

CSL *COORDINATED SCIENCE LABORATORY*

**AN EXPERT DISTRIBUTED
ROBOTICS SYSTEM WITH
COMPREHENSION AND
LEARNING ABILITIES IN THE
AIRCRAFT FLIGHT DOMAIN**

WALTZ, DAVID L.
DEJONG, GERALD F.

APPROVED FOR PUBLIC RELEASE. DISTRIBUTION UNLIMITED.

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Annual Technical Report for Research in

AN EXPERT DISTRIBUTED ROBOTICS SYSTEM
WITH COMPREHENSION AND LEARNING ABILITIES IN THE
AIRCRAFT FLIGHT DOMAIN

For the Period

January 1, 1983 - December 31, 1983

Submitted to

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Investigators:

D. L. Waltz, G. DeJong

February 1984

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS None		
2a. SECURITY CLASSIFICATION AUTHORITY N/A		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release, distribution unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) T-138		5. MONITORING ORGANIZATION REPORT NUMBER(S) N/A		
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab., Univ. of Illinois	6b. OFFICE SYMBOL (If applicable) N/A	7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research		
6c. ADDRESS (City, State and ZIP Code) 1101 W. Springfield Ave. Urbana, Illinois 61801		7b. ADDRESS (City, State and ZIP Code) Bolling Air Force Base Washington, D.C. 20332		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR	8b. OFFICE SYMBOL (If applicable) N/A	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER Contract No. F49620-82-K-0009		
8c. ADDRESS (City, State and ZIP Code) Bolling Air Force Base Washington, D.C. 20332		10. SOURCE OF FUNDING NOS.		
		PROGRAM ELEMENT NO. N/A	PROJECT NO. N/A	TASK NO. N/A
11. TITLE (Include Security Classification) An Expert Distributed Robotics System with				
12. PERSONAL AUTHOR(S) Comprehension and Learning Abilities in the Aircraft Flight Domain Waltz, David L. and DeJong, Gerald F.				
13a. TYPE OF REPORT Annual Technical	13b. TIME COVERED FROM 1/1/83 to 12/31/83	14. DATE OF REPORT (Yr., Mo., Day) February 1984	15. PAGE COUNT 40	
16. SUPPLEMENTARY NOTATION N/A				
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Artificial intelligence, knowledge representation, expert systems, natural language understanding, knowledge acquisition, perceptual reasoning, qualitative process theory.		
FIELD	GROUP			SUB. GR.
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Research has continued during the past year on critical components for a comprehensive expert system for on-board use in an aircraft. We report progress on (1) a system that can reason about the operation of a gas turbine engine; (2) a system about route and trajectory meta-planning; (3) a temporal reasoning system; (4) a system for extracting speaker goals from natural language dialogue; (5) systems for acquiring new knowledge schemas from natural language input; and (6) systems for high level perceptual reasoning.				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL		22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL None	

18. (continued)

metaplanning, speaker goal extraction, temporal reasoning, on-board aircraft applications, pilot aids, mechanism modeling

ABSTRACT

Research has continued during the past year on critical components for a comprehensive expert system for on-board use in an aircraft. We report progress on (1) a system that can reason about the operation of a gas turbine engine; (2) a system about route and trajectory meta-planning; (3) a temporal reasoning system; (4) a system for extracting speaker goals from natural language dialogue; (5) systems for acquiring new knowledge schemas from natural language input; and (6) systems for high level perceptual reasoning.

Keywords: Artificial intelligence, knowledge representation, expert systems, natural language understanding, knowledge acquisition, perceptual reasoning, qualitative process theory, metaplanning, speaker goal extraction, temporal reasoning, on-board aircraft applications, pilot aids, mechanism modeling

"Table of Contents"

1 Introduction	3
2 Spatial and Temporal Modelling in Natural Language	4
2.1 Progress	4
2.2 Proposed Work	5
3 Qualitative Modelling in the Aircraft Engine Domain	6
3.1 Why a Qualitative Model of the Engine?	6
3.2 Issues in Modeling the Engine	7
3.3 Uses of the Current Model	9
3.4 Summary	11
4 Metaplanning System for Air Traffic Control	13
5 Explanatory Schema Acquisition	15
5.1 Introduction	15
5.2 Current Status	15
5.3 System Organization	16
5.4 Sample Transcript	17
6 Incremental Scene Interpretation	32
7 Something or the Other	34
8 References	36
9 Publications	38
10 Personnel	39
11 Inventions and Patent Disclosures	40

1. Introduction

Research has continued during the past year on critical components for a comprehensive expert system for on-board use in an aircraft. In particular, we have made substantial progress on a system that can reason about the operation of a gas turbine engine, and on a system about route and trajectory meta-planning. The ability to represent and reason about timing and temporal relations is central to such an expert system, so we have given special emphasis to a temporal reasoning system. An expert system of the sort we envision must also be able to communicate with human users in natural language; research has also been concentrated on extracting speaker goals from dialogue, and on learning new knowledge from natural language and input. Finally, we have investigated high level perceptual reasoning; such reasoning is necessary for recognizing objects and relations between them, and is also needed for assessing a general situation, based on sensor readings.

While much has been accomplished, much also remains to be done. In particular, the integration of all these systems into a unified aircraft expert system must await the completion of the components, and will in addition require advances in our understanding of the judgement of importance, commonsense reasoning, retrieval and use of knowledge relevant to a current problem; in addition, a very large knowledge base and sophisticated sensor interpreters will also be required. Our goal has been to demonstrate the feasibility of constructing some of the key components. Our progress toward this goal is summarized in the following sections.

2. Spatial and Temporal Modelling in Natural Language

2.1. Progress

Temporal knowledge plays a fundamental role in not only our understanding of times, dates, events, and actions, but in our understanding of basic natural language as well as our planning and remembering processes. The goal of this research is to design and construct a natural language system which will extract temporal knowledge from language, as well as construct inferences which are commonly made from that knowledge, and link these to knowledge about physical causality as well as spatial knowledge. The design of the system is based not only on past work in temporal and spatial representation, as well as on the ongoing research in natural language universals being conducted by Dr. La Raw Maran here at CSL.

At the current time construction of the NALATIK system (NATural Language Temporal Inference and Knowledge system) is well underway. The two lowest levels, the time interval level, and the event level, have been designed and implemented for the first time in Interlisp [Teitelman78], on a Xerox 1108 Scientific Information Processor. The time interval level represents pure temporal information, consisting of intervals and instants in time, linked into a relational network. The event level describes primitive events, and their relationships to time intervals and points. Events also are linked to one another and to spatial and causal information at the event level. A complete description of this system may be found in [Spoor83].

