

A basic agent

STEVEN VERE AND TIMOTHY BICKMORE

Lockheed AI Center, 3251 Hanover Street, Palo Alto, CA 94304, U.S.A.

Received September 18, 1989

Revision accepted February 22, 1990

A basic agent has been constructed which integrates limited natural language understanding and generation, temporal planning and reasoning, plan execution, simulated symbolic perception, episodic memory, and some general world knowledge. The agent is cast as a robot submarine operating in a two-dimensional simulated "Seaworld" about which it has only partial knowledge. It can communicate with people in a vocabulary of about 800 common English words using a medium coverage grammar. The agent maintains an episodic memory of events in its life and has a limited ability to reflect on those events. A person can make statements to the agent, ask it questions, and give it commands. In response to commands, a temporal task planner is invoked to synthesize a plan, which is then executed at an appropriate future time. A large variety of temporal references in natural language are interpreted with respect to agent time. The agent can form and retain compound future plans, and replan in response to new information or new commands. Natural language verbs are represented in a state transition semantics for compatibility with the planner. The agent is able to give terse answers to questions about its past experiences, present activities and perceptions, future intentions, and general knowledge. No other artificial intelligence artifact with this range of capabilities has previously been constructed.

Key words: agent, synthetic intelligence, episodic memory, temporal planning, natural language semantics, state transition semantics, relative adjective, reflective demon, machine consciousness, symbolic perception.

Un agent de base a été construit, lequel intègre une compréhension et une génération de langage naturel limitées, une planification et un raisonnement temporels, une exécution de plans, une perception symbolique simulée, une mémoire épisodique et une connaissance générale du monde. L'agent est représenté comme un sous-marin robot qui navigue dans un milieu simulé bidimensionnel dont il a une connaissance partielle. Il peut communiquer avec le monde grâce à un vocabulaire anglais d'environ 800 termes courants et à une grammaire de niveau intermédiaire. L'agent possède une mémoire épisodique des événements de sa vie sur lesquels il a une capacité de méditation limitée. Une personne peut faire des déclarations à l'agent, lui poser des questions et lui donner des commandements. En réponse aux commandements, un manipulateur de tâches temporel est sollicité afin d'élaborer un plan, lequel est par la suite exécuté en temps opportun. Une vaste gamme de références temporelles en langage naturel sont interprétées en fonction du temps de l'agent. L'agent peut élaborer et conserver des plans futurs composites ainsi que replanifier en réponse à de nouvelles données ou de nouveaux commandements. Les verbes du langage naturel sont représentés sous forme de sémantique de transition d'état afin d'être compatibles avec le manipulateur. L'agent est en mesure de fournir des réponses concises aux questions concernant ses expériences passées, ses activités et ses perceptions courantes, ses intentions futures et ses connaissances générales. Il n'existait auparavant aucun autre produit ayant une telle gamme de possibilités dans le domaine de l'intelligence artificielle.

Mots clés : agent, intelligence synthétique, mémoire épisodique, planification temporelle, sémantique de transition d'état, adjectif relatif, démon réflexif, conscience automatique, perception symbolique.

[Traduit par la revue]

Comput. Intell. 6, 41-60 (1990)

1. Introduction

Our conception of an agent is an integrated artificial intelligence (AI) artifact, living (at present) in a simulated environment, which can communicate in limited natural language, plan, reason, act on and perceive its environment, and reflect on its experiences. Early in 1987, at the beginning of this project, we wondered why no one had yet built an agent with these capabilities. There did not seem to be any real barriers. Research on the *components* of intelligence, e.g., planning, temporal reasoning, knowledge representation, learning, and natural language understanding and generation, has made good progress. *The underlying thesis of this work is that AI component research and computer hardware have in fact progressed to the point where it is now possible, by a resolute effort, to construct a complete integrated agent.* This paper reports on several years of work resulting in successful construction and experimentation with such an agent. The agent's name is Homer. This paper presents an account of its design and capabilities.

A full cognitive level agent is a very interesting concept in its own right. The achievement of such an AI artifact is viewed as a useful end goal, rather than as a means to justify and motivate theoretical investigations, although that may be a side effect. A full agent is also of practical significance as the cognitive, supervisory component of unmanned systems and vehicles such as a robot submarine. Unlike the many robot vehicle projects, our agent work is not focussed on low-level vehicle control or navigation and path-planning algorithms. These well-studied topics are avoided by working with a simulated world and vehicle.

The major components of the present agent are a temporal task planner and reasoner based on DEVISER V (Vere 1985a, b), a plan interpreter (executer), a world simulation with animated two-dimensional black and white graphics, an episodic memory and reflection system, a natural language parser and interpreter, a "lexipedia" combining lexical and encyclopedic world knowledge, and a tactical-level sentence generator. The encyclopedic knowledge component

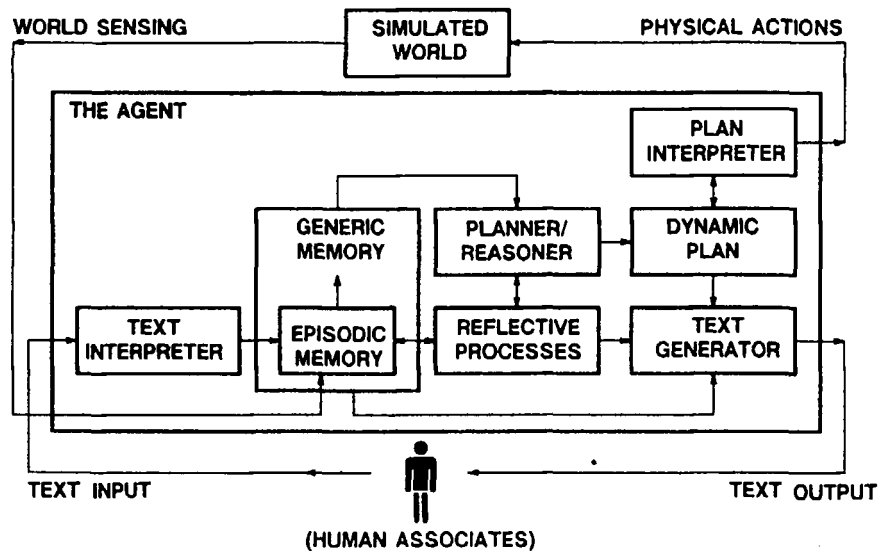


FIG. 1. Agent system block diagram.

of the agent is still quite sketchy, however. Figure 1 shows a rough block diagram of the major agent components. Because the scope of the agent is so extensive, we decided not to attempt to integrate *inductive* learning too. However, this is a natural future extension.

The parser/interpreter provides a moderate coverage of English syntax. The following features are accepted: statements, commands, and polar and constituent queries; active and passive voices; simple, perfect, progressive, and perfect-progressive aspects; past, present, future, and infinitive tenses; negated and conjunctive sentences; modal qualification; intransitive, transitive, and bitransitive verbs; recursive structures, including prepositional phrases, relative clauses, and embedded nominal clauses; count and noncount nouns; possessives; definite noun phrases; adjectives, including comparative and superlative forms; and a few adverbs. The grammar is expressed in a definite clause grammar style and is compiled into Lisp functions. Synonyms and idiomatic phrases can be defined. There is a simple pronoun reference capability. The issue of propositional phrase attachment is intentionally avoided by requiring relativizers, such as "which," for noun attachment, and by usage conventions for verb attachment. In general, for the natural language understanding component, our ambitions are limited. The vocabulary is limited, the syntax is limited, and usage limitations are assumed. One is expected to talk to Homer somewhat like one would talk to a small child or a retarded adult, using simple words and constructions, without relying on nuance or metaphor.

The natural language parser and semantic interpreter system was custom built, and operates bottom-up, giving all semantically meaningful literal interpretations of a sentence. It translates natural language sentences into the state transition representation which will be described later. Parsing and semantic evaluation are interleaved activities, so that semantically meaningless parse subtrees can be eliminated as early as possible. This interpreter runs asynchronously (pipelined) while the words of a sentence are being typed in. In this way most of the parsing and interpretation computation time is masked by the type-in time, and the results

of the interpretation are usually ready within a second after the last punctuation mark of a sentence is typed.

A natural language sentence generator (Bickmore 1988), extending the work by Danlos (1987), provides the capability to incrementally construct English sentences. Complex sentences including coordination, subordination, and all of the other syntactic categories understood by the parser can be generated. Mapping from the state transition informalism to sentence specifications is currently performed by a pattern-matcher, which uses declarative production-rule style translations. Currently, only single sentences are generated by the system in response to polar or constituent queries and in giving status reports (e.g., "I see two birds."), and multi-sentence responses are given to definitional queries (e.g., "What is a camera?"), following McKeown (1985).

The agent's present action repertoire includes the following capabilities: it can swim about within its world, pick up, hold, and drop objects, photograph objects, perceive objects, refuel, and "hear" and generate natural language utterances over a simulated radio-telephone link. This repertoire, while limited, is adequate for many of the tasks expected of an unmanned submersible, such as location and retrieval of objects, placement of sensors, underwater inspection, and surveillance. The agent can also shoot objects. This capability exists primarily to enable scenarios involving the first law of robotics ("A robot may not injure a human being, or, through inaction, allow a human being to come to harm." (Asimov 1950)). We can run such scenarios in our simulation without actuating politicians and the news media. Homer will shoot inanimate objects and animals, for example, a mine or a shark, but not people.

The agent system is set up so that the agent is always talking to one, named individual, as if the conversation were taking place over a radio telephone. In reality, the person interacting with the agent types in sentences and the agent's utterances come out simultaneously in text and through a speech synthesizer. The agent can talk to several different people sequentially, as if a telephone handset were being passed between different people in a room. A new person on the other end of the conversation is announced with the

phrase: "This is (person).", as in "This is Tim.". This device allows agent conversations about third parties and conversations about the agent's conversations with other people, or with the same person at other times.

All natural language sentences accepted by the text interpreter are translated into "INFORM" events, which are recorded in episodic memory. At the time they are entered into memory, facts are extracted from the event and stored too. A command is a special case of an inform event in which the information transmitted is that the informer has a goal and wants the informee to achieve it for him. A question is a special case of a command in which the goal of the informer is to be informed about something by the informee. There is a demon in the reflective processes which reacts to general commands by extracting the goals, processing them, and sending them to the temporal planner for plan synthesis. A separate demon reacts to questions, again extracting goals for the planner, but calling the planner in "inference" mode, i.e., using the planner strictly as an inference engine. In inference mode, the planner can use only inferences and not real actions to achieve goals or subgoals. When a temporal plan has been achieved, it is passed on to the plan interpreter for execution. In the case of a question, the answer is extracted from the inference "plan" generated by the planner and verbalized through the text generator. In the present agent, commands and questions cannot be changed or revoked once they are issued. The plan interpreter is a straightforward process which ensures that the agent will execute no action before its time, and then only when the action's predecessors in the plan have terminated. In replanning, as discussed later in detail, a plan in execution may be interrupted and sent back to the planner for revision.

