Natural Language in Four Spatial Interfaces

Kenneth Wauchope, Stephanie Everett, Dennis Perzanowski, Elaine Marsh
Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory, Code 5512
4555 Overlook Avenue, S.W.
Washington, DC 20375-5337, USA
[wauchope|severett|dennisp|marsh]@aic.nrl.navy.mil

Abstract

We describe our experiences building spoken language interfaces to four demonstration applications all involving 2- or 3-D spatial displays or gestural interactions: an air combat command and control simulation, an immersive VR tactical scenario viewer, a map-based air strike simulation tool with cartographic database, and a speech/gesture controller for mobile robots.

1 Introduction

The NAUTILUS natural language processor has been under development at our facility since about 1988. During those years it has been used and tested in five different demonstration projects, four of which we describe in some detail in this report: an air combat command and control simulation, an immersive VR tactical scenario viewer, a map-based air strike simulation tool with cartographic database, and a speech/gesture controller for mobile robots. All four applications involve spatial displays or interactions, including 2D map-based graphical displays (radar screen, geographic map), 3D perspective scenes, and hand gesture input.

2 NAUTILUS

NAUTILUS is built around an early version of the PROTEUS chart parser from New York University's Courant Institute (Grishman, 1986). The three subsequent system components were developed at our own facility.

2.1 PROTEUS

PROTEUS syntactic grammars consist primarily of two rule types: context-free rules written in BNF notation, and restriction rules written in a high-level algorithmic language. Each restriction rule is attached to a particular nonterminal in the right-hand side of one or more context-free rules. During parsing the restriction fires immediately after that nonterminal has been constructed, testing the subtree at that point for well-formedness, or attaching an attribute value for use later on. Nodes rejected by a restriction are not added to the active chart and so do not contribute to the remainder of the parse.

2.2 TINSEL

One of the attributes composed by PROTEUS during parsing is an operator-operand regularized form intended to serve as the representation to which semantic selection and interpretation rules can be applied. The TINSEL semantic interpreter (Wauchope, 1990) applies case-frame rules and selection restrictions to the PROTEUS regularized output. The interpreter can either be invoked post-parse (applied top-down to each candidate sentential regularization) or interleaved with PROTEUS, testing each individual clausal or noun phrase immediately upon construction. In interleaved mode, if a node's regularization does not pass the case-frame or selection criteria then the node is not added to the chart, which can prune the search space and reduce parsing time considerably. If the node does pass selection, its regularization is augmented with the relevant semantic class and role information, becoming an intermediate semantic representation suitable for further processing such as reference resolution and quantifier scoping.

The TINSEL interpreter is primarily model-driven, which is to say that the case frame behavior of each predicate in the domain must be explicitly encoded in a declarative semantic representation. As a result we have not attempted to incorporate any generalized case-frame rules into the interpreter itself, so TINSEL is not bound to any particular theory of thematic relations, giving the system devel-

oper maximum flexibility in devising useful semantic representations. TINSEL does contain some general rules for handling noun phrases, however, such as automatically attempting to interpret certain prepositional phrases as implicit BE-verb relative clauses (*the hammer on the table* ⇒ *the hammer which is on the table*), etc.

2.3 FOCAL

The FOCAL (FOCus ALgorithm) reference resolution module was developed by visiting MIT graduate student Gina-Anne Levow. It resolves definite, indefinite, and pronominal references as subsets of objects from a closed-world model developed for each application. Model objects have a TINSEL semantic class attribute, permissable identifying specifiers (*S.S. Loveboat*, waypoint *No. 2*, *NTDS* icons), and a marker indicating if the object represents a collection of unindividualized entities (*map rings*, *aircraft trails*). FOCAL uses semantic class, number, recency, and constituent order within the sentence when choosing antecedents for anaphoric references. It assumes demonstratives (*that fighter*) to be anaphoric and attempts to resolve other definite references (*the fighters*) first as anaphoric and then as universal. Since none of our interfaces to date has involved declarative or hypothetical utterances, indefinite expressions (*a fighter*) are interpreted strictly as closed-world references, i.e., one of the known fighters.

2.4 FUNTRAN

The FUNTRAN (FUNctional TRANslator) module takes TINSEL and FOCAL output and constructs a quantified logical form suitable for evaluation in the runtime environment to issue a command or query to the target application. The logical quantifiers and connectives (FORALL, EXISTS, NOT, AND, etc.) have generic procedural definitions as Lisp macros, so the system developer just needs to develop a so-called Translation Function (TF) for each of the TINSEL predicates in the domain. TFs are Lisp functions (defuns) of the same name as the predicate, taking keyword arguments corresponding to each of the predicate's semantic slots, and exchanging appropriately coded information with the target application via so-called Interface Functions, described next. FUNTRAN also composes simple fragmentary English responses to database queries based on the results of the TF predicate evaluation.

