Fantry, G., Nirenburg, S., Beale, S., and Johnson, B. 2008. Revealing the conceptual substrate of biomedical cognitive models to the wider community. *Proceedings of Medicine Meets Virtual Reality* 16, ed. Westwood, J.D., Haluck, R.S., Hoffman, H.M., Mogel, G.T., Phillips, R., Robb, R.A., Vosburgh, K.G., 281 – 286.

McShane, Marjorie and Sergei Nirenburg. Forthcoming-a. Use of Ontology, Lexicon and Fact Repository for Reference Resolution in Ontological Semantics. New *Trends of Research in Ontologies and Lexical Resources*, ed. by Alessandro Oltramari.

McShane, M., and Nirenburg, S. Forthcoming-b. A Knowledge representation language for natural language processing, simulation and reasoning. *Intl. Journal of Semantic Computing Special Issue: Semantic Knowledge Representation.*

McShane, M., Nirenburg, S., and Beale, S. In preparation. *Natural Language in Cognitive Systems*. Expected completion: late 2011.

Neumann, J. von, and O. Morgenstern. 1944. *Theory of Games and Economic Behavior*. Princeton, NJ: Princeton University Press, 1944.

Nirenburg, S., McShane, M., and Beale, S. 2008. A simulated physiological/cognitive "double agent." *Proceedings of the AAAI 2008 Fall Symposium on Biologically Inspired Cognitive Architectures.*

Nirenburg, S., T. Oates and J. English. 2007. Learning by Reading by Learning to Read. *Proceedings of the International Conference on Semantic Computing*. San Jose, August.

Nirenburg, S., McShane, M., and Beale, S. Forthcoming. A Cognitive Architecture for Simulating Bodies and Minds. *Proceedings of AMIA-2011.*

Nirenburg, S. and Raskin, V. 2004. *Ontological Semantics*. Cambridge, MA: MIT Press.

Poesio, M., R. Vieira and S. Teufel. *1997. Resolving bridging references in unrestricted text. Proceedings of the ACL Workshop on Operational Factors in Robust Anaphora Resolution*, Madrid, July, 1-6.

Simon, H. 1991. Bounded rationality and organizational learning. *Organization Science* 2 (1): 125–134

Wooldridge. Computationally grounded theories of agency. 2000. *Proceedings of ICMAS-00*. IEEE Press, pp.13-22.