For the agent's planned vocabulary, we have selected about 1900 words, which are approximately the union of Ogden's Basic English (Ogden 1934) and the 1000 most frequent English words. About 800 out of the 1900 words are presently functional. The Basic English list was designed to give broad semantic coverage, so that the vocabulary (and associated world knowledge) will "span" the space of everyday experience. However, Basic English has been found to be too Spartan for convenient communication. By adding frequent English, people can more easily generate sentences which the agent can understand, because frequent English contains all the words that we use in everyday communication, especially verbs, which are weakly represented in Basic English. According to one source, for example, just 300 words account for 95% of everyday English (Haley 1977). In summary, the frequent English component of the vocabulary ensures that proper "weight" is given to normal usage, while the Basic English component ensures that all important, but perhaps less common, concepts are accounted for.

The agent is implemented in Common Lisp and runs on Symbolics Lisp computers. It is a large system, consisting of approximately 63 000 lines of Lisp functions, demons, action and inference rules, grammar rules, and facts.

This agent project is in the general area of the "computer individual" envisioned by Nilsson (1983). The closest previous approximation to our agent is the venerable SHRDLU (Winograd 1972), which Waltz later reported to be "non-portable and nonextensible" (Waltz 1982). In retrospect, it is amazing what Winograd was able to do on a computer

with a tiny 100K memory space. Homer exceeds the scope of SHRDLU in several significant ways. Homer understands, plans, and acts in an explicit "agent time," using a sophisticated temporal planner and reasoner, has a (simulated) perception system for acquiring information about its world, and has only incomplete knowledge of that world. Actions and events can occur in the world which Homer did not plan and did not anticipate. Compound future plans can be retained and replanned in light of new information. Homer also has a relatively large general vocabulary of common English concepts and associated world knowledge.

This combination of capabilities makes it possible to experiment with many interesting new scenarios involving an interplay between perception, episodic memory, temporal reasoning, and natural language communication. The necessity for integration of components forces the issue of compatibility of representations and conventions. For example, it was found desirable to adopt an unconventional "state transition semantics" for the representation of natural language verbs, to facilitate interaction with the temporal task planner. Frame and conceptual dependency style action representations were not found to be sufficiently deep, as will be elaborated later. There is, of course, a wealth of related work pertaining to the individual components of the agent, and some of these will be referenced later when these components are discussed.

Topics presented in the remainder of this paper are the agent's simulated world and its model of that world, the symbolic perception system, a sample of scenarios which can presently be performed, "agent time" and the agent's time reference system, the location reference system for modelling spatial prepositions, the "state transition semantics" system and primitive relations for modelling the meaning of natural language sentences, the agent's episodic memory system and associated reflective demons, the dual representations of executable actions, the activation and mechanisms of replanning, and the interpretation of "extreme range" (relative) adjectives (e.g., "big") using world knowledge. Finally, we will answer some possible criticisms of our approach, and discuss difficulties, limitations, and future directions.

2. The agent's physical world

To exercise the agent's capabilities in a concrete environment, a test domain involving a simulated autonomous underwater vehicle was selected. Other candidate domains considered were space, air, and land vehicles. The Seaworld domain seemed to present the best technical opportunities for an early application of an autonomous agent for several reasons. First, the agent is likely to require healthy amounts of computing hardware. Electrical power and weight are less costly in a buoyant sea vehicle than in an aircraft or space vehicle. Second, perception seems to be the weakest link in fielding an autonomous agent. For off-road land vehicles, quick interpretation of natural scenes seems unlikely in the near future. At sea (and also in civilian space missions) the problem of perception is relatively easier. We have been told that with existing sensor technology it is even now possible to classify and identify all large objects likely to be encountered at sea (though probably not in a harbor).

Before presenting behavior scenarios and discussing the operation of the agent in detail, it is necessary to understand more about this world and the agent's model of it. Figure 2

shows an example of the world. The large main window is the stage on which agent scenarios are acted out. The agent is represented by a small submarine figure, in front of which is drawn the agent's perception sector. Everything in the world is a discrete object. Some objects are fixed, such as the island in the center and the rock to the right. Other objects have varying locations; examples are floating logs, ships, mines, and icebergs. At any time, every object has a point location, which is the two-element Cartesian coordinate of a designated point of the object, using the coordinate system shown at the edges of the window in the figure. For submarines and ships, the "location" of the object is the coordinate of the bow of the vessel. Another key parameter of an object is its orientation. Dialogs are shown in the window at the lower left. Agent time, that is, the time and date as the agent believes it to be, is explicitly displayed in the one-line window at the extreme lower left of the screen. Agent time can be accelerated or decelerated for various purposes, as will be illustrated in later scenarios.

The reinitialization of the agent is a frequent occurrence in the development and debugging of such a complex system. At reinitialization, the agent knows its own location and orientation, as well as that of the fixed objects: the island, pier, dry dock, rock, and breakwater. It does not know the location or orientation of any of the other objects in the Seaworld, except for the package, which is on the pier, and Steve and Tim, represented by human stick figures. About the other objects it has no initial knowledge. It can gain information about the other objects in two ways. It can "see" objects and it can also be told, via natural language statements, about the other objects. The agent believes everything it is told, even if this contradicts its world knowledge beliefs.

On reinitialization, the movable objects are placed randomly in the world, with a few exceptions. This allows a variety of experience for the agent on successive incarnations. It is also possible to create, rotate, move, and delete objects using the mouse. At present, except for the agent itself, only one other object moves spontaneously in the world. This is the ship called the Smirnov, which patrols back and forth across the lower portion of the screen. This serves chiefly as a moving obstacle. Movement of other objects can be accomplished by attaching the mouse to them and dragging them through desired paths.

While obstacle avoidance and path planning are not a central concern in our agent work, the agent must nevertheless behave plausibly, and not plow through solid objects. For obstacle avoidance purposes, objects are modelled as either circles or rectangles which enclose the object. For perception and some other purposes, an object is represented by a set of points, typically less than 10, on its periphery. An object is "seen" if one of these points falls within the perception sector. Also, if the agent is told to go to an object, it actually will move so that the tip of its bow comes to rest at one of these points. There is one compound object in the Seaworld, the dry dock. This is composed of the walls which are its subobjects. The dry dock exists primarily to exercise containment concepts, such as "enter," "in," "come out," etc.

Homer moves from one point to another by executing a "go" action, which takes the agent to a particular coordinate on the screen. Avoidance of obstacles is accomplished algorithmically inside of the "go," using a straightforward waypoints algorithm. The complete "go" is then achieved as

a series of movements to each waypoint. When a new object is seen, the waypoints are recomputed. This typically takes less than a second. The agent's movement then continues using the new waypoints. The task planner does not concern itself with obstacle avoidance maneuvers, except in estimating travel durations (by a call to the waypoints algorithm). If the agent was supposed to arrive somewhere at a specified time, unexpected obstacles can cause it to arrive a little late, since it has only one speed.

The symbolic perception system

The agent has a symbolic perception system which is coupled to the Seaworld simulation. This perception is a separate process that runs approximately every two seconds. If an object is within the borders of the agent's perception sector, shown on the screen at all times, that object is "seen" and a record of the perception goes into the agent's episodic memory. This perception sector is 120 degrees wide and 150 feet in depth. When the agent sees an object, it also immediately "knows" the physical parameters of the object: its type, location, orientation, dimensions, color (if any, one per object), and velocity (if any).

It is also assumed that the perception can distinguish object uniqueness. In other words, if Homer sees a log today, swims away, and comes back tomorrow and sees the same log, it will realize that this is the same log. For enhanced realism, one object can occlude another so the agent can't see it. For example, if the agent is on the right side of the breakwater looking toward the island, it will not be able to see objects on the other side of the breakwater even if they are within the perception sector. Changing some of these assumptions about perception, such as object uniqueness, would not greatly affect the operation of the agent.

As previously mentioned, nonvisual perception of common large objects at sea is now possible. Of course, on land detailed outdoor perception is still an unsolved research problem. However, in special situations, such as on a space station, it is quite possible for all objects to have machine readable labels and grids on them to bypass the vision problem. Homer's "symbolic perception" would be an excellent approximation in such circumstances.

If the agent is told false information and perception contradicts what was said, the new information from perception immediately overrides the false information. For example, suppose the agent is looking directly at Tim, and they are on the right side of the Seaworld. If someone tells the agent "Tim is at the barge," the agent will update Tim's location to one corner of the barge. However, on the next perception cycle, the perception process will record that Tim is at the old location, and this will terminate the location at the barge. Homer will "believe" that Tim really was briefly at the barge and then returned. No attempt is presently made to have Homer judge if a person has lied, based on the implausibility of objects moving so fast from one point to another.

3. Sample scenarios

In this section are presented a number of scenarios illustrating some of the agent's capabilities. These are just a few of approximately 100 agent scenarios which have been successfully conducted. It is helpful to imagine the appropriate action taking place on the agent's animated graphics

screen. Commentary on the scenarios is included in parentheses.

Scenario 1

This scenario begins with a floating log in view in front of the agent and takes place in the afternoon.

STEVE> What is in front of you?
 HOMER> A log.
 STEVE> Do you own the log?
 HOMER> No I don't.
 STEVE> The log belongs to you.
 HOMER> Oh. (The agent responds "Oh" to indicate it has been told something new. All statements are accepted as true.)
 STEVE> Cows eat grass.
 HOMER> I know. (Illustrates the response to information the agent already knows.)
 STEVE> Do you own the log now?
 HOMER> Yes I do.
 STEVE> Lose the log.
 HOMER> OK. (After attempting to form a plan and failing:) Sorry. ("Sorry" indicates a plan cannot be successfully formed.)
 STEVE> Turn away from your log.
 HOMER> OK. I'm turning 34 to 191 [degrees].
 (Agent turns so that it is facing away from the log and can no longer see it.)
 STEVE> Do you see the log now?
 HOMER> No I don't.
 (We now use the mouse to drag the log away from its original location to some distant point in the Seaworld.)
 STEVE> Turn around.
 HOMER> OK. I'm turning from 191 to 18.
 (Agent executes a 180 degree turn, "expecting" to see the log again, but doesn't.)
 HOMER> I've lost the log. It's disappeared.
 STEVE> Where is your log?
 HOMER> I don't know.
 STEVE> Go to the rock.
 HOMER> OK. (Agent plans its movement to the rock and then sets out immediately. The missing log has been placed so that the agent will see it on the way to the rock.)
 HOMER> I'm going to a rock.
 (Execution of actions is automatically announced.)
 (As soon as the log comes into view:)
 HOMER> I've found the log. (Agent continues on its way to the rock.)
 HOMER> I've reached the rock.
 (We now accelerate time to the following morning. The agent and log are still where they were the previous day.)
 STEVE> Tim wants the log that you lost yesterday afternoon.
 HOMER> Oh. OK. (The "Oh" is the response to the statement as a statement. The "OK" is an indication that the statement has been interpreted as a command. The agent will attempt to achieve goals for people, if it knows about them. The agent picks up the log again, swims around to the west side of the island, and drops the log at the feet of the Tim stickfigure.)
 STEVE> This is Tim.
 HOMER> Oh.
 TIM> Where were you last night?