¹This research is being carried out by David T. Spoor, a graduate student, under the supervision of Prof. David L. Waltz.

2.2. Proposed Work

Presently the NALATIK project is preparing for a major push to link up the representations already built with natural language input. A comprehensive picture of temporal references and usage in natural language has recently emerged from the cognitive universals in natural language project. In the near future we hope to utilize this data, along with an already implemented and tested parser, to generate temporal representations directly from natural language. After linking the parser and the representation system we will analyze the operation of the system, and make revisions where necessary.

Additionally we hope to incorporate both causal and spatial knowledge in our system, linking it to the current event representation, and parsing system. Finally we will extend our base temporal representation as dictated by the development outlined above. While we have endeavored to produce as complete a representation as possible, we hope to learn yet more about temporal representation as the above outlined development occurs.

The current system is targeted at understanding descriptions of aircraft operation during both approaches and departures. This system will take natural language descriptions of aircraft motions, and configuration changes, from which it will build an internal model. From this internal model, and knowledge about standard operating procedures, both probable trajectory and configuration can be determined, along with any unusual departures from standard procedures.

3. Qualitative Modelling in the Aircraft Engine Domain

We have studied the aircraft turbo-jet engine in order to determine the extent to which qualitative modeling is useful in the engine domain. Qualitative modeling is a recent field of study in artificial intelligence, and has been useful for such tasks as circuit recognition [Kleer79], troubleshooting [Kleer79] [Forbus82], and simulation [Cross83] [Forbus81].

3.1. Why a Qualitative Model of the Engine?

A good qualitative model of an aircraft engine has many possible uses. The model could be used as an expert system: 1) to make engine simulations less expensive by constraining the equations which need be solved in a numerical simulator; 2) to explain the results of a numerical simulation by giving causal explanations for given change, which would then be useful for troubleshooting; 3) to predict engine response to changes in the input; and 4) to detect approaching operational limits, provide warnings to the pilot, and to give suggestions for avoiding those limits.

In order to fully understand the utility of the above uses, it is important to understand the capabilities of existing numerical simulators. A numerical engine simulator has the ability to predict the predict the real-time response of the engine to input changes. During the simulation, the user may request a plot of the behavior of desired parameters, or ask the simulator to project future trends. It is also possible to add rules to the simulator so that it can identify operational limits.

²This research is being carried out by Raman Rajagopalan, a research assistant, under the supervision of Prof. David L. Waltz.

Although the capabilities of numerical simulators are extensive, the results are obtained only after applying very complex mathematical techniques. Furthermore, current numerical simulators provide only numerical values as results, and any interpretation of these values is left to the user. It is also not possible for the numerical simulator to provide explanations of its results (e.g., the numerical simulator cannot provide the user with the causes for the changes it predicts). From these facts, we can see that a qualitative model of the engine will be valuable both in terms of computation time as well as the ability to provide more information. To what extent are the goals given above realizable? What information should one include in a qualitative model of the engine? In what way should available information be represented? These are the kinds of questions we have addresses in our research.

3.2. Issues in Modeling the Engine

Before we discuss our model and its uses in its current form, we will describe the difficulties found in modeling the engine. One difficulty in describing the operation the engine is due to the presense of nonlinearities. These nonlinearities arise due to the existence of a continuous and simultaneous feedback between the turbine and the compressor, and by the structure and operation of the compressor and turbine. Since the operation of the engine is dependent upon its current state, and because of the inherent nonlinearities, it is not possible to easily maintain rates of change and times within the engine, and therefore transient analysis is impractical for a qualitative model. Without transient analysis, we cannot detect operational limits as they occur, can only point out that the operation of the engine is close to an operational limit when the engine is

in a steady state condition. If, in going from a steady state to a desired state, a limit was exceeded, we also have a limited ability to detect and explain which changes could have contributed to this condition. This ability will be useful for diagnosis and troubleshooting.

In order to take the first step in understanding the operation of the turbo-jet engine, we have built a causal model of the engine. This is a model of the relationships which exist between operational parameters of the engine. From studying basic engine texts [Group80,Treager79], we have found that all the relationships which exist between any two parameters can be expressed by the notation I+, I-, and I. I+ and I- relationships are linear, and indicate that for a given change in a parameter (increase or decrease), the other parameter will behave in either a positively correlated (I+) or negatively correlated (I-) fashion. The symbol I indicates that the two parameters share a nonlinear relationship. In the current model, two types of nonlinear relationships exist, those which can be called piece-wise linear, and those described by a convex curve. We have included qualitative models of these curves; the exact effect is dependent on the current state of the engine. Each such relationship also includes information on the time taken for the change in one parameter to propagate to the other. The time taken for a particular change is not given in real-time, but is a comparison with the other processes which occur.

Once the relationships which exist between parameters were defined, the task of organizing the available information remained. In order to attain maximum flexibility, and usefulness, we modeled the engine in a hierarchical fashion, and individually represented the major components at each hierarchical level. We have considered two levels, the top level being a

representation of the engine as a black box which takes in air and fuel (inputs) and produces thrust (output).

The second level includes models of the major components of the engine. Each such component is modeled individually and contains default values for structural parameters (sizes of ducts, etc), a description of the operation limits of that part, a description of operation of that part in terms of the relationships described earlier, and finally a quantitative description of the part. The parts we have modeled include the compressor, combustor, turbine, and exhaust. In addition, we have included models of the relationship between the environment and the operation of the engine, as well the effect of changes in the throttle setting in the cockpit.

The current model contains a total of thirty-five relationships between twenty-seven parameters and rules for detecting nine operational limits. A smaller number of default values, representations of nonlinear curves, and equations also exist. Finally, we have included rules which can detect unrealistic inputs (e.g., an altitude change without a corresponding change in airspeed or throttle setting).

3.3. Uses of the Current Model

What are the uses of this model? The current model is useful for analyzing the changes that may have taken place in going from one state of to another, for predicting the changes in internal parameters of the engine when the inputs are perturbed, for providing a first step in diagnosis, and in providing suggestions to a pilot in the event of engine trouble. We have implemented the model to simulate the effects of changes in the inputs to the engine (altitude, airspeed, and throttle setting). In addition, we have

also added an explanation generation capability whereby any change predicted by the simulation can be explained.

The simulation of internal changes due to input changes is not a trivial process. Conflicts arise during such a simulation, and since we do not have time (real-time) and rates of change available, we cannot resolve the conflicts directly. of the engine, this information cannot be easily added to the model. When analysis is the goal, we have both initial and final state values available, and this information may be readily applied in order to resolve conflicts.