2.5 Back End Translator

At this point the generic NAUTILUS code ends and the system developer must hand-craft an application-specific interface layer between the Translation Functions and the target. The nature of that interface depends on whether NAUTILUS and the target are running in the same Lisp process or as separate Unix processes, possibly on different machines. In one of our projects (InterLACE), the target application is just a Lisp program running in the same process space as the NLP system, so the primitive Interface Functions (IFs) for communicating between the two are just Lisp function calls. In two others (InterVR and InterROB), the target application runs on another Unix machine on the local net, so the IFs on the NAUTILUS side must encode and transmit message strings over an IPC socket to a corresponding decoder layer linked into the application. In the fourth project (Eucalyptus) we developed versions for both approaches, one where the application object code (compiled from C) is loaded into Lisp and the IFs are foreign function calls, and the other doing IPC message passing.

2.6 Speech I/O

For speech input we use the Phonetic Engine (PE200) from Speech Systems Inc. with the speech recognition software running on a Sun workstation to which the PE200 hardware is connected by a serial line. Under various circumstances we have linked the software in with either NAUTILUS or the application, or have run it as a separate process communicating with NAUTILUS via an IPC socket. For speech output, a DECtalk speech synthesizer is connected to the other Sun serial port and can be sent output from NAUTILUS either by Unix system calls or by writing data directly to the port.

3 Application Projects

3.1 Eucalyptus

NAUTILUS was first used in the Eucalyptus (Wauchope, 1994) spoken language interface to the KOALAS Airborne Early Warning C2 simulation (Barrett and Aldrich, 1990). The original KOALAS interface consisisted of a mouse-sensitive simulated radar screen with a conventional graphical user interface composed of command pushbuttons, dialog boxes and scrolling display windows. Our objective in Eucalyptus was to make the same command and data access functionality available via natural language, integrated as much as possible with the graphical interface to allow multimodal interactions. For example, a NL command to the system might result in the display of the same dialog box used in the corresponding GUI command, but with the dialog's data fields fully or partially filled from the NL input;

the user can then fill in any remaining empty fields and issue final acceptance either graphically or verbally. To that end, the command-oriented Interface Functions in Eucalyptus consist largely of calls to the base functions underlying the KOALAS GUI.

Eucalyptus also includes deictic reference, allowing the user to click on one or more radar blips or screen locations while speaking verbal references like *this fighter* or *these CAP stations*. When a mouse click occurs, NAUTILUS asks the application for the identities of all the objects located at or near the mouse event, and then takes the subset of those objects that match the semantics of the verbal phrase (which can be determined from predicate context as well: for example the word *here* in *Have fighter 1 refuel here* necessarily refers to a tanker aircraft). To avoid the problems of time-correlating speech with graphical input and distinguishing anaphora from deixis, we reserve the words *this*, *these* and *here* for deictic reference and *that*, *those* and *there* for anaphoric reference, and allow no more than one plural deictic reference per utterance.

Database query is used both in answering explicit interrogatives (*Which fighters aren't holding CAP station?*) as well as dereferencing qualified NPs (*moving aircraft*). The system can also interpret NP sentence fragments as followup commands or queries by substituting the NP into the semantically relevant slot of the prior utterance's logical form.

As originally designed, FOCAL expected a closed-world model of all domain objects to be available at startup time. This had to be modified somewhat in Eucalyptus since the KOALAS world includes hypothetical objects (suspected threat aircraft) which the user and system can create and destroy at will. References to hypothetical entities are resolved by having FOCAL dynamically consult the application database for the current object population at the time the phrase was uttered.

The core syntactic grammar of about 150 context-free rules and 50 restriction rules developed for Eucalyptus has been re-used in all the other NAUTILUS projects. Each one has augmented it with a few dozen additional rules for handling application-specific constructs like *station two sector one* (Eucalyptus), *latitude forty degrees north longitude ninety five degrees west* (InterVR), *the town of Leipzig* (InterLACE), and *thirty degrees left* (InterROB).

The Eucalyptus lexicon totals about 425 words, many of them unused morphological variants generated automatically by the PROTEUS lexical macros; by comparison, the vocabulary for the speech recognition front end is only 260 words. Total input coverage is on the order of 100 million utterances, deliberately high to test the speech system's ability to detect a wide variety of noun phrase determiners. Unlike the NAUTILUS grammar, the speech grammar excludes iteration and recursion (such as compounding) to maintain a reasonable level of recognition accuracy. An experimental addition of relative clauses to the speech grammar was "productive" only in the linguistic sense, since the resulting exponential increase in grammar size caused recognition rates to drop to unacceptable levels.

3.2 InterVR

We next used NAUTILUS in InterVR (Everett et al., 1994), a spoken language controller for an immersive 3D or "Virtual Reality" tactical combat simulation viewer. Here the emphasis was on the utility of speech I/O in an eyes- and hands-busy virtual environment. Like Eucalyptus, InterVR supports commands, queries, complex reference and anaphora, and NP followups. A non-immersive desktop version of the viewer allowed mouse selection of a platform and thus singular deictic reference (*this helicopter*), but the immersive display version did not include a dataglove or other pointing device. We did not have time or resources to tackle the problem of resolving referents based on visual context (for example having *that helicopter* refer to the one nearest the center of the user's field of view), but we are currently investigating the interaction of vision and language in the InterROB project, to be discussed shortly.