HOMER> At the rock.
 TIM> How long were you at the rock?
 HOMER> 17 hours.

Scenario 2

In this scenario, a package is on the end of the pier, and the agent knows it is there. It also knows the initial location of the barge. This scenario illustrates the interpretation of a temporal reference in a command and the agent's understanding of its future intentions. Initially the agent is neither at the pier nor at the rock.

TIM> Drop the package at the barge next Saturday at 9 p.m.
 HOMER> OK.
 (Agent plans its future activities, but then takes no immediate action, because the specified time has not arrived.)
 TIM> Are you going to be at the pier next Saturday?
 HOMER> Yes. (When it picks up the package.)
 TIM> Why are you going to go to the pier?
 HOMER> I'll go to the pier in order to pick up the package.
 TIM> What time are you going to pick it up?
 HOMER> 8:56 p.m.
 TIM> Where will the package be next Sunday?
 HOMER> At the barge.
 TIM> Will the package be at the pier a week from now?
 HOMER> No.
 (The agent starts its "mission" early, using estimated travel and action times, to achieve the goal at the specified time. We now use the mouse to move the barge to a new location and then tell the agent about its new location.)
 TIM> The barge is 100 feet north of the rock. (This need only be an approximation.)
 HOMER> Oh.
 (Agent realizes that this contradicts its previous knowledge of the barge's location and replans. We now accelerate time ahead to about 8:50 p.m. the following Saturday.)
 HOMER> I'm going to the package.
 HOMER> I've reached it. (An example of a reflective comment, discussed later in the paper.)
 HOMER> I'm picking it up (Does it.)
 HOMER> I'm going to the barge. (Does it.)
 HOMER> I see two birds.
 HOMER> I've reached it.
 HOMER> I'm dropping the package at the barge. (Does it.)
 HOMER> I've left the pier. (Also a reflective comment.)

Scenario 3

This illustrates a negative command.

STEVE> Go to the island.
 HOMER> OK. (Agent goes to the nearest point on the shore of the island.)
 STEVE> Don't leave the island today.
 HOMER> OK.
 STEVE> This is Tim.
 HOMER> Oh.
 TIM> Take a picture of the Codfish.
 HOMER> OK. (then, after some delay) Sorry.
 (Agent would have to leave the island to take the picture.)

No excuses are presently offered when a plan cannot be formed.)

(We now advance agent time to the next day.)

TIM> Take a picture of the Codfish.

HOMER> OK.

(Agent generates a plan to go to a point near the Codfish and snap a picture, and then immediately executes the plan.)

Scenario 4

This illustrates compound plan retention and replanning to accommodate additional goals. By "compound plan" is meant a plan that achieves more than one set of goals, where a single set of goals is derived from a single command. The scenario takes place in the morning.

TIM> Go to the barge tomorrow evening.

HOMER> OK.

(Agent forms a plan to go from its present location to the barge, but must wait until the following evening in agent time.)

TIM> Take a picture of the pier this afternoon.

HOMER> OK.

(The second command creates goals which must be achieved before the plan already formed to go to the barge. This causes the planner to create a plan to go from the agent's present location to a point near the pier first, and a replan of the earlier plan so that the agent goes from the pier directly to the barge. Details of these mechanisms will be given later. We now advance agent time to the afternoon and the first part of the plan is triggered. Fuzzy time constraints like "this afternoon" are procrastinated. The afternoon is defined as ending at 5 p.m. In the absence of other constraints or activities, the agent will wait until just before 5 p.m. to take the picture. The agent then remains there until just before 11 p.m. the next night and then swims to the barge to complete its mission.)

Scenario 5

This illustrates an understanding of containment concepts and the consequences of the agent's own actions.

TIM> The package is in the dry dock.

HOMER> Oh. (This contradicts where the agent thought the package was.)

TIM> Please pick up the package.

HOMER> OK. (Swims into the dry dock and picks up the package.)

HOMER> I'm going into the dry dock.

HOMER> I've reached the package.

HOMER> I'm picking it up.

HOMER> I've entered the dry dock. (This realization occurs a few seconds after the actual event.)

TIM> Come out of the dry dock.

HOMER> OK. (Agent picks an arbitrary point outside of the dry dock and goes there, still holding onto the package.)

TIM> Did you remove the package from the dry dock a few minutes ago?

HOMER> Yes I did.

TIM> Did you bring the package into the dry dock today?

HOMER> No I didn't.

TIM> How many ships did you see after you left the dry dock?

HOMER> None.

Scenario 6

This illustrates a conditional command.

STEVE> If you see an animal tomorrow, photograph it.

HOMER> OK.

(We now create a bird and position it in the water near the barge. Then we advance time to the next day.)

STEVE> Go to the barge.

HOMER> OK.

(Homer goes to the barge, sees the bird, and then maneuvers to photograph it. If the bird should "fly away," the plan is aborted. Animals seen on subsequent days are not photographed.)

4. Agent time and time references

It is well to recall that in nonrelativistic situations, time is simply a scalar quantity, known or knowable to all people and agents. Assuming a fixed reference point, any point in time can then be represented by a single integer, representing the number of seconds that point is displaced from the reference. All the paraphernalia of clocks and calendars exist, to a first approximation, because the human mind cannot easily grasp and manipulate 10 digit integers. "Agent time" is such an integer, representing the number of seconds since January 1, 1900. This convention for the agent time system derives from the Symbolics time system, on which agent time is based. Under normal circumstances, the agent time parameter is increased by one each second. Changing the agent time factor from 1 accelerates or decelerates the rate of sweep of agent time relative to real time. For example, when the time factor is 10, agent time increases by 10 seconds for each second of real time. We can change the agent time factor with control keys. This time factor only changes the rate of passage of time; it does not allow time travel into the past. The date and time corresponding to agent time is continually shown in the small one-line window at the lower left of the agent interaction screen (cf. Fig. 2), with the time factor at the extreme right. For implementation reasons, agent times are actually represented internally in two varieties, "absolute agent time" and "relative agent time." The absolute time is the number of seconds since 1900, as described above. The relative time is the number of seconds since the last reinitialization. Acceleration of agent time is the most common operation and has been illustrated in several of the scenarios above. Without this capability, the testing of scenarios with large time displacements would be effectively impossible. Of course, if we accelerate time ahead at several months per minute, the agent's experiences during the acceleration are going to be very sparse.

When communicating with the agent, the time at which a sentence is received and interpreted is (to within a few seconds) the same as the time the sentence was generated (speech time). Thus a relative reference such as "in two hours" can be assumed to be relative to the present moment at the time of interpretation. This is in contrast with non-real time (delayed) communications, such as letters or quoted statements, where to interpret a relative reference such as "tomorrow," it is necessary to displace the reference relative to the time of expression, rather than the time of interpretation. In real-time communication, all explicit relative temporal references can be translated into an interval of agent

TABLE 1. Relative time references

Past	Future
Yesterday	Tomorrow
Yesterday morning	Tomorrow morning
Yesterday afternoon	Tomorrow afternoon
Yesterday evening	Tomorrow evening
Last night	Tomorrow night
Last week	Next week
Last month	Next month
Last year	Next year
Last spring	Next spring
Last summer	Next summer
Last fall	Next fall
Last winter	Next winter
Last Monday	Next Monday
Last Tuesday etc.	Next Tuesday etc.
n seconds ago	In n seconds, n seconds from now
n minutes ago	In n minutes, n minutes from now
n hours ago	In n hours, n hours from now
n days ago	In n days, n days from now
n weeks ago	In n weeks, n weeks from now
n months ago	In n months, n months from now
n years ago	In n years, n years from now
n centuries ago	In n centuries, n centuries from now

time. For example, the temporal reference "tomorrow," when uttered on June 15, 1989, translates into the interval (2822972400 2823058800) in agent time. The width of the interval is 86400 seconds, the length of a 24-hour day. Using this principle, the agent is able to understand the common relative temporal references listed in Table 1 (Azar 1984). In addition, the following "present period" references are also accepted: "today, tonight, this morning, this afternoon, this evening, this week, this month, this year, this century." Where reasonable, relative references can also be augmented with a time of day, as in "next Saturday at 3 p.m." Examples of other temporal references which are correctly understood are "at 2 a.m.," "in 1945," "on Monday (at 4:19 p.m.)," and "on October 23, 1985 (at 4:19 p.m.)."

As an example, the sentence "Tim saw a snake two weeks ago." results in the following internal representation:

```
(SCHEMA INFORM
  (AGENT STEVE)
  (PERSON HOMER)
  (TYPE (DECLARATIVE))
  (INFO (ISA .E.SNAKE/38 SNAKE)
    (MAJOR.CLAUSE (SCHEMA SENSE
      (AGENT TIM)
      (THING
        .E.SNAKE/38)
        . . .
        (TIME .T12/46)))
    (REFERENCE TIME .T12/46
      (UNIVERSAL.TIME
        (INTERVAL 2822335840 2823545440))))))
```

The integers in the interval, of course, depend on the agent time when the sentence is spoken, because the reference is relative. (Here and throughout the paper, the convention is adopted that terms beginning with a dot are variables.) An absolute reference like "at 4:19 p.m. on October 23, 1985" would of course translate into the same interval (120 seconds wide) regardless of the speech time when it is uttered. The width of the interval is commensurate with the amount of slack in the reference. The reference "last winter" generates an interval three months wide, "next year" an interval 12 months wide, etc.

The algorithmic interpretation of most of the relative time references is direct. However, the periods morning, afternoon, evening, and night are not precisely defined. In particular, when does morning begin, when does afternoon end, and when do evening and night begin and end? For these questions, we have adopted the following approximations: morning begins at 5 a.m., afternoon ends at 5 p.m., evening is from 5 p.m. to 11 p.m., and night is from 6 p.m. to 5 a.m. In a fuzzy reasoning system, these boundaries could be blurred, but at present the agent does not have any fuzzy reasoning capability. Purists are welcome to refine this system by calculating the time of sunrise and sunset for particular latitudes, days of the year, and elevations above sea level. For aviation purposes the Federal Aviation Administration has had to define "night" as beginning 30 minutes after sunset and ending at 30 minutes before sunrise. In general, when dealing with "natural concepts" for which no accepted precise definition exists, we adopt reasonable approximations for the agent to use. Natural concepts with precise definitions, such as "hour" and "aunt," are in fact extremely rare among the common English words in the agent's vocabulary. Consequently, we agree with Johnson-Laird (1982) that a concern for infallible, mathematically precise definitions is in most cases inappropriate for natural concepts and natural language understanding in general.