Unlike analysis, we do not have the 'final state' information available when the goal is prediction. Here, the fact that we have a hierarchical model of the engine is useful. From the top-level model of the engine, the change in thrust due to any input change can be found. Then, at the second level, the end result (the change in thrust) is known, from which conflicts can be resolved by choosing the path which leads to the determined change in thrust.

Once the changes in the operational parameters have been determined, it is possible to determine whether any operational limits could have been exceeded. However, without quantitative information, a definite statement of whether or not the engine has exceeded a limit is not possible.

The results of the simulation provide a starting point for making suggestions about avoiding operational limits. In addition to having information available as to the changes in the parameters, we also have timing information available, as well as the paths followed by nonlinear relationships.

Since the time information is a comparison with other processes, we know which processes are the fastest. This is useful when in making suggestions in an emergency situation, where time is a major constraint.

The information concerning paths is useful in determining if any interesting points (e.g., maxima, minima, inflection points, break points, etc.) were crossed, and in giving insight into how close a given operating point is to such a point. This information could be useful for detecting whether the engine is likely to enter a new state of operation. For example, if the engine is operating at top efficiency, we know that any further increase in the fuel-air ratio will not have as positive an effect.

Having information available on the parameter changes, including points of conflict or coincidence, is useful for identifying alternate methods of accessing a given parameter. Paths may exist both increasing or decreasing the value of a parameter. This information can be used to suggest actions to take when a limit is being approached.

3.4. Summary

We have studied the turbo-jet engine to determine of feasibility and usefulness of qualitative modeling in this domain. We have found that expert systems based on this technology have real potential for analysis, prediction, diagnosis, and troubleshooting. In spite of the fact that quantitative information is not readily available, we have found that useful qualitative models are possible, and the proposed uses are possible to a limited extent. We have implemented the model so that it simulates the results of changes in engine inputs. This simulation can be a pure simulation (prediction), or employ the results of a quantitative simulation

(analysis). Finally, there is much scope for the use of a qualitative model; however, a model of the structure of the engine and a limited quantitative model of the engine will have to be integrated with our qualitative system in order to realize all the possibilities.

4. Metaplanning System for Air Traffic Control

Two approaches have resulted from the AI attempts to create general problem solvers: one technique, dating back to the 1950's, relies on uniform algorithms with no special domain knowledge to search a solution space for a workable sequence of operations; the other technique, of more recent vintage, incorporates domain knowledge in default plans that more or less fit a range of problems, and then uses planning "metaknowledge" (knowledge about knowledge) to patch these plans to fit specific problems. Thus far the best developments of the latter approach have been in Lenat's work on heuristics [Lenat80] and Wilensky's work on metaplanning [Wilensky81]. In both cases much effort has been devoted to discovering the metaknowledge of problem solving. Other issues such as the representation of this knowledge and the structure of the problem solving engine are also part of the research. Our work has been to develop metaplanning theory and produce new results on the content of planning metaknowledge and its representation and use in a planning system.

We are creating a metaplanning system for the domain of air traffic control. This domain allows problem solving in a variety of modes. Common algorithmic approaches yield partial solutions to conflict resolution, flow control, and routing, but cannot handle the range of variables influencing the problem (e.g. pilots demands, winds, and route layout). An AI approach offers greater coverage of the domain. Using metaplanning an expert's planning knowledge can be captured in default plans in much the same way that deductive knowledge is encoded in a rule-based expert system. AI

³This research is being carried out by Shaun Keller, a graduate student, previously under the supervision of Prof. R. T. Chien and currently under the supervision of Prof. David L. Waltz.

contributes the planning metaknowledge used to select order, merge, and otherwise modify the default plans for specific problems. Other research at CSL has developed qualitative models of aircraft performance equations and related this model to interpreting air traffic control commands, but the results have not been incorporated into the current effort [Cross83].

Requisite domain and metaknowledge for the collision avoidance task has been studied as well as a frame-based representational structure. An architecture for the planning system has been created. The system operates by watching the air traffic and projecting along flight plans looking for conflicts. Detection of conflicts triggers a metaplan to classify the problem and activates an appropriate default plan. Simulation determines if the default plan will work. If simulation shows plan failure then the failure type triggers the appropriate metaplan for plan modification. A common reason for default plan failure is that it causes the violation of some goal. For example, resolving one conflict in a certain way may cause another. A metaplan to resolve goal conflict may call for another choice to be made in the steps of the default plan, or another default plan may have to be inserted to correct the problem. Work is proceeding on system implementation in Interlisp-D on a Xerox 1108.

5. Explanatory Schema Acquisition

5.1. Introduction

Any system which is to use Natural Language (NL) to interact with a user must have at its disposal a large collection of real world knowledge [DeJong82a,Waltz82]. This shared repertoire of world knowledge is the common ground upon which language communication is based.

Past attempts to create such systems have relied on various methods of encoding such real world knowledge, ranging from formal, mathematical descriptions such as the predicate calculus to frame based systems [Bobrow77,Minsky75] and schemata [Bobrow75,DeJong79,Schank77]. Whatever the choice of representation, the bottleneck in designing and building such systems seems to lie in the acquisition of world knowledge.

One important area of current Artificial Intelligence (AI) research, therefore, is automated learning. Our research is centered on the construction of a schema-based system incorporating a new type of learning process (Explanatory Schema Acquisition) where a case for "human justifiability" has already been made [DeJong82b]. A schema is a collection of objects, events and actions which are packaged together to provide a natural-language understanding system with convenient framework for representing and accessing its large core of world knowledge.

5.2. Current Status

We are building an explanatory schema acquisition system in the story processing domain. When the system is given a story input for which it has

⁴This research is being carried out by Ashwin Ram and Alberto Segre, two graduate students, under the supervision of Prof. Gerald DeJong.

no matching schema, the system should be capable of either creating a new schema, modifying an existing schema, or combining several existing schemata in order to provide enough world knowledge to adequately explain (and therefore "understand") the story input. This implementation should help to flesh out the ideas expressed in [DeJong82b], providing a testing ground on which to check these ideas for completeness and consistency.

In short, the various techniques covered by use of the term "Explanatory Schema Acquisition" come into play after the system has already done its best in understanding as much as possible of the input story. This "understanding" consists of the construction of a causally connected internal model of the actions, states and events described in the input story. The causal connections which underly the character's actions in the story are used to determine whether there are any interesting (i.e., possibly useful in understanding future situations). If such a condition exists, the learning portion of the system attempts to generalize a new schema from the story model. The new schema could be a combination of existing schemata ("Schema Composition"), an alteration of an existing schema ("Schema Alteration"), a transformation of an existing schema into a volitional schema ("Volitionalization"), or the use of an existing schema to achieve a side effect ("Secondary Effect Elevation").