The InterVR speech component has a vocabulary comparable in size to that of Eucalyptus but a more constrained input range (about 1 million utterances) mainly due to a less liberal variety of NP determiners. The IPC code developed for Eucalyptus ported immediately to the new application, and NAUTILUS's modular architecture allowed speech modeling, NLP knowledge base development and IF coding to be pursued independently by different team members with a minimum of coordination.

3.3 InterLACE

InterLACE (Wauchope, 1996) is an integrated natural language interface and graphical map display for the Air Force's LACE land/air combat simulation system (Anken, 1989). LACE includes a large object-oriented cartographic database of most of central Germany, containing a total of over 12,000 objects such as towns, lakes, rivers, and railroads. Since the application is written in Common LISP and so can run in the same process environment as NAUTILUS, we dispensed with independently modeling FOCAL entities for the domain and just let the

LACE database objects serve as the extensions of referring expressions. To avoid having to enter hundreds of foreign proper names into the PROTEUS lexicon, we modified the PROTEUS lexical tagger to assume that any input word might be a proper name (applying that assumption only to non-English words failed the first time we encountered the river Main and the Czech towns of Most and As). Deictic reference operates similarly to Eucalyptus: a mouse click can select a number of overlapping map objects at once, to be resolved by an accompanying verbal reference; for example *What's the population here?* would resolve to a town object whereas *Does this cross the Elbe?* might resolve to a road.

The InterLACE domain necessitated extending FUNTRAN to generate proper logical forms for spatial comparatives and superlatives (*Is Wurzen closer than Grimma?, the closest lake to Eilenburg*) and implicit and explicit reflexives (*Do these roads cross [each other]?*), and also introduced direct address (*Tank 1, head north*) and iterative arguments to navigation commands (*Head north on road E2 for 2 km to Wurzen*). An experimental study of NL inputs from novice InterLACE users showed that of 822 inputs, 14 contained typos or misspellings and 30 contained ungrammaticalities, for an illformedness rate of 5%. Of the remaining 778 utterances NAUTILUS failed to understand 23 (3%) due to incomplete coverage.

Since the PE200's phonetic rules are for American English and (unlike PROTEUS) the module cannot be tricked into recognizing unknown inputs as possible proper names, a complete speech input component for InterLACE was impractical. For demo purposes we opted to implement a 160-word speech interface containing just fifty German proper names, few enough that the recognizer doesn't have too much trouble distinguishing them, for an input coverage of about 10 million utterances. Similarly for speech output we provided the same vocabulary to DECtalk along with phonetic transcriptions to produce acceptable German pronunciations.

3.4 InterROB

InterROB is a new project exploring the integration of spoken and gestural inputs to a pair of mobile robots with rangefinder vision capability. To date the system accepts commands only, using a 66-word speech vocabulary with an input range of about 11,000 utterances. The robots currently recognize two types of gesture: distance (hands held apart) and direction (wave left/right). Since the system does not yet query information from the robots, deictic reference must currently be resolved on the robot side rather than (as in the systems described earlier) by having NAUTILUS choose from a set of candidates provided by the application. This means that phrases like *that waypoint* or *the waypoint over there* must be assigned a special FOCAL extension (a pseudo-object called a gesture-waypoint) which is not one of the four actual waypoint objects in the closed world, whereas with query capability NAUTILUS might be able to obtain enough information from the robot to determine which of the four actual waypoints is being gestured toward.

Another goal in InterROB is to go beyond the usual restriction of deictic reference to demonstrative or indexical references (*that, here, there*) and allow gestures to accompany any sort of definite or indefinite NP. This could then be extended to include extralinguistic context in general, such as interpreting *the waypoint* to mean the one the robot is currently facing, or *my right* to mean 90 degrees perpendicular to the way the robot perceives the operator to be facing. We also plan to extend the robotic vision capabilities with additional hard/software to allow visual object recognition for lexical acquisition.

References

Ralph Grishman. 1986. PROTEUS parser reference manual. PROTEUS Project Memorandum #4, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, July.

Kenneth Wauchope. 1994. Eucalyptus: integrating natural language input with a graphical user interface. NRL technical report NRL/FR/5510–94-9711, February.

Chris Barrett and Charles Aldrich. 1990. Final report: KOALAS test planning tool concept demonstration: users manual. Los Alamos National Laboratory, Los Alamos, New Mexico.

Craig S. Anken. 1989. LACE: land air combat in ERIC. Rome Air Development Center RADC-TR-89-219, October.

Kenneth Wauchope. 1990. A tandem semantic interpreter for incremental parse selection. NRL technical report 9288, September.

Stephanie S. Everett, Kenneth Wauchope, and Manuel A. Pérez. 1994. A natural language interface for virtual reality systems. NCARAI technical report AIC-94-046.

Kenneth Wauchope. 1996. Multimodal interaction with a map-based simulation system. NCARAI technical report AIC-96-019.