In the absence of explicit temporal references, past and future tenses on natural language verb phrases also provide bounds, or equivalently, an interval. Past tense is interpreted as constraining an event to the interval ($-\infty$ (now)) and future tense to ((now) ∞), where (now) is the moment of utterance in absolute agent time.

The representation of facts with limited temporal extent is based on the system of "terminator" relations in the planner (Vere 1985*b*). The underlying assumption is that facts are inherently immortal. They do not just fade away or succumb to natural causes. If a fact stops being true at a certain time, it is because some other fact terminates it. Associated with each fact is a time interval over which the fact is true. For facts that are eternally true, this interval is from minus infinity to infinity. A fact continues to be true until it is terminated by another fact, called its terminator. Explicit links are maintained in both directions between terminators and terminatees, forming temporal fact chains.

One common example is a chain of location facts for a moving object. To avoid excessive proliferation of location facts, when an object starts moving, a new location fact (LOCATION (object) IN.TRANSIT) is recorded, which terminates the location fact (LOCATION (object) (starting-location)). Typically, a few intermediate locations are recorded during motion of the agent, as when a new object is seen which was not expected. For the case of an object

moving within the agent's field of view, new locations are recorded about every two seconds, the frequency of activation of the agent's perception process.

For events, an agent time slot in relative agent time is recorded, specifying when the event began. For non-instantaneous events, a duration slot is also filled in when the event terminates. An ongoing activity is indicated by enclosing the schema in an "EXISTS" fact. For example,

```
(EXISTS (SCHEMA STATE.CHANGE
        (AGENT HOMER)
        (TIME 67794) ...))
```

When the activity ends, a (NOT (EXISTS (SCHEMA ...))) fact is recorded, and the duration of the event is recorded in a duration slot of the schema. Given the start time and duration of an activity, the time when the activity ends can, of course, also be determined, and so does not need to be explicitly stored.

The following are examples of the kinds of temporal questions the agent can answer using these representations:

```
When are you going to arrive at the island?
What time did you photograph the barge?
How long are you going to be in the dry dock?
How long did you see the iceberg?
Did you see any birds while you were going to the pier?
Did anyone enter the dry dock after you gave Tim the package?
```

5. The location reference system

As mentioned earlier, the Seaworld in which the agent lives is mapped by two Cartesian coordinates. This seems like a plausible assumption. Even now it is possible to determine locations on the surface of the earth within 50 feet using the global positioning system (GPS), and we may expect this capability to improve in the future. (A GPS ground receiver determines position by processing radio signals from geostationary satellites.) Inertial navigation units of excellent precision exist, which would allow a real agent to update its location while moving. The "location" of an object is the coordinate of a designated point on the object, such as the bow of a ship. For objects known to the agent, this coordinate and the object orientation are known. For example, if a boat identified as BOAT__9 is at coordinates (600, 100) and has an orientation of 45 degrees, the agent will be aware of the following facts: (LOCATION BOAT__9 (POINT 600 100)) and (ORIENTATION BOAT__9 (45 DEGREES)).

The agent must also be able to understand natural language location references, such as "Tim is at the pier," "the log is between you and the rock," or "the ship is south of the island." Internally, such location references are represented by a literal of the form

```
(REFERENCE LOCATION .location (.relation .subject
                                .object) .z).
```

The last term, .z, is a utility term introduced for low-level implementation reasons. As an example, "Tim is at the pier" is represented by (REFERENCE LOCATION .location (AT TIM PIER1) .z) and "the log is between you and the rock" is represented by (REFERENCE LOCATION .location (BETWEEN LOG3 (SET HOMER ROCK1)) .z). As an

example of the use of location literals, the following backward chained inference checks to see if .thing, at location .x .y, is "at" .object:

```
(OBJECT.LOCATION.REFERENCE INFERENCE
 ((LOCATION .object (POINT .x2 .y2))
  (ORIENTATION .object (.orientation2 DEGREES))
  (.margin = 2 * (OBJECT.RADIUS .thing))
  (POINT.IN.OBJECT? .x .y .object .margin. .x2 .y2
                    .orientation2))
 --->
 ((REFERENCE LOCATION (POINT .x .y)
                      (AT .thing .object) .z)))
```

The function OBJECT.RADIUS returns a value which is half of the longest dimension of an object. The function POINT.IN.OBJECT? checks to see if point .x .y lies within .margin distance of .object when it is at .x2 .y2 with orientation .orientation2. As an illustration, this rule would infer that when a person six feet tall is within six feet of a door, the person is called "at" the door.

6. State transition semantics and the primitive relations

In attempting to design an agent in which planning and natural language understanding are integrated, the agent architect immediately faces serious problems with several of the most popular approaches to natural language semantics. The problem is that they do not provide a deep representation for verbs. The approaches to be considered are "logicalized English," frames, and Schank's "conceptual dependency" system (Schank and Riesbeck 1981).

An example of "logicalized English" is the translation of the sentence "A man painted a hollow tube." into something like $\exists x \exists y \text{ man}(x) \text{ paint}(x, y), \text{ hollow}(y) \text{ tube}(y)$. For doing database retrieval and some kinds of theoretical analysis, this may be adequate. The problem is that it does not tell us what the words, such as paint, *mean*, i.e., what are the consequences of painting something, what does an agent have to do before starting to paint something, and perhaps in the extreme *how* does one paint something with a brush or spray gun. The typical logicist response to this is "Oh, you're talking about painting *axioms*." (Translation: that's not my department). But where *are* those axioms for common English verbs, and what are they going to contain? Nobody has troubled himself to write them down, and apparently logicists do not feel this is a high priority activity. This criticism is not of logic per se, but of the failure to provide the essential axioms which would explicate the meaning of the predicates. Without this information, an agent is unable to determine how to change the color of an object or how to protect it from rust. More generally, logicalized English is not a deep enough representation for the meaning of either verbs or nouns.

The frame approach to verbs, such as paint, draws nearly the same criticisms as logicalized English. A frame system may have a "paint frame" with an agent slot, which must be a person, an object slot which must be a physical object, possibly a color slot, and possibly a tool slot which defaults to brush, but may also be spray gun or spray can. The natural language interpreter tries to fill in the slots from the contents of one or more sentences about painting. In our view, frame representations for verbs are directly analogous to Lisp function calls with keyword arguments and default

values, but function calls where the body of the function is undefined.

Schank's 11 primitive actions were a first and influential attempt to do a deep analysis of verbs. However, their weaknesses and deficiencies are now well appreciated (Cercone and Schubert 1975; Wilensky 1986). They are either much too crude or totally inadequate for representing many verbs, e.g., "release" (ungrasp), "turn on" (a machine), "melt," and "stop." Also, a system of primitive actions is not directly compatible with state-based AI planners. Like logicized English, primitive actions require "inferences" to be specified for them.

The approach to action verb semantics, which we believe is required for an agent with deep understanding of natural language and which can plan and reason about information transmitted in English, is what we shall call "state transition semantics." Something like it has been suggested or used by only a small AI minority: Cercone and Schubert (1975), Allen (1987), Wilensky (1973), Waltz (1981), Neumann (1984), and McKeivitt and Wilks (1987). In a number of cases, it was directly stimulated by the requirement that the natural language system also be able to plan. The explanation is that planning is concerned with representing and reasoning about state changes. To plan to use an action, you must have a representation for what the action does to the world state, what the preconditions of the action are, and, for macro-actions, a procedural expansion of the action into subactions. Briefly, state transition semantics is a framelike description of verbs *augmented* by a description of the effect of the action on world state. The system is quite powerful, and state transition definitions of Schank's primitive actions are easily written. In writing state transition representations for the verbs in the agent's vocabulary, the previous work by Jackendoff (1983) and Miller and Johnson-Laird (1976) in the analysis of English verbs was found to be particularly relevant and helpful, although our system of primitives is different.

As an illustration of state transition semantics, consider the following definition for one sense of the verb "give":

```
(ACTION (.agent .recipient .thing)
  (CONSTRAINTS
    (ISA .agent ANIMAL)
    (ISA .recipient ANIMAL)
  )
  (ANTECEDENT
    (POSSESS .agent .thing)
    (GOAL .agent (POSSESS .recipient .thing))
  )
  (CONSEQUENT
    (POSSESS .recipient .thing)
    (NOT (POSSESS .agent .thing))
  )
)
```

The condition (GOAL .agent (POSSESS .recipient .thing)) serves to distinguish the give from involuntary transfers of possession, such as by robbery or through government confiscation. In reality, however, this goal condition represents a mental state which is not directly observable (except in one's self), but only inferable, and the conditions for making the inference are complex. Consequently, in the working definition in the agent's lexicon this condition is dropped. In the agent's taxonomy, a person is classified as one kind

of animal. Nonhuman animals (e.g., dolphins) can give and receive objects too.

Because this style of semantics is formulated in terms of world state, the question of what relations to use to describe the world becomes of prime importance. Our approach here is to use a set of primitive relations, borrowing freely from earlier research. Some of the primitives are listed below:

```
(ALIVE .organism)
(CAN .change) describes possible changes in world state
(CAUSE .x .y) for cause and effect
(COLOR .thing .color)
(CONNECTED .x .y) .x and .y are physically connected
(EMOTION.INDEX .animal .emotion .value)
(EXISTS .thing) specifies that a .thing exists or that an
  activity is ongoing
(GOAL .agent .state) the goal of an .agent is to achieve
  .state
(IN .object1 .object2) for physical containment
(ISA .thing .class) establishes a class hierarchy
(KNOWS .agent .fact) a person knows a fact
(LENGTH .object .length)
(LOCATION .thing .location) where location is a Cartesian
  coordinate
(ORIENTATION .object .angle)
(PART .part .whole)
(POSSESS .agent .thing) possession of objects by people
  or organizations
(QUANTITY.OF .thing .quantity) how many or how
  much of a thing there is
(SHAPE .object .shape)
(SUBSTANCE.OF .thing .substance) what something is
  made out of
(TEMPERATURE .object .temperature)
(TRUE.OVER .facts .t1 .t2) .facts hold over the interval
  between .t1 and .t2
(VALUE .thing .monetary.value)
(VOLUME .object .volume)
(WEIGHT .object .weight)
```

These primitive relations are analogous to a semantic alphabet, in that complex meanings can be composed from them. The use of primitive relations is controversial, because there is some loss of nuance in the decomposition process. Unfortunately, no viable alternative is at present available which is capable of integration with state-of-the-art planners. Loss of nuance can be tolerated in the agent, but inability to plan cannot.