5.3. System Organization

The system can be roughly divided into three components. The first component (the parser) takes English input and translates this input into an intermediate form or conceptual representation. While much work has been done in this area by other researchers, it is beyond the scope of this research.

The second component (the understander) takes conceptual input and builds a causally connected model of the events in the input story. The understander relies heavily on what knowledge it already possess about the world the characters interact with. In addition, the understander maintains individual character goal structures in order to explain why certain characters perform certain actions. As of the reporting date, the understander is almost complete. A transcript of a session where the understander "acts" on a sample input is included below.

The third component (the learning subsystem) operates on the model constructed by the understander. It relies heavily on failed expectations to trigger the learning process. This portion of the system will be the next order of business.

The system is coded in INTERLISP-D and runs on the Xerox 1108 series LISP machines.

5.4. Sample Transcript

What follows is a sample transcript of the understander operating on a sample story which deals with a kidnapping (the system has no prior knowledge of what a kidnapping is). The expressions marked "Story input:" are the inputs used by the understander. Those inputs beginning with an asterisk are the English language equivalents of the respective conceptual representation and are ignored by the understander. They are included only for the convenience of the reader.

```
1_(ProcessStory KIDNAP]
```

```
Processing story...
```

Story input:

(* Fred is Mary's father.)

Story input:

Processing:

```
[PARENT [SUBJECT (PERSON (NAME (FRED]
          (OBJECT (PERSON (NAME (MARY]
```

Adding FRED2 to model in OBJECT bucket.
Adding PERSON4 to model in CHARACTER bucket.
Initializing a character model for PERSON4 (FRED2)
Adding MONEY2 to model in OBJECT bucket.
Adding MARY2 to model in OBJECT bucket.
Adding PERSON5 to model in CHARACTER bucket.

Initializing a character model for PERSON5 (MARY2)

Adding PARENT2 to model in STATE bucket.

Primary inference from PARENT2: CARES-FOR

Processing:

```
[CARES-FOR [SUBJECT (PERSON (NAME (FRED]
            (OBJECT (PERSON (NAME (MARY]
Adding CARES-FOR3 to model in STATE bucket.
```

Primary inference from PARENT2: CARES-FOR

Processing:

```
[CARES-FOR [SUBJECT (PERSON (NAME (MARY]
            (OBJECT (PERSON (NAME (FRED]
Adding CARES-FOR4 to model in STATE bucket.
```

Story input:

(* Fred is rich)

Story input:

Processing:

```
[POSSESS [SUBJECT (PERSON (NAME (FRED]
          (OBJECT (MONEY (AMOUNT (?N]
```

Adding POSSESS8 to model in STATE bucket.

Story input:

(* John approached Mary.)

Story input:

Processing:

[MOVE [ACTOR (PERSON (NAME (JOHN]
 (TO (LOCATION (OF (PERSON (NAME (MARY]

Adding JOHN2 to model in OBJECT bucket.
Adding PERSON6 to model in CHARACTER bucket.

Initializing a character model for PERSON6 (JOHN2)

Adding LOCATION4 to model in OBJECT bucket.
Adding MOVE5 to model in ACTION bucket.

Precondition of MOVE5: POSSESS-SOCIAL-CONTROL

Processing:

[POSSESS-SOCIAL-CONTROL [SUBJECT (PERSON (NAME (JOHN]
 (OBJECT (PERSON (NAME (JOHN]
Adding POSSESS-SOCIAL-CONTROL2 to model in STATE bucket.

Trying to account for JOHN2's MOVE5:

Primary effect of MOVE5: AT

Processing:

[AT [SUBJECT (PERSON (NAME (JOHN]
 (LOCATION (LOCATION (OF (PERSON (NAME (MARY]
Adding AT4 to model in STATE bucket.

Story input:

(* John pointed a gun at Mary.)

Story input:

Processing:

[AIM [ACTOR (PERSON (NAME (JOHN]
 (OBJECT (GUN))
 (AT (PERSON (NAME (MARY]

Adding GUN2 to model in OBJECT bucket.
Adding AIM2 to model in ACTION bucket.

Precondition of AIM2: SPATH

Processing:

[SPATH [SUBJECT (PERSON (NAME (JOHN]
 (OBJECT (PERSON (NAME (MARY]
Adding SPATH2 to model in STATE bucket.

Precondition of AIM2: POSSESS

Processing:

(POSSESS [SUBJECT (PERSON (NAME (JOHN]
(OBJECT (GUN)))

Adding POSSESS10 to model in STATE bucket.

Hypothesizing BUY to achieve POSSESS10

Processing:

(BUY [ACTOR (PERSON (NAME (JOHN]
(OBJECT (GUN)))

Adding BUY2 to model in ACTION bucket.

Trying to account for JOHN2's BUY2:

Trying to account for JOHN2's AIM2:

Priming THREATEN on the basis of AIM2.

((THREATEN [ACTOR (PERSON (NAME (JOHN]
[SUBJECT (PERSON (NAME (MARY]
(INSTRUMENT (GUN)))

[(MTRANS [ACTOR (PERSON (NAME (JOHN]
(TO (PERSON (NAME (MARY]

AIM2)

Story input:

(* John told Mary to get into his car.)

Story input:

Processing:

[MTRANS

[ACTOR (PERSON (NAME (JOHN]

[TO (PERSON (NAME (MARY]

(MOBJECT

(GOAL

[SUBJECT (PERSON (NAME (JOHN]

(OBJECT (MOVE [ACTOR (PERSON (NAME (MARY]

(TO (LOCATION (OF (CAR (OWNER (PERSON (NAME (JOHN]

Adding CAR2 to model in OBJECT bucket.

Adding LOCATION5 to model in OBJECT bucket.

Adding MOVE6 to model in ACTION bucket.

Adding GOAL16 to model in GOAL bucket.

Adding MTRANS2 to model in ACTION bucket.

Precondition of MTRANS2: CPATH

Processing:

[CPATH [SUBJECT (PERSON (NAME (JOHN]

(OBJECT (PERSON (NAME (MARY]
Adding CPATH2 to model in STATE bucket.

Activating schema THREATEN:

Processing:

(THREATEN [ACTOR (PERSON (NAME (JOHN]
[SUBJECT (PERSON (NAME (MARY]
(INSTRUMENT (GUN)))

Adding THREATEN2 to model in ACTION bucket.