The complete set of primitive relations is likely to be in the size range of 50 to 100. We expect this set of primitive relations to evolve and be determined empirically in the course of writing descriptions for all of the major senses of the words in the agent vocabulary. In addition to the primitive relations, there is also a need for an indefinite number of conventional slot names for structured objects. Working with a set of explicit primitives helps to avoid circular definitions.

Of course, many actions and events, especially abstract ones, cannot easily be specified *directly* in terms of state primitives. Rather, it is more reasonable to represent them in terms of lower-level actions. For example, it is possible to build a definition of the verb "loan" on the above definition of "give":

```

(ACTION (.agent .borrower .thing)
  (LABEL X
    (SCHEMA GIVE
      (AGENT .agent)
      (RECIPIENT .borrower)
      (THING .thing)
    )
  )
  (USUALLY (LABEL .use (SCHEMA .action
    (AGENT .borrower)
    (INSTRUMENT .thing)
    (CONSTRAINTS
      (AFTER X .use))))))

(BELIEVE .agent
  (LABEL .return (SCHEMA GIVE
    (AGENT .borrower)
    (RECIPIENT .agent)
    (THING .thing)
    (CONSTRAINTS
      (AFTER X .return))))))

(USUALLY .return)
)

```

Here LABEL is used to label a fact for later reference and USUALLY is the construct for specifying default information. This definition says that a loan is a kind of give where (1) the borrower usually uses the borrowed thing in some action, (2) the loaner believes the thing will be returned, and (3) it usually is returned.

Another example is definition of (one sense of) the verb "stop" as the transition from the existence to nonexistence of an activity:

```

(ACTION (.activity)
  (SCHEMA STATE.CHANGE
    (ANTECEDENT (EXISTS .activity))
    (CONSEQUENT (NOT (EXISTS .activity)))
    (CONSTRAINTS (ISA .activity ACTIVITY)))
  )
)

```

Note that the definition of "destroy" is similar to "stop" except that the thing that ceases to exist is an object, rather than an activity.

7. The episodic memory

A record of all of the agent's actions and (simulated) perceptions, as well as natural language inputs and outputs, are stored in an episodic memory (Tulving 1972; Kolodner 1984). Kolodner's system contained hand-coded representations for relatively abstract diplomatic events. In contrast, Homer's personal memories are real recorded events, and at the level of executable actions and perceptions, such as seeing an object, hearing a person make a statement, or noting the change of location of an object. From these, more abstract events such as "return" or "have a conversation with" can be inferred mechanically.

An episodic memory module is included in the agent architecture because this seems to be an essential ingredient for machine consciousness (Natsoulas 1978). (If you feel uncomfortable with this concept, you can skip to the next paragraph without loss of continuity.) The term consciousness, while fuzzy, seems to denote the knowledge by a system of its own past experience. An episodic memory recording

system seems to be one tangible, essential element of human consciousness. This interpretation is corroborated by observations of human patients with Korsakoff's psychosis. This is an anterograde amnesia in which the patient is unable to recall any personal events after onset of the condition, i.e., exhibits a dysfunction of episodic memory. Korsakoff's psychosis is commonly described as "moment-to-moment consciousness" (Cermak 1982) (i.e., these patients are viewed as having seriously impaired consciousness), indicating the close relationship between normal consciousness and functioning episodic memory. Because the short-term memory of Korsakoff amnesics is unimpaired, it is clear that consciousness and short-term memory are not the same phenomenon. There are also interesting similarities between the behavior of Korsakoff amnesics and most AI systems in their common oblivion to their own past experience. As one researcher put it (Schank 1983), how intelligent can a program be when it reads the same paragraph three times and it fails to get mad, bored, or even to notice? Glimmerings of episodic memory and consciousness in existing AI systems can be seen in the "history list" idea, originating with Teitelman's programmer's apprentice package for BBN-LISP (INTERLISP) in 1972 (Teitelman 1972).

Of course, memories alone do not "do" anything. The agent requires "reflective processes," to continually monitor and process the episodic memory and perform such functions as abstraction, forgetting of unimportant memories, and noticing loops in past behavior.

An episodic memory is also essential for realistic dialog and question answering. For example, suppose we ask the agent: "Did you return to the island?" To respond to the question properly, the agent must recall (from its episodic memory) its past locations and its traveling actions. The agent "returned" only if it was at the island at time t1, went to the island at time t2, and t1 was before t2.

Summarization is an example of a reflective process in episodic memory and can be performed primarily in a bottom-up manner, as in the work of Borhardt (1984) and Neumann (1984). Reflective "realizers" for the following common natural concepts have been implemented: "reach," "return," "leave," "enter," "pass," "find," "lose," "disappear," and repetition noticing. At present when the agent realizes that one of these events has occurred, it simply blurts out a verbal announcement, without attempting to judge whether this is likely to be interesting to anyone. And of course this realization is recorded in episodic memory, so that it remembers when it noticed the event. These realizers are a step toward more general reflective processes.

Announcements such as "I've reached the package" and "I've entered the dry dock," seen in earlier scenarios, illustrate these realizations. The pattern for "reach" is a change of location where the new location is "at" some object and the old location was not "at" that object. "Return" is a special case of "reach" in which the moving object has been "at" the reference object previously in its existence. The new location does not have to be identical to the old location. It is only necessary that both satisfy the conditions for "at." "Leave" is the inverse of "reach." "Enter" is similar to these. "Pass" is somewhat more complex. It is the change of state where a reference object lies "between" the new location and the old location of the moving object, where "between" is approximated to mean lying within the diamond-shaped region between the two object

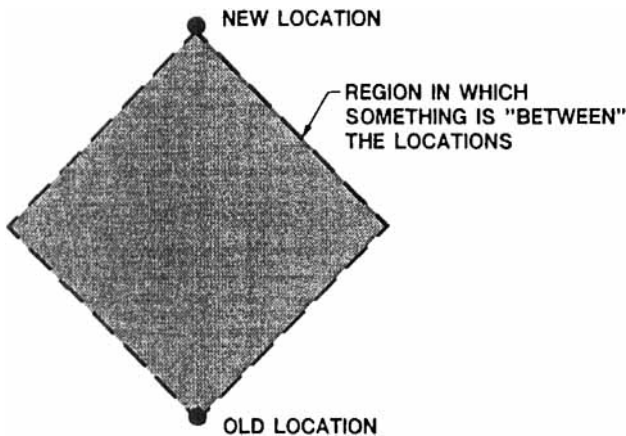


FIG. 3. Approximation to "between."

locations, as shown in Fig. 3. The reference object can, of course, be moving. "Pass" is more complex than the others because it must not fire repeatedly on a single pass trajectory, yet must reset itself so that if the moving object loops around, a new pass is correctly recognized. The agent has a repetition recognizer, so that if an action is known which has happened before, this is duly noted. For example, if the agent's activities cause it to depart from the island on two separate occasions, on the second occurrence it will say "I've left the island again." For repetitions greater than two, the number of occurrences are also verbalized: e.g., "I've left the island five times." These realizations occur for any facts or events that the agent is aware of, whether through self-knowledge, perception, or communication. These realizations are not always self-centered, as illustrated in the following scenario:

Scenario 7

(Homer is positioned near the entrance to the dry dock, and can see into it. We attach the mouse to a small boat and drag it into the dry dock.)

HOMER > A boat has reached the dry dock.

HOMER > The boat has entered the dry dock.

(Using the mouse, the boat is now dragged out of the dry dock.)

HOMER > The boat has left the dry dock.

STEVE > Turn away from the dry dock. (So Homer won't be able to see what is happening.)

HOMER > OK.

HOMER > I'm turning from 86 to 308 [degrees]. (Turns away.)

STEVE > The boat is at [coordinates] 350 300.

(This could be a lie, but the agent has no way of knowing.)

HOMER > Oh.

(After a few seconds:)

HOMER > The boat has entered the dry dock again.

Episodic memory has been discussed here as if it were a separate module from the generic memory which contains the general world knowledge. In practice, in building the agent it was found desirable to store the record of the agent's personal experiences in the same module with generic information. The motivation was that all the planning and reasoning processes would have been unnecessarily complex if every memory reference had to be screened to decide if it was episodic or generic and then routed to one of two dif-

ferent memories. Thus, looking up in memory the historical event of the death of Theodore Roosevelt or the personal event when the agent found a green box on the pier are computationally identical processes. However, in the agent episodic events *can* be distinguished from historical events, if necessary, and for the convenience of the reflective processes they are also chained together in a bidirectional list for linear access.

The agent actually has two separate episodic memories. The primary one is that just described, containing events in the agent's life. The other episodic memory is inside the planner. This planner episodic memory records detailed planner activities, such as the ordering of two actions, the expansion of a goal into a particular action, the violation of a constraint, etc. This planner episodic memory exists to guide the selective backtracking system of the planner. When the planner is reinvoked, this special control episodic memory is erased. The two episodic memories use the same functions, but of course contain different data. The planner's episodic memory is not presently accessible to the natural language system, so it is not possible to ask a question about the internal behavior of the planner, such as "When did you decide to refuel?"

8. Dual action representations

The agent maintains a dual representation of executable actions. One is the linguistic model used to determine the meaning of natural language sentences. The other is the planner's model used in generating plans. The two models are related, but distinct, with the planner's model generally being more detailed and personalized. For a command, the lexical model is used to translate the command into a set of goals for the planner, which then uses the planner's model of actions to create a plan to achieve those goals.