Precondition of THREATEN2: SPATH
Precondition of THREATEN2: POSSESS
Trying to account for JOHN2's THREATEN2:
Primary effect of THREATEN2: BELIEF

Processing:

[BELIEF [SUBJECT (PERSON (NAME (MARY]
(OBJECT (IN-DANGER [SUBJECT (PERSON (NAME (MARY]
(FROM (PERSON (NAME (JOHN]

Adding IN-DANGER2 to model in OBJECT bucket.

Adding BELIEF3 to model in BELIEF bucket.

Primary inferences from BELIEF3:

Adding GOAL17 (ESCAPE2) to MARY2's character model
Adding GOAL18 (SUBSUME-GOALS2) to MARY2's character model

Trying to account for JOHN2's MTRANS2:
Primary effect of MTRANS2: BELIEF

Processing:

[BELIEF
[SUBJECT (PERSON (NAME (MARY]
(OBJECT
(GOAL
[SUBJECT (PERSON (NAME (JOHN]
(OBJECT
(MOVE [ACTOR (PERSON (NAME (MARY]
(TO (LOCATION (OF (CAR (OWNER (PERSON (NAME (JOHN]

Adding BELIEF4 to model in BELIEF bucket.

Primary inferences from BELIEF4:

Anticipating:

ACTION MOVE6
ACTOR PERSON5 (MARY2)
FOR PERSON6 (JOHN2)

Adding COMPLIANCE-BOX2 (MOVE6) to MARY2's character model

Story input:

(* John drove Mary to his hotel.)

Story input:

Processing:

[DRIVE [ACTOR (PERSON (NAME (JOHN]
[PASSENGER (PERSON (NAME (MARY]
[TO (LOCATION (OF (MOTEL (RESIDENT (PERSON (NAME (JOHN]
(VEHICLE (CAR (OWNER (PERSON (NAME (JOHN]

Adding MOTEL2 to model in OBJECT bucket.
Adding LOCATION6 to model in OBJECT bucket.
Adding DRIVE2 to model in ACTION bucket.

Precondition of DRIVE2: AT

Processing:

[AT [SUBJECT (PERSON (NAME (JOHN]
(LOCATION (LOCATION (OF (CAR (OWNER (PERSON (NAME (JOHN]
Adding AT5 to model in STATE bucket.

Hypothesizing MOVE to achieve AT5

Processing:

[MOVE [ACTOR (PERSON (NAME (JOHN]
(TO (LOCATION (OF (CAR (OWNER (PERSON (NAME (JOHN]
Adding MOVE7 to model in ACTION bucket.

Trying to account for JOHN2's MOVE7:

Precondition of DRIVE2: AT

Processing:

[AT [SUBJECT (PERSON (NAME (MARY]
(LOCATION (LOCATION (OF (CAR (OWNER (PERSON (NAME (JOHN]
Adding AT6 to model in STATE bucket.

Hypothesizing MOVE to achieve AT6

Processing:

[MOVE [ACTOR (PERSON (NAME (MARY]
(TO (LOCATION (OF (CAR (OWNER (PERSON (NAME (JOHN]
Adding MOVE8 to model in ACTION bucket.

Trying to account for MARY2's MOVE8:

MARY2 performed MOVE8 for (JOHN)'s benefit
Found COMPLIANCE-BOX2 in character model of MARY2
ACTOR PERSON5 (MARY2)
ACTION MOVE8
FOR PERSON6 (JOHN2)
Found GOAL18 in character model of MARY2
ACTOR PERSON5 (MARY2)
SUBSUME-GOALS2 OF PERSON6 (JOHN2)

Precondition of DRIVE2: POSSESS

Processing:

[POSSESS [SUBJECT (PERSON (NAME (JOHN]
 (OBJECT (CAR (OWNER (PERSON (NAME (JOHN]
Adding POSSESS11 to model in STATE bucket.

Hypothesizing BUY to achieve POSSESS11

Processing:

[BUY [ACTOR (PERSON (NAME (JOHN]
 (OBJECT (CAR (OWNER (PERSON (NAME (JOHN]
Adding BUY3 to model in ACTION bucket.

Trying to account for JOHN2's BUY3:

Trying to account for JOHN2's DRIVE2:
Primary effect of DRIVE2: AT

Processing:

[AT [SUBJECT (PERSON (NAME (JOHN]
 (LOCATION (LOCATION (OF (MOTEL (RESIDENT (PERSON (NAME (JOHN]
Adding AT7 to model in STATE bucket.

Primary effect of DRIVE2: AT

Processing:

[AT [SUBJECT (PERSON (NAME (MARY]
 (LOCATION (LOCATION (OF (MOTEL (RESIDENT (PERSON (NAME (JOHN]
Adding AT8 to model in STATE bucket.

Story input:

(* John locked Mary in his hotel room.)

Story input:

Processing:

[CONFINES [ACTOR (PERSON (NAME (JOHN]
 [SUBJECT (PERSON (NAME (MARY]
 (IN (ROOM [RESIDENT (PERSON (NAME (JOHN]
 (LOCATION (MOTEL]

Adding ROOM1 to model in OBJECT bucket.
Adding CONFINES1 to model in ACTION bucket.

Precondition of CONFINES1: AT

Processing:

[AT [SUBJECT (PERSON (NAME (JOHN]
 (LOCATION (ROOM [RESIDENT (PERSON (NAME (JOHN]
 (LOCATION (MOTEL (RESIDENT (PERSON (NAME (JOHN]

Adding AT9 to model in STATE bucket.

Hypothesizing MOVE to achieve AT9

Processing:

[MOVE [ACTOR (PERSON (NAME (JOHN)
 (TO (ROOM [RESIDENT (PERSON (NAME (JOHN]
 (LOCATION (MOTEL (RESIDENT (PERSON (NAME (JOHN]

Adding MOVE9 to model in ACTION bucket.

Trying to account for JOHN2's MOVE9:

Precondition of CONFINE1: AT

Processing:

[AT [SUBJECT (PERSON (NAME (MARY]
 (LOCATION (ROOM [RESIDENT (PERSON (NAME (JOHN]
 (LOCATION (MOTEL (RESIDENT (PERSON (NAME (JOHN]

Adding AT10 to model in STATE bucket.

Hypothesizing MOVE to achieve AT10

Processing:

[MOVE [ACTOR (PERSON (NAME (MARY]
 (TO (ROOM [RESIDENT (PERSON (NAME (JOHN]
 (LOCATION (MOTEL (RESIDENT (PERSON (NAME (JOHN]

Adding MOVE10 to model in ACTION bucket.

Trying to account for MARY2's MOVE10:

MARY2 performed MOVE10 for (JOHN)'s benefit
Found GOAL18 in character model of MARY2
ACTOR PERSON5 (MARY2)
SUBSUME-GOALS2 OF PERSON6 (JOHN2)

Trying to account for JOHN2's CONFINE1:

Story input:

(* John called Fred.)

Story input:

Processing:

[DIAL-TELEPHONE [ACTOR (PERSON (NAME (JOHN]
 (SUBJECT (PERSON (NAME (FRED]

Adding DIAL-TELEPHONE1 to model in ACTION bucket.

Trying to account for JOHN2's DIAL-TELEPHONE1:

Primary effect of DIAL-TELEPHONE1: CPATH

Processing:

[CPATH [SUBJECT (PERSON (NAME (JOHN]
(OBJECT (PERSON (NAME (FRED]

Adding CPATH3 to model in STATE bucket.

Primary effect of DIAL-TELEPHONE1: CPATH

Processing:

[CPATH [SUBJECT (PERSON (NAME (FRED]
(OBJECT (PERSON (NAME (JOHN]

Adding CPATH4 to model in STATE bucket.

Priming TELEPHONE on the basis of DIAL-TELEPHONE1.

([TELEPHONE [ACTOR (PERSON (NAME (JOHN]
(SUBJECT (PERSON (NAME (FRED]

[(MTRANS [ACTOR (PERSON (NAME (JOHN]
(TO (PERSON (NAME (FRED]

DIAL-TELEPHONE1)

Story input:

(* John told Fred that he had Mary.)

Story input:

Processing:

[MTRANS [ACTOR (PERSON (NAME (JOHN]

[TO (PERSON (NAME (FRED]

(MOBJECT (POSSESS [SUBJECT (PERSON (NAME (JOHN]

(OBJECT (SOCIAL-CONTROL

(SUBJECT (PERSON (NAME (MARY]

Adding SOCIAL-CONTROL1 to model in OBJECT bucket.

Adding POSSESS12 to model in STATE bucket.

Adding MTRANS3 to model in ACTION bucket.

Precondition of MTRANS3: CPATH

Trying to account for JOHN2's MTRANS3:

Primary effect of MTRANS3: BELIEF

Processing:

[BELIEF [SUBJECT (PERSON (NAME (FRED]

(OBJECT (POSSESS [SUBJECT (PERSON (NAME (JOHN]

(OBJECT (SOCIAL-CONTROL

(SUBJECT (PERSON (NAME (MARY]

Adding BELIEF5 to model in BELIEF bucket.

Primary inferences from BELIEF5:

Activating schema TELEPHONE:

Processing:

(AT (LOCATION (OF (RESTAURANT (NAME (TRENO'S]
Adding BELIEF6 to model in BELIEF bucket.

Primary inferences from BELIEF6:

Story input:

(* Fred delivered the money.)

Story input:

Processing:

(GIVE [ACTOR (PERSON (NAME (FRED]
[TO (PERSON (NAME (JOHN]
(OBJECT (MONEY)))

Adding GIVE2 to model in ACTION bucket.

Precondition of GIVE2: POSSESS
Trying to account for FRED2's GIVE2:
Primary effect of GIVE2: POSSESS

Processing:

(POSSESS [SUBJECT (PERSON (NAME (JOHN]
(OBJECT (MONEY)))

Adding POSSESS13 to model in STATE bucket.

Story input:

(* Mary arrived home in a taxi.)

Story input:

Processing:

[MOVE [ACTOR (PERSON (NAME (MARY]
[TO (LOCATION (OF (HOUSE (RESIDENT (PERSON (NAME (MARY]
(INSTRUMENT (CAR (OWNER (YELLOWCAB]

Adding HOUSE1 to model in OBJECT bucket.
Adding LOCATION8 to model in OBJECT bucket.
Adding YELLOWCAB1 to model in OBJECT bucket.
Adding CAR3 to model in OBJECT bucket.
Adding MOVE11 to model in ACTION bucket.

Precondition of MOVE11: POSSESS-SOCIAL-CONTROL

Processing:

[POSSESS-SOCIAL-CONTROL [SUBJECT (PERSON (NAME (MARY]
(OBJECT (PERSON (NAME (MARY]

Adding POSSESS-SOCIAL-CONTROL3 to model in STATE bucket.

(SHOOT NIL))
//ACTIVATION NIL)
[THREATEN //ISA ACTION //ROLES (ACTOR SUBJECT INSTRUMENT)
//PRECONDITIONS
((SPATH (SUBJECT ACTOR)
 (OBJECT SUBJECT))
 (POSSESS (SUBJECT ACTOR)
 (OBJECT INSTRUMENT)))
//PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS
[(BELIEF (SUBJECT SUBJECT)
 (OBJECT (IN-DANGER (SUBJECT SUBJECT)
 (FROM ACTOR])
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //SUGGESTS NIL
//ACTIVATION (((AIM (ACTOR ACTOR)
 (AT SUBJECT)
 (OBJECT INSTRUMENT))
 ((MTRANS (ACTOR ACTOR)
 (TO SUBJECT])
(AT //ISA STATE //ROLES (SUBJECT LOCATION)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY
(MOVE (ACTOR SUBJECT)
 (TO LOCATION))
//SUGGESTS NIL //ACTIVATION NIL)
(BELIEF //ISA BELIEF //ROLES (SUBJECT OBJECT)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(BUY //ISA ACTION //ROLES (ACTOR OBJECT FROM)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(CARES-FOR //ISA STATE //ROLES (SUBJECT OBJECT)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY
NIL //SUGGESTS NIL //ACTIVATION NIL)
(CONFINED //ISA ACTION //ROLES (ACTOR SUBJECT IN)
//PRECONDITIONS
((AT (SUBJECT ACTOR)
 (LOCATION IN))
 (AT (SUBJECT SUBJECT)
 (LOCATION IN)))
//PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL //SECONDARY-INFERENCES
NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL //SUGGESTS NIL
//ACTIVATION NIL)
(CPATH //ISA STATE //ROLES (SUBJECT OBJECT)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(DRIVE //ISA ACTION //ROLES (ACTOR PASSENGER VEHICLE TO)
//PRECONDITIONS
([AT (SUBJECT ACTOR)
 (LOCATION (LOCATION (OF VEHICLE]