The most common example is the action "go." It is both a verb and an executable action. The linguistic model captures the general concept of what it means to go from one place to another. The planner model captures additional detail involved when the agent itself moves about. For example, the planner model includes an estimate of fuel consumption and travel time, both based on the distance to be travelled. There is actually a third definition of "go" (and other executable actions.) This is a function within the simulation which handles the animation and simulation parameters when the agent actually executes the action in a plan. However, this function is, strictly speaking, part of the simulation rather than the agent itself. The linguistic definition in the lexicon is

```
(WORD go
  IRREGULAR-FORMS ((WENT (FORM PAST))
                   (GONE (FORM PAST-
                          PARTICIPLE)))
  (AUXILIARY.VERB)
  (ACTION (.agent .new.containment .old.place .new.place)
  ATTRIBUTES ((TRANSITIVE NO)
              (PREPOSITIONS ((FROM .old.place)
                              (INTO .new.containment)
                              (IN .new.containment)
                              (TO .new.place))))))
; note: old.containment is not mentioned because you
; "come" out of, not "go" out of
(SCHEMA MOVE
```

```

(AGENT .agent)
(THING .agent)
(OLD.PLACE .old.place)
(NEW.PLACE .new.place)
(NEW.CONTAINMENT .new.containment)
(CONSTRAINTS (ISA .agent ANIMAL)
  (NOT (ISA .old.place .new.containment))
  (NOT (ISA .new.containment .old.place))
  (NOT (ISA .old.place .new.place))
  (NOT (ISA .new.place .old.place)))
)
)
)

```

Informally, this says that a “go” is a movement in which the agent moves itself. In turn, MOVE is defined in the lexicon as

```

(WORD move
  (CHANGE (.thing .old.place .new.place
    .old.containment .new.containment)
    ATTRIBUTES ((TRANSITIVE NO)
      (AGENT.POSSIBLE? T)
      (PREPOSITIONS ((FROM .old.place)
        (OUT_OF .old.containment)
        (TO .new.place)
        (INTO .new.containment)
        (IN .new.containment))))
    (SCHEMA STATE.CHANGE
      (CONSTRAINTS
        (ISA .thing OBJECT)
        (NOT (POINT.IN.OBJECT?
          (SECOND .new.location)
          (THIRD .new.location)
          .old.containment 3))
        (.new.location.reference = '(AT .thing
          .new.place))
        (.new.location.reference = '(IN .thing
          .new.containment))
        (.old.location.reference = '(AT .thing
          .old.place))
        (.old.location.reference = '(IN .thing
          .old.containment))
        (NEQ .new.location 'IN.TRANSIT)
        (NEQ .old.location 'IN.TRANSIT)
        (NOT (EQUAL .old.location .new.location)))
      )
      (ANTECEDENT
        (*ALREADY.INFER *START.PROTECT
          (LOCATION .thing .old.location))
        (*START.PROTECT (REFERENCE LOCATION
          .old.location
          .old.location.reference
          IN.AT))
        (REFERENCE LOCATION .new.location
          .new.location.reference .old.location)
      )
      (CONSEQUENT (LOCATION .thing .new.location))
    )
  )
)
)
)

```

This is a ground-level action, defined directly in terms of state changes. Essentially, a move is any change of location.

The location reference predicates in the antecedent interpolate between location references such as “in the dry dock” and true locations, i.e., Cartesian coordinates in the world. The annotation *ALREADY.INFER means the conditions must either be already explicitly stored as true or be inferable as true. For example, location of an object can be inferred if the object is contained in something, and the location of the container is known. The annotation *START.PROTECT means that the condition is only protected from violation until the action starts. Unless explicitly declared otherwise, all action preconditions are protected until the action terminates. The old location literal cannot be protected in the usual way because the agent’s location is sometimes updated during motion. If the old location were protected, this would be interpreted as a goal violation, erroneously triggering replanning.

When the go definition is expanded to the state transition level, we obtain

```

(SCHEMA STATE.CHANGE
  (AGENT .agent)
  (CONSTRAINTS
    (ISA .agent OBJECT)
    (NOT (POINT.IN.OBJECT?
      (SECOND .new.location)
      (THIRD .new.location)
      .old.containment 3))
    (.new.location.reference = '(AT .agent .new.place))
    (.new.location.reference = '(IN .agent
      .new.containment))
    (.old.location.reference = '(AT .agent .old.place))
    (.old.location.reference = '(IN .agent
      .old.containment))
    (NEQ .new.location 'IN.TRANSIT)
    (NEQ .old.location 'IN.TRANSIT)
    (ISA .agent ANIMAL)
    (NOT (ISA .old.place .new.containment))
    (NOT (ISA .new.containment .old.place))
    (NOT (ISA .old.place .new.place))
    (NOT (ISA .new.place .old.place))
  )
  (ANTECEDENT
    (*ALREADY.INFER *START.PROTECT
      (LOCATION .agent .old.location))
    (*START.PROTECT (REFERENCE LOCATION
      .old.location
      .old.location.reference
      IN.AT))
    (REFERENCE LOCATION .new.location
      .new.location.reference .old.location)
  )
  (CONSEQUENT (LOCATION .agent .new.location))
)

```

For planning purposes, a “go” is in fact modelled as consisting of two components, a “start.go,” which marks the beginning of the “go,” and a “go” which models most of the changes associated with the agent moving from one place to another.

```

(START.GO ACTION
  ((FUEL.LEVEL .agent .fuel .old.fuel.level)
  (*ALREADY.INFER (LOCATION .agent
    .old.location))

```

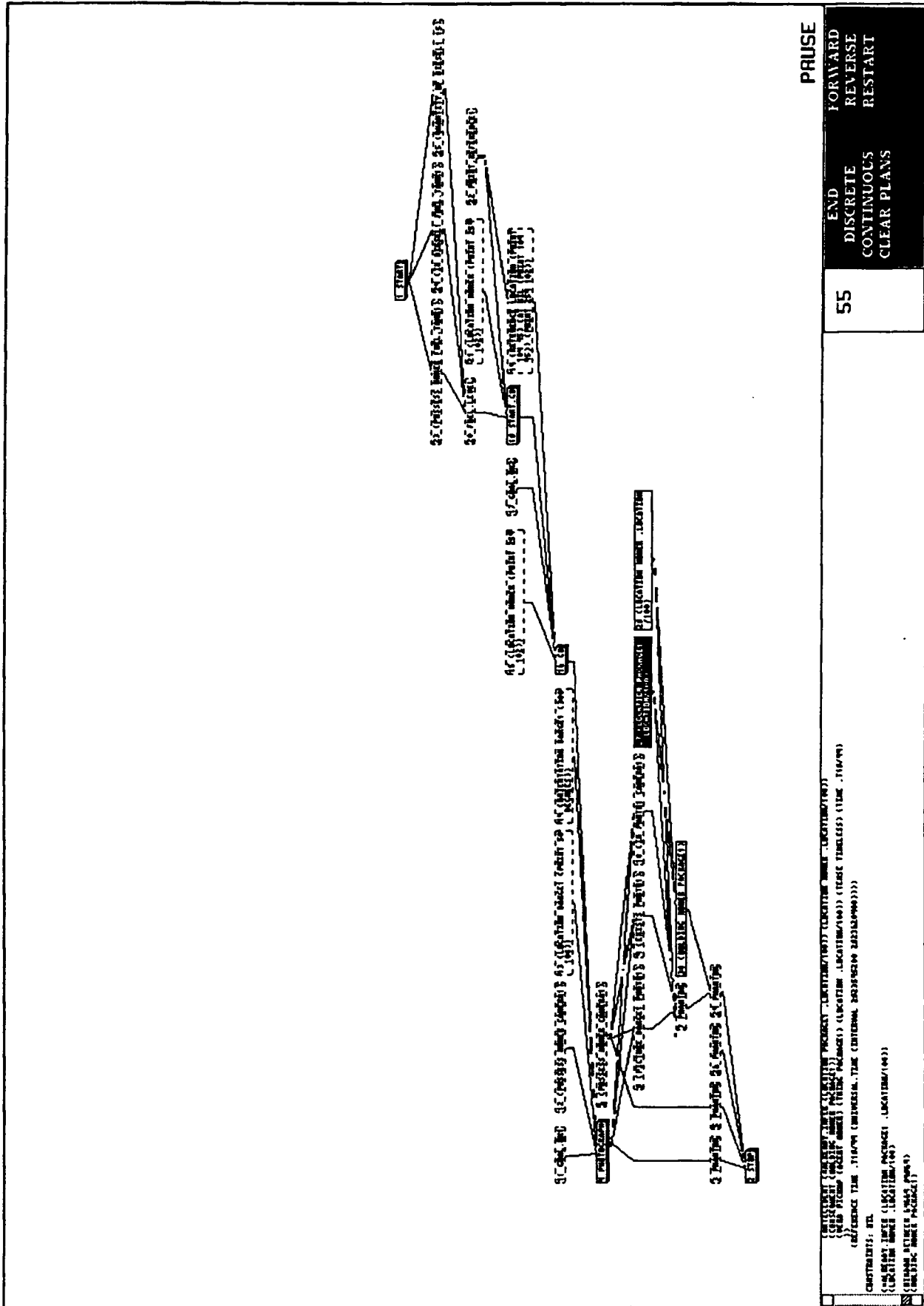


FIG. 4. Partial plan, with goals derived from "Pick up the package."

```

(*MUST.EXPAND (REFERENCE LOCATION
               .new.location
               .new.location.reference
               .old.location))
(REAL.LOCATION? .old.location)
; DON'T USE AN "IN"
REFERENCE IF YOU ARE
FAR AWAY
(OR (NEQ 'IN (FIRST .new.location.reference))
    (< (DISTANCE2 .old.location .new.location)
        (2 * (OBJECT.RADIUS (THIRD
                              .new.location.reference))))))
(.duration = (TRAVEL.DURATION .agent
              .old.location .new.location))
(.new.fuel.level = (.old.fuel.level -
                   *AGENT.FUEL.
                   CONSUMPTION.RATE*
                   * .duration))
(> .old.fuel.level 0)
(OR (> .new.fuel.level 3)
    (AND (ISA1 (THIRD .new.location.reference)
              'FUEL.PUMP)
         (> = .new.fuel.level 0))))
--->
((GO.IN.PROGRESS .agent .old.location .token
 .new.location .duration .new.fuel.level .fuel
 .new.location.reference)
 (LOCATION .agent IN.TRANSIT))
(GO ACTION
 ((ISA .agent PERSON)
 (.goal = (ACTION.GOAL))
 (*CONSECUTIVE *MUST.EXPAND
 (GOAL .agent .goal))
 (*GOAL (LOCATION .agent .new.location))
 (.token = (VERE.GENSYM))
 (*BOUND (GO.IN.PROGRESS .agent .old.location
 .token .new.location .duration .new.fuel.level
 .fuel .new.location.reference))
 (.new.containment = (EXTRACT.CONTAINMENT
 .new.location.reference))
 (.new.place = (EXTRACT.PLACE
 .new.location.reference))
 ; This has to be here as well as in start.go, so that if
 ; .new.location.reference changes location, this action
 ; is also a client, and replanning will be triggered.
 (REFERENCE LOCATION .new.location
 .new.location.reference .old.location))
--->
((LOCATION .agent .new.location)
 (QUANTITY.OF .fuel .new.fuel.level))
(DURATION .duration)
(PREFERRED.FOR (LOCATION .agent .new.location)))

```

The `start.go` component initiates the action and changes the location of the agent to `IN.TRANSIT`. The `GO.IN.PROGRESS` literal is essentially a parameter passing device between the two components. The condition `(GOAL .agent .goal)` is a multiple agent feature which ensures that the agent executing the goal is properly motivated. For example, Homer may have the goal of Tim being at a new location. Without the `GOAL` condition, the planner would simply backchain into a "go" action by Tim to the new place. This

would be wrong because Homer can't execute the action of Tim going somewhere. Only Tim can do that. Even assuming that Tim is cooperative, he can't be expected to read Homer's mind. What is in fact required is for Homer to plan a "request" action, which has the effect of transforming `(GOAL HOMER .x)` into `(GOAL TIM .x)`. Having transferred the goal from Homer to Tim, it is then reasonable to assume that Tim will carry out the go action. Of course, Tim may refuse the request. At present we do not allow actions to have unpredictable consequences. This leads to the general problem of synthesizing conditional plans, which is beyond the capability of the present planner. However, a limited capability could in fact be achieved by only a minor extension, whereby the undesired outcome would trigger replanning in the same way as new information from perception or communication presently triggers replanning. A reflective process would need to monitor this, and super-vene to avoid futile repetitions of unsuccessful behavior.