[AT (SUBJECT PASSENGER)
(LOCATION (LOCATION (OF VEHICLE)
(POSSESS (SUBJECT ACTOR)
(OBJECT VEHICLE)))
//PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS ((AT (SUBJECT ACTOR)
(LOCATION TO))
(AT (SUBJECT PASSENGER)
(LOCATION TO)))
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(ESCAPE //ISA ACTION //ROLES (ACTOR FROM)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(GIVE //ISA ACTION //ROLES (ACTOR TO OBJECT)
//PRECONDITIONS
((POSSESS (SUBJECT ACTOR)
(OBJECT OBJECT)))
//PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS ((POSSESS (SUBJECT TO)
(OBJECT OBJECT)))
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(GOAL //ISA GOAL //ROLES (SUBJECT OBJECT)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(MOVE //ISA ACTION //ROLES (ACTOR FROM TO)
//PRECONDITIONS
((POSSESS-SOCIAL-CONTROL (SUBJECT ACTOR)
(OBJECT ACTOR))
(AT (SUBJECT ACTOR)
(LOCATION FROM)))
//PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS ((AT (SUBJECT ACTOR)
(LOCATION TO)))
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(MTRANS //ISA ACTION //ROLES (ACTOR TO MOBJECT)
//PRECONDITIONS
((CPATH (SUBJECT ACTOR)
(OBJECT TO)))
//PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS ((BELIEF (SUBJECT TO)
(OBJECT MOBJECT)))
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(PARENT //ISA STATE //ROLES (SUBJECT OBJECT)
//PRECONDITIONS NIL //PRIMARY-INFERENCES ((CARES-FOR (SUBJECT SUBJECT)
(OBJECT OBJECT))
(CARES-FOR (SUBJECT OBJECT)
(OBJECT SUBJECT)))
//PRIMARY-EFFECTS NIL //SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS
NIL //ACHIEVED-BY NIL //SUGGESTS NIL //ACTIVATION NIL)
(PERSON //ISA CHARACTER //ROLES (NAME)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL

```
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(POSSESS //ISA STATE //ROLES (SUBJECT OBJECT)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY
(BUY (ACTOR SUBJECT)
(OBJECT OBJECT))
//SUGGESTS NIL //ACTIVATION NIL)
(POSSESS-SOCIAL-CONTROL //ISA STATE //ROLES (SUBJECT OBJECT)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL
//PRIMARY-EFFECTS NIL //SECONDARY-INFERENCES NIL
//SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL //SUGGESTS
NIL //ACTIVATION NIL)
(RELEASE //ISA ACTION //ROLES (ACTOR OBJECT)
//PRECONDITIONS
((POSSESS-SOCIAL-CONTROL (SUBJECT ACTOR)
(OBJECT OBJECT)))
//PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS ((POSSESS-SOCIAL-CONTROL
(SUBJECT OBJECT)
(OBJECT OBJECT)))
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(SPATH //ISA STATE //ROLES (SUBJECT OBJECT)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS NIL
//SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL //ACHIEVED-BY NIL
//SUGGESTS NIL //ACTIVATION NIL)
(SUBSUME-GOALS //ISA STATE //ROLES (SUBJECT OF)
//PRECONDITIONS NIL //PRIMARY-INFERENCES NIL //PRIMARY-EFFECTS
NIL //SECONDARY-INFERENCES NIL //SECONDARY-EFFECTS NIL
//ACHIEVED-BY NIL //SUGGESTS NIL //ACTIVATION NIL))
```

6. Incremental Scene Interpretation

The interpretation of visual images is an important feature of a Natural Language (NL) system designed to interact with its physical environment. In a robotics system visual information can be used as a feedback path allowing the system to monitor its own actions and to detect the occurrence of unexpected events. Complete visual information about the objects in a scene is usually not available in any single image do to the partial or complete occlusion of one object by another. However, a time-series of images of a dynamic scene, or multiple views of a static scene, can be used to incrementally build up representations of the objects in the scene as more information becomes available from the processing of successive images. Thus, during object recognition, the system we are building will first formulate an initial hypothesis about the identity of a particular object based upon currently available information. Then, hypothesis refinement is performed as new information from successive images is added to the description of the object until the object's identity can be instantiated by the observed data from the scene.

The primary research effort in the development of the vision system focuses upon selecting an appropriate representation for the 3-dimensional (3-D) data acquired from a time-series of images, and the development of a schema-based strategy for constructing the object representations and performing object recognition from partial descriptions. The vision system constructs 3-D representations of the objects in the scene using range data obtained from a laser range-finder or from a pair of cameras arranged to

⁵This research is being carried out by Edward Altman, a graduate student, under the supervision of Prof. Narendra Ahuja.

provide stereo vision. All objects known to the system are stored in a database of models using a frame-based representation. To facilitate object matching, the models are indexed according to key features, such as surface shapes and relationships among object subparts. The objects found in the scene are also described in terms of key features which typify different classes of objects expected to be observed in the scene. After a set of features has been determined for an object in the scene, a discrimination net is used to select a schema to guide the more detailed processing of the object until an inconsistency is found or a reliable classification is achieved. The schemata provide detailed knowledge about expectations, plans, goals, procedures, and methods for evaluating the object currently under scrutiny.

7. Something or the Other

A speaker uses speech to affect the behavior and beliefs of other people. Through words he can perform speech acts such as making a request or giving a warning. Until the speech act value of a sentence has been determined, that sentence has not been fully understood. For this reason a theory of speech acts must play an important part in any comprehensive natural language understanding system. Such a system should be able to look beyond the literal meaning of a sentence to determine what sort of action is being performed through the use of that sentence, and should also be able to decide whether or not that action has been successfully completed. To date very little has been done to develop a model of speech acts in artificial intelligence. The work that has been done deals with very limited sets of speech acts, and operates within very narrow domains. The goal of this research is to develop a more complete model of speech acts which can be implemented as part of a general natural language system.

We are currently working on identifying and classifying the different kinds of actions which can be performed through the use of speech. Although several broad classification schemes for speech acts have been suggested in the past [Austin62,Searle76] these schemes are too general to be useful for artificial intelligence systems. Starting from the categories laid out by Austin and Searle we are developing a more detailed taxonomy which can be used as a basis for the representation of speech act knowledge in a natural language processor. Such a taxonomy will not only provide criteria for identifying speech acts in a text or dialogue, but will also help guide a

⁶This research is being carried out by Patricia Halko, a graduate student, under the supervision of Prof. David L. Waltz.

system's reasoning and inferencing processes. Once a speech act has been recognized and classified, information connected with specific classes of acts can be used to make inferences about the speaker, her plans, goals and intentions, and the relationship between the speaker and hearer(s). Predictions can also be made about the future behavior of the hearer(s) and speaker, and the content of further dialogue between them.