The `FUEL.LEVEL` precondition backchains into an inference which can lead to a refueling action being planned if the present fuel level is too low. This feature is not particularly sophisticated.

Many actions at the linguistic level, such as "take," are not directly executable by the agent. Take (in the sense of "take a thing to a place") is a state change in which the agent and another object both change locations together, and the agent is the cause of those changes. This contrasts with "send," in which the agent causes an object to change location, but does not end up at the same place as the object. To achieve the effect of a take, the agent must "pick up" the thing and then move to the desired place. Picking up the object has the effect of connecting the agent and the object. The planner then explicitly reasons that when two objects are connected and one moves to a new location, the other one does too. Some linguistic concepts are totally unexecutable, even by a combination of simpler actions. For example, the agent could not form a plan to "lose" an object, as illustrated in scenario 1. It could drop an object somewhere and then swim away, but it would continue to believe that the object was still where it was, rather than at an unknown location, as required by the definition of "lose".

Most verb schemas are expanded down to the state transition level for sentence interpretation and storage in memory. However, a few special "basic schemas" are left unexpanded. Examples are `INFORM`, `SENSE`, and `BUY`. The reason is efficiency in question answering. To explain this further is not worth the lengthy digression that would be required. Even for these basic schemas, upon storage the antecedents and consequences are also derived and stored. For example, if Steve informs Homer that fact P is true, the agent records that at the time of the inform event, Steve already knew P.

9. Replanning

This section considers replanning in more detail. Scenario 2 involved replanning induced by new information. Scenario 4 involved replanning to accommodate additional goals. A good planner protects achieved goals by establishing and monitoring goal protection conditions (Vere 1987) in a plan. Not all consequences of an action are protected because many are just side effects with no real purpose. It is the viola-

tion of a protected condition that leads to replanning, which takes place in three steps:

1. the detection of the violation of a goal protection condition in a plan;
2. the selective demolition of the part of the plan which depended on the violated condition, as described in Vere (1985a), which results in the recreation of goals to be achieved; and
3. The (re)planning to achieve those recreated goals, using procedures that are identical to those that generated the original plan. Because the world state is different this second time around, the new plan will be different from the old plan.

To illustrate these steps, we will consider a somewhat simpler scenario than scenario 2.

Scenario 8

STEVE > Take a picture of the barge today at 4 p.m.
 HOMER > OK. (Forms plan and waits.)
 STEVE > Pick up the package tomorrow afternoon.
 HOMER > OK.

Just after the second command is given, the planner state is as shown in Fig. 4. The boxes labelled 27, 28, and 30 are goal nodes derived from the second command. The rest of the diagram is the plan for carrying out the first command. The constraint "tomorrow afternoon" results in a time-interval constraint being placed on the package phantom node 29, to which goal node 30 is consecutive. The finished plan is shown in Fig. 5. The executable actions consist of a "start.go," a "go," and a "pickup." The start.go changes the location of the agent from its old location to "IN.TRANSIT." The go action actually models the movement of the agent from one point to another, and accounts for travel time and fuel consumption. Note that the fact (LOCATION PACKAGE1 (POINT 626 138)) is a precondition for several of the actions. This fact is known to the agent at reinitialization. Goal protection relations are not explicitly shown in the figure, but one consequence of the generation of the plan is to protect the package location fact between the start node and the action nodes 28, 30, and 32.

We now tell the agent that the package is somewhere else.

STEVE > The package is 150 feet east of the rock.
 HOMER > Oh.

From this statement it is inferred that the location of the package is now 952 212, and a new fact is stored: (LOCATION PACKAGE1 (POINT 952 212)). Because location is a function, this is recognized as contradicting and terminating the previous location fact. Whenever a fact is terminated, a check is made to see if the terminated fact was involved in a protection relation with any plan nodes. If it was, those plan nodes are "demoted" back to goals. This demotion process is recursive, so that potentially large sections of the existing plan may be reduced back to the original goals. In this particular case, all of the lower half of the plan is demolished, leaving the plan in the same state as in Fig. 4. As mentioned above, details of the demotion process have been previously published. Here the operation is the same, and only the situation which triggers the protection violation is new. With demolition complete, the planner is reinvoked. When planning is complete, a new plan has been

created that is structurally identical to the old one, except that the new location fact is incorporated, affecting the travel duration and fuel consumption parameters of the plan.

In this case we informed the agent of the new location while the agent was idle, biding its time until the arrival of tomorrow afternoon. A similar effect would result if the agent were informed, either in a statement or by seeing the package in a new location, while it was carrying out the plan. The source of the new information and the time of acquisition are not material, only the fact that it violates a protection relation of the plan. Plan demolition will take place as before, and if a demoted node is currently in execution, it is aborted and the present agent state (e.g., present location, orientation, and fuel level) is recorded. The new plan is generated, and then executed at the appropriate time.

10. Interpretation of extreme range adjectives using world knowledge

One part of everyday world knowledge is knowing normal parameter ranges for common objects. The agent vocabulary contains the names of about 250 common objects and substances. Examples are apple, chair, house, ice, milk, rat, truck, camera, baby, clock, mountain, chin, key, moon, gasoline, and jewel. We believe an agent should have a rough idea of the typical size, color, shape, area, composition, weight, density, speed, monetary value, temperature, and age of those objects and substances, where these parameters are appropriate. This is in accordance with the principle "don't be stupid" (Kibler and Morris 1981). We can make progress toward intelligent behavior simply by avoiding gross ignorance about the world. How big are apples? They aren't ten miles in diameter or 0.001 inches in diameter. They typically have a rather narrow range. However, it is always possible to haggle over *exactly* how big apples can be. What about that apple the size of a pumpkin at the state fair last year? We have adopted the following guideline: *try to specify a range that cover 99% of all cases*. If an agent can be right 99% of the time, it will be doing extremely well. For example, the the lexipedia contains the following information: (DIAMETER APPLE (INTERVAL 1 5) INCHES). We did not conduct any statistical surveys to get these numbers; we just put down something reasonable. The average parameter value is assumed to be in the middle of the specified interval. If that is not reasonable, we specify the average value in a separate fact. For apples, this is not necessary. The information provided about the common objects is not regarded as part of a mathematical definition. Consequently, if the agent is told about an apple six inches in diameter, it does not reject the information, but merely accepts the apple as unusually large.

This brings us to one important use for these parameter ranges: the interpretation of "extreme range" adjectives, such as "big," "old," "hot," "slow," "tall," "narrow," and "expensive." What meaning should an agent attach to the phrase "a large apple?" If it doesn't know how big apples usually are, it can only note that this apple is bigger than average. If it knows that the average apple is about three inches in diameter, then it can derive more information from being told that one is large. Perhaps the agent would like to pass this apple through a hole two inches in diameter. Will that be possible? Probably not. What about a small apple? Yes, that's worth a try. Or suppose the agent

learns that a large apple is floating in the path of a cruise ship. Should Homer radio a warning, in accordance with the first law of robotics? No. Even a large apple is not likely to cause an accident, because the agent knows that ships are hundreds of feet long and weigh thousands of tons. Some simple knowledge about parameter ranges can help to avoid unintelligent behavior.

Parameter ranges of this kind have been entered into the agent's lexipedia, where appropriate, for all of the agent's ~250 objects and substances. The extreme range adjectives mentioned above typically specify a parameter as being on the high or low side of the average. Thus "large" means on the high side of the average, and "small" on the low side. In the agent's interpretation system, a parameter that differs from the average by more than 40% of the range between the average and the interval bound is assumed to qualify as an extreme parameter. In other words, "large" means in the interval $(\text{average} + 0.4 \times (\text{maximum} - \text{average}), \text{maximum})$ and "small" means in the interval $(\text{minimum}, \text{minimum} + 0.6 \times (\text{average} - \text{minimum}))$. Thus a large apple means having a diameter between 3.8 and 5 inches. Rare cases which exceed the 99% bounds are classified according to the nearest normal subinterval, so that a six-inch apple is classed as large. Actually, it should be called "very large." However, at present, use of "very" with such adjectives has not been implemented. Clearly, "very" would apply to a much narrower interval at the extremes of the normal range, perhaps the top and bottom 10%. We are fully aware that this is only an approximation to English usage, and that there will be situations, particularly near the boundaries between normal and extreme, where this interpretation will differ from the usage of some native English speakers. Nevertheless, it seems to be a useful approximation for most cases. Obviously, a distribution function would be more accurate than an average plus upper and lower bounds. However, for the vast majority of common objects, a distribution function seems out of the question, or at least exceedingly difficult to obtain.

11. Reply to some criticisms

At this point we would like to answer a few criticisms of the agent, which we have either already heard or expect to hear from various constituencies.

The agent doesn't do buzzwording as well as Figby's system

At one time it was thought that intelligence and AI had something to do with generality. The agent is subject to criticism from researchers specializing in one intelligence component, who may be able to point out that their one component works better than the corresponding component of an integrated agent. However, agents deserve credit for their "horizontal depth." Like an athlete in the decathlon, an agent should be able to earn points for its performance in each cognitive event, without having to beat all of the narrower systems in their one specialty event. (In fact, the idea of an "AI decathlon" as an organized event, like the computer chess competitions, seems worth pursuing by someone with organizational skills.) Researchers specializing in a particular component of intelligence should realize that an integrated agent is in fact a validation and justification of their own work.

"Real" researchers work with real vehicles

The Seaworld simulation allows flexibility in experimenting with scenarios. e.g., testing the first law of robotics, that might be dangerous, illegal, time consuming, or prohibitively expensive to perform in the real world. However, some robotics-oriented researchers are pleased to believe that simulations are too simplistic and deceptive for serious AI investigators. We tend to agree that if your interest is robotics, low-level control, or perception, then it is probably a good idea to work with the genuine article. However, our agent emphasizes cognitive-level functions. On this end of the spectrum, very challenging problems can be addressed more efficiently by working with a good world simulation. When the agent's cognitive processes are sufficiently mature, we intend to integrate them into a real-world system. In the meantime, by working with a simulated world, the distinct issues of perception, robotics, and low-level control can be factored out, and the high engineering costs associated with real-time vehicle projects can be avoided. This position really represents a middle ground on this issue, since much of the work on "agents" in AI doesn't interface with any world at all, even a simulated one! Simulations are a useful and accepted tool in many other areas of science and technology. Aircraft designers, earthquake researchers, nuclear fusion physicists, and cosmologists all use simulations routinely. Military officers practice their skills on war game simulations. Airline pilots qualify on new aircraft types by training on simulators. There is simply no good reason why the powerful tool of simulation should be denied to AI researchers.

Suppose an ichthyosaur rams the sub? The agent's concept of beauty is flawed

The real universe is not completely predictable. Even the wisest and most intelligent people work with only approximate mental models and incomplete knowledge. They are sometimes surprised at events, and their plans sometimes go awry. For this reason we believe that the quest for absolute certainty and infallibility in AI systems is misguided. The agent has been constructed using the 99% rule, which means that low-probability events, Ripleyesque facts, and contrived sophist counterexamples are politely ignored. Homer is expected to make mistakes just like everyone else.

In the same way, we are at present content to work with approximations to the meanings of common natural language concepts. As an example, the adjectives "pretty" and "beautiful" are not perfect synonyms, beautiful having the nuance of higher degree. Yet to a first approximation they mean the same, and Homer uses the same meaning for both, namely "that which increases happiness in people when they see or hear it." Much can be done with approximate meanings, especially if we intend the agent to engage in useful but ordinary activities (where verbal nuance can usually be ignored), rather than write screenplays or negotiate treaties (where verbal nuance is obviously important). For a useful agent, it isn't necessary to rival the linguistic abilities of a university-educated adult. If someone could even duplicate the capabilities of a three-year-old child, they would deserve a Nobel prize.

The agent is too ad hoc

Slightly different standards of judgment apply to theoretical and experimental AI work. Theories are supposed to be

elegant, hygienic, and infallible. Experimental systems are supposed to work. To build a large, complex AI system in finite time, simplicity can be more valuable than intricate theoretical elegance.

Autonomous vehicles don't really need natural language

People are very properly concerned about autonomous vehicles running amok. It is desirable to be able to interrogate an autonomous vehicle, before it departs, about what it is going to do and how it is going to carry out a command. It is desirable to be able to communicate with the vehicle while it is out on a mission, to get information, and give it advice on how to get out of scrapes. It is desirable to be able to "debrief" the vehicle when it returns from a mission, to find out what happened and why. These communications are most easily conducted in natural language. Thus an autonomous vehicle that can communicate in limited natural language offers enhanced capability over one that can't. Technical innovations, such as indoor plumbing and bit-mapped graphics, often begin as extravagant capability enhancements and then evolve into "necessities" when people become accustomed to them.

12. Difficulties, limitations, and future directions

Building a complete agent has been somewhat laborious. The system is fragile. The agent system has now grown to the point where some inertia is experienced in extending it. As feature X is added, feature Y stops working in certain cases and has to be repaired. Good progress is still being made, but at a slower rate than at the beginning. Less than half of the projected vocabulary has so far been implemented. There are certainly many sentences that could be constructed even with the implemented vocabulary that would not be properly understood, since interpretation of sentence meaning is literal and does not make extensive use of context (exceptions being pronouns and definite noun phrases). Many of the common English verbs represent tough concepts, almost demanding a separate theory for each one. Examples are "mean" (the verb), "keep," "select," "prevent," "work," and "allow." The deep analysis of the meaning of some of the most common words in the English language has not yet been done. This is a research area which could tolerate more emphasis.

Homer is a little slow, in the most literal sense. Interpretation and generation of natural language sentences take at most a second or two. However, plan generation takes from 10 seconds to several minutes. Question answering requires anywhere from a few seconds to a few minutes, depending on difficulty. The system for retrieval of schema events in episodic memory is inefficient and needs to be reimplemented. As the lifespan of the agent increases, the number of events in its episodic memory gradually degrades the speed of question answering involving personal events. For this reason, the lifespan of the agent is usually kept short. However, the efficiency of the present set of reflective demons is not appreciably affected by the size of the experience base.

The Seaworld domain has limitations in attempting to exercise a generalist agent, since many common objects and activities only occur naturally on land. Concepts like "school," "horse," and "bed" are difficult to work into a Seaworld scenario, although boats do have beds and horses could be on board a transport ship. However, this relates to the exercise of general world knowledge and vocabulary.

We would like to let Homer have shore leave and experience life on land. We would also like to upgrade the world from two-dimensional to three-dimensional graphics, for greater realism, and to allow scenarios exercising concepts involving the third spatial dimension, like "raise" and "under."

Many common concepts have a heavy spatial or visual content to them, which is not adequately captured by writing logical statements, describing state transitions, or setting up class hierarchies. Examples are "edge," "groove," and "corner." These are going to require some kind of analog-style representation for adequate understanding. This representation would be something like an internal mind's eye graphics screen on which scenes and relationships would be temporarily "drawn" for a variety of purposes.

We intend to fill in a lot more world knowledge, using the common English vocabulary of the agent as a guide. This is based on the hypothesis that word frequency roughly correlates with knowledge frequency. In other words, if a bit of world knowledge is frequently used, there is probably a direct association with a frequently used word.

One of the most glaring limitations is the agent's terse responses to questions. More generally, the agent is too passive, usually only speaking when spoken to (reflective comments excepted), and usually only acting when commanded. It is desirable for the agent to be able to initiate dialog, instead of just answering questions and giving reports on its actions and observations. It seems most natural for the agent to plan speech actions (requests, questions, etc.) just as it plans overt physical actions. For example, one way to learn the location of an object is to ask a person. One way to accomplish a goal is to ask another person to achieve the goal for you. And to enable people to achieve a delegated goal, the agent will need to make statements to supply needed information and motivation.

In summary, this paper has been a progress report on an experimental attempt to build an integrated agent. The experiment is continuing in the directions described in this section. The results thus far have been encouraging.

- ALLEN, J.F. 1987. Natural language understanding. Benjamin/Cummings, Menlo Park, CA. Sect. 11.4.
- ASIMOV, I. 1950. *I. Robot*. Signet Books, New York, NY.
- AZAR, B.S. 1984. Basic English grammar. Prentice-Hall, Englewood Cliffs, NJ.
- BICKMORE, T.W. 1988. An object-oriented approach to tactical text generation. Lockheed AI Center Technical Report, Palo Alto, CA.
- BORCHARDT, G.C. 1984. A computer model for the representation and identification of physical events. M.S. thesis, Computer Science Department, University of Illinois at Urbana-Champaign, IL.
- CERCONE, N., and SCHUBERT, L. 1975. Toward a state based conceptual representation. Proceedings of the 4th International Joint Conference on Artificial Intelligence, Tbilisi, U.S.S.R. pp. 88-90.
- CERMAK, L.S. 1982. Human memory and amnesia. Erlbaum, Hillsdale, NJ.
- DANLOS, L. 1987. A French and English syntactic component for generation. *In* Natural language generation. Edited by G. Kempen. Martinus Nijhoff, Dordrecht, The Netherlands. pp. 191-218.
- HALEY, V.B. 1977. International English. Charles E. Tuttle, Rutland, VT.
- JACKENDOFF, R. 1983. Semantics and cognition. MIT Press, Cambridge, MA.

- JOHNSON-LAIRD, P.N. 1982. Formal semantics and the psychology of meaning. *In Processes, beliefs, and questions. Edited by S. Peters and E. Saarinen.* Reidel, Boston, MA. pp. 1-68.
- KIBLER, D., and MORRIS, P. 1981. Don't be stupid. Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, B.C., pp. 345-347.
- KOLODNER, J.L. 1984. Knowledge-based self-organizing memory for events. *In Artificial and human intelligence. Edited by A. Elithorn and R. Banerji.* North-Holland, Amsterdam, The Netherlands. pp. 57-66.
- MCKEOWN, K.R. 1985. Text generation. Cambridge University Press, New York, NY.
- MCKEVITT, P., and WILKS, Y. 1987. Transfer semantics in an operating system consultant: the formalization of actions involving object transfer. Proceedings of the 10th International Joint Conference on Artificial Intelligence, Milan, Italy, pp. 569-575.
- MILLER, G.A., and JOHNSON-LAIRD, P.N. 1976. Language and perception. Harvard University Press, Cambridge, MA.
- NATSOULAS, T. 1978. Consciousness. *American Psychologist*, 33: 906-914.
- NEUMANN, B. 1984. Natural language description of time-varying scenes. Technical Report No. 105, Fachbereich Informatik. University of Hamburg, Hamburg, Germany.
- NILSSON, N. 1983. Artificial intelligence prepares for 2001. *AI Magazine*, 4(4): 7-14.
- OGDEN, C.K. 1934. The system of basic English. Harcourt Brace Jovanovich, Inc., New York, NY.
- SCHANK, R.C. 1983. The current state of AI: one man's opinion. *AI Magazine*, 4(1): 3-8.
- SCHANK, R.C., and RIESBECK, C.K. 1981. Inside computer understanding: five programs plus miniatures. Erlbaum, Hillsdale, NJ.
- TEITELMAN, W. 1972. Automated programming[sic] — the programmer's assistant. Proceedings of the Fall Joint Computer Conference, pp. 917-921.
- TULVING, E. 1972. Episodic and semantic memory. *In Organization of memory. Edited by E. Tulving and W. Donaldson.* Academic Press, New York, NY. pp. 381-403.
- VERE, S.A. 1985a. Splicing plans to achieve misordered goals. Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, pp. 1016-1021.
- 1985b. Temporal scope of assertions and window cutoff. Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, pp. 1055-1059.
- 1987. Planning. *In Encyclopedia of AI. Edited by S. Shapiro.* Wiley, New York, NY. pp. 748-758.
- WALTZ, D.L. 1981. Toward a detailed model of processing for language describing the physical world. Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, B.C., pp. 1-6.
- 1982. The state of the art in natural-language understanding. *In Strategies for natural language processing. Edited by W. Lehnert and M. Ringle.* Erlbaum, Hillsdale, NJ. pp. 3-32.
- WILENSKY, R. 1983. Planning and understanding. Addison-Wesley, Reading, MA.
- 1986. Some problems and proposals for knowledge representation. Report No. UCB/CSD 86/294, Computer Science Division, University of California at Berkeley, CA.
- WINOGRAD, T. 1972. Understanding natural language. Academic Press, New York, NY.