8. References

- [Austin62] J. L. Austin, How to Do Things with Words, Oxford Univ. Press, 1962.
- [Bobrow75] D. Bobrow and D. Norman, "Some Principles of Memory Schemata," in Representation and Understanding, D. Bobrow and A. Collins (ed.), Academic Press, New York, NY, 1975.
- [Bobrow77] D. Bobrow, R. Kaplan, M. Kay, D. Norman, H. Thompson and T. Winograd, "GUS, A Frame Driven Dialog System," Artificial Intelligence 8, 1 (1977), .
- [Cross83] S. E. Cross, "Qualitative Reasoning in an Expert System Framework," T-124, Coordinated Science Lab, Urbana, IL, May 1983.
- [DeJong79] G. DeJong, "Skimming Stories in Real Time: An Experiment in Integrated Understanding," Research Rept. 158, Yale Computer Science Department, New Haven, CT, 1979.
- [DeJong82a] G. F. DeJong and D. L. Waltz, "Understanding Novel Language," Working Paper-36, Coordinated Science Lab, Univ. of Illinois, Urbana, IL, July 1982.
- [DeJong82b] G. DeJong, "Automatic Schema Acquisition in a Natural Language Environment," Proc. National Conf. on Artificial Intelligence, Pittsburgh, PA, August 1982, 410-413.
- [Forbus81] K. Forbus and A. Stevens, "Using Qualitative Simulation to Generate Explanations," 4490, Bolt Beranek and Newman, Cambridge, MA, March 1981.
- [Forbus82] K. Forbus, "Qualitative Process Theory," Artificial Intelligence Memo 664, MIT Artificial Intelligence Lab, Cambridge, MA, February 1982.
- [Group80] G. A. E. Group, Aircraft Gas Turbine Guide, General Electric, Cincinnati, OH, October 1980.
- [Kleer79] J. Kleer, "The Origin and Resolution of Ambiguities in Casual Arguments," Proc. International Joint Conference on Artificial Intelligence-79, Tokyo, Japan, 1979, 197-203.
- [Lenat80] D. B. Lenat, "The Nature of Heuristics," HPP-80-26, Stanford University, Stanford, CA, December 1980.
- [Minsky75] M. L. Minsky, "A Framework for Representing Knowledge," in The Psychology of Computer Vision, P. Winston (ed.), McGraw-Hill, New York, NY, 1975, 211-277.
- [Schank77] R. C. Schank and R. Abelson, Scripts, Plans, Goals, and Understanding, Lawrence Erlbaum Assoc., Hillsdale, NJ, 1977.
- [Searle76] J. R. Searle, "A Taxonomy of Illocutionary Acts," in Language, Mind and Knowledge, K. Gunderson (ed.), Univ. of Minnesota Press, 1976.
- [Spoor83] D. T. Spoor, "A System for Reasoning about Time and Events," M.S. thesis, Electrical Engineering Dept., Univ. of Illinois, Urbana, IL, 1983.

- [Teitelman78] W. Teitelman, Interlisp Reference Manual, Xerox PARC, Palo Alto, 1978.
- [Treager79] I. E. Treager, Aircraft Gas Turbine Engine Technology, Second Edition, McGraw-Hill, New York, NY, 1979.
- [Waltz82] D. L. Waltz, R. T. Chien and G. DeJong, "An Expert Distributed Robotics System with Comprehension and Learning Abilities in the Aircraft Flight Domain," T-116, Coordinated Science Lab, Univ. of Illinois, Urbana, IL, July 1982.
- [Wilensky81] R. Wilensky, "Meta-Planning: Representing and Using Knowledge About Planning in Problem Solving and Natural Language Understanding," Cognitive Science 5, (December 1981), 197-233.

9. Publications

Debrunner, C., "A Two Dimensional Activation Cell," WP-46, Coordinated Science Laboratory, Univ. of Illinois, Urbana (December 1982).

DeJong, G., "Acquiring Schemata Through Understanding and Generalizing Plans," Proc. of the 8th Intern. Joint Conf. on Artificial Intelligence, Karlsruhe, West Germany (1983).

DeJong, G., "An Approach to Learning from Observation," Proc. of the 1983 Intern. Workshop on Machine Learning, Univ. of Illinois at Urbana-Champaign (June 1983).

DeJong, G., "Artificial Intelligence Implications for Information Retrieval," Proc. of the 6th Annual Intern. ACM SIGIR Conf., Washington, D.C. (June 1983).

DeJong, G. and Waltz, D., "Understanding Novel Language," The International Journal of Computers and Mathematics with Applications, Vol. 9, No. 1, pp. 131-147 (1983).

Maran, L. R., Spoor, D. T., and Waltz, D. L., "Encoding the Natural Language Meaning of Time - Toward a Conceptual Model for Temporal Meaning," WP-37, Coordinated Science Laboratory, Univ. of Illinois, Urbana (May 1983).

Segre, A. M., Sherwood, B. A., and Dickerson, W. B., "An Expert System for the Production of Phoneme Strings from Unmarked English Text Using Machine Induced Rules," Proc. 1st Conf. of the European Chapter of the Assoc. for Computational Linguistics, Pisa, Italy (September 1983).

Waltz, D. L., "Helping Computers Understand Natural Languages," IEEE Spectrum, Vol. 20, No. 11, pp. 81-84 (November 1983).

Waltz, D. L., and Dorfman, M. H., "The Holes in Points," The Behavioral and Brain Sciences, Vol. 6, No. 4, pp. 612-613 (December 1983).

Waltz, D., Genesreth, M., Hart, P., Hendrix, G., Joshi, A., McDermott, J., Mitchell, T., Nilsson, N., Wilensky, R., and Woods, W., "Artificial Intelligence: An Assessment of the State-of-the-Art and Recommendation for Future Directions," AI Magazine, Vol. 4, No. 3, pp. 55-67 (Fall 1983); also appears in Unmanned Systems, Vol. 2, No. 2, pp. 19-30 (Fall 1983).

Waltz, D. L., and Pollack, J., "Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation," invited for special issue of Cognitive Science (1984).

10. Personnel

There have been four investigators on this project: Professors David Waltz (principal investigator), Gerald DeJong, Narendra Ahuja and R.T. Chien. We regret to report that Professor R.T. Chien died in December 1983, after being ill for the preceding several months. Nine graduate students have contributed to the project: David Spoor, Raman Rajagopalan, and Patricia Halko (advisees of Professor Waltz), Alberto Segre, Ashwin Ram, Christian Debrunner and Paul Harrington (advisees of Professor DeJong), Shaun Keller (advisee of Professor Chien, now being advised by Professor Waltz), and Edward Altman (advisee of Narendra Ahuja). Paul Harrington received his M.S. degree in May 1983. Christian Debrunner has switched to another research area.

11. Inventions and Patent Disclosures

There are no inventions or patent disclosures to report.