

Taxonomy, Descriptions, and Individuals
in Natural Language Understanding

Ronald J. Brachman
Bolt Beranek and Newman Inc.

KLONE is a general-purpose language for representing conceptual information. Several of its prominent features -- semantically clean inheritance of structured descriptions, taxonomic classification of generic knowledge, intensional structures for functional roles (including the possibility of multiple fillers), and procedural attachment (with automatic invocation) -- make it particularly useful in computer-based natural language understanding. We have implemented a prototype natural language system that uses KLONE extensively in several facets of its operation. This paper describes the system and points out some of the benefits of using KLONE for representation in natural language processing.

Our system is the beneficiary of two kinds of advantage from KLONE. First, the taxonomic character of the structured inheritance net facilitates the processing involved in analyzing and responding to an utterance. In particular, (1) it helps guide parsing by ruling out semantically meaningless paths, (2) it provides a general way of organizing and invoking semantic interpretation rules, and (3) it allows algorithmic determination of equivalent sets of entities for certain plan-recognition inferences. Second, KLONE's representational structure captures some of the subtleties of natural language expression. That is, it provides a general way of representing exactly the quantificational import of a sentence without over-committing the interpretation to scope or multiplicity not overtly specified.

The paper first presents a brief overall description of the natural language system. Then, prior to describing how we use KLONE in the system, we discuss some of the language's features at a general level. Finally we look in detail at how KLONE affords us the advantages listed above.

1. THE TASK AND THE SYSTEM

Generally speaking, we want to provide a natural interface to a subsystem that knows how to present conceptual information intelligently (on a bit-map display) -- in this case the Augmented Transition Network (ATN) grammar from the LUNAR system [5]. The information presentation subsystem allows flexible specification of coordinate system mappings, including rectangular windows, from parts of the ATN onto a sequence of "view surfaces". Object types can be assigned arbitrary presentation forms (graphic or alphanumeric), which can be modified in particular cases. Parts of the grammar are displayed according to standing orders and special requests about shape and projection.

Our task is to command and control the intelligent graphics subsystem through natural language. For example, a sample dialogue with the system might include this sequence of utterances:

- (1) Show me the clause level network.
[System displays states and arcs of the S/ network]
- (2) Show me S/NP.
[System highlights state S/NP]
- (3) Focus in on the preverbal constituents.
[System shifts scale and centers the display on the

preverbal states]

- (4) No. I want to be able to see S/AUX.
[System "backs off" display so as to include state S/AUX]

At the same time, we would like to ask factual questions about the states, arcs, etc. of the ATN (e.g. "What are the conditions on this <user points> arc?"). Questions and commands addressed to the system typically (1) make use of elements of the preceding dialogue, (2) can be expressed indirectly so that the surface form does not reflect the real intent, and (3) given our graphical presentation system, can make reference to a shared non-linguistic context. The issues of anaphora, (indirect) speech acts, and deixis are thus of principal concern.

The natural language system is organized as illustrated in Figure 1*. The user sits at a bit-map terminal

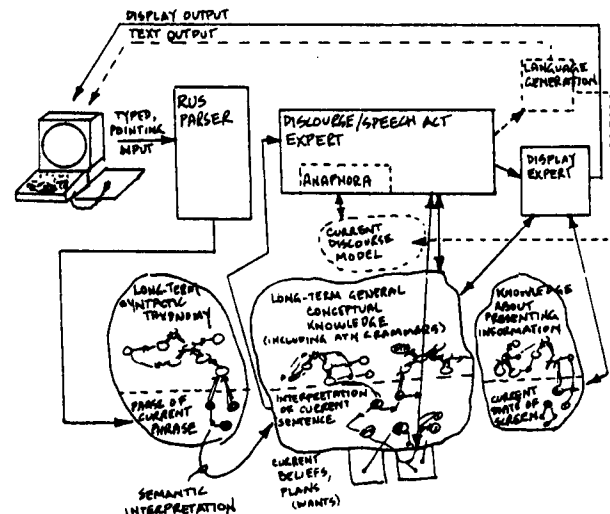


Figure 1. System structure
(highlighting types of knowledge involved).

equipped with a keyboard and a pointing device. Typed input from the keyboard (possibly interspersed with coordinates from the pointing device) is analyzed by a version of the RUS System [2] -- an ATN-based incremental parser that is closely coupled with a "case-frame dictionary". In our system, this dictionary is embodied in a syntactic taxonomy represented in KLONE. The parser produces a KLONE representation of the syntactic structure of an utterance. Incrementally along with its production, this syntactic structure triggers the creation of an interpretation. The interpretation structure -- the literal (sentential) semantic content of the utterance -- is then processed by a discourse expert that attempts to determine what was really meant. In this process, anaphoric expressions must be resolved and indirect speech acts recognized. Finally, on the basis of what is determined to be the intended force of

* Dashed elements of the figure are proposed but not yet implemented.

displayForm associated with it. The subnetwork below ARC expresses the classification of the various types of arcs in the ATN and how their conceptual structures vary. For example, a CONNECTING-ARC has a nextState (the state in which the transition leaves the parsing process), while for POP-ARCs the term is not meaningful (i.e. there is no nextState Role). Links that connect the Roles of more specific Concepts with corresponding Roles in their parent Concepts are considered to travel through the appropriate Cables. Finally, the structure of an Individual Concept is illustrated by CATARC#0117. Each IRole expresses the filling of a Role inherited from the hierarchy above -- because CATARC#0117 is a CAT-ARC, it has a category; because it is also a CONNECTING-ARC, it has a nextState, etc.

The structure of a Concept is completed by its set of Structural Descriptions (SD's). These express how the Roles of the Concept interrelate via the use of parameterized versions ("ParaIndividuals") of other Concepts in the network to describe quantified relations between the ultimate fillers of the Concept's Roles. The quantification is expressed in terms of set mappings between the RoleSets of a Concept, thereby quantifying over their sets of fillers. In addition to quantified relations between potential Role fillers, simple relations like subset and set equality can be expressed with a special kind of SD called a "RoleValueMap" (e.g. the relation that "the object of the precondition of a SEE is the same as the object of its effect"). SD's are inherited through cables and are particularized in a manner similar to that of Roles.

There is one important feature of KLONE that I would like to point out, although it is not yet used in the natural language system. The language carefully distinguishes between purely descriptive structure and assertions about coreference, existence, etc. All of the structure mentioned above (Concepts, Roles, SD's and Cables) is definitional. A separate construct called a Nexus is used as a locus of coreference for Individual Concepts. One expresses coreference of description relative to a Context by placing a Nexus in that Context and attaching to it Individual Concepts considered to be coreferential. All assertions are made relative to a Context, and thus do not affect the (descriptive) taxonomy of generic knowledge. We anticipate that Nexuses will be important in reasoning about particulars, answering questions (especially in deciding the appropriate form for an answer), and resolving anaphoric expressions, and that Contexts will be of use in reasoning about hypotheticals, beliefs, and wants.

The final feature of KLONE relevant to our particular application is the ability to attach procedures and data to structures in the network. The attached procedure mechanism is implemented in a very general way. Procedures are attached to KLONE entities by "interpretive hooks" (ihooks), which specify the set of situations in which they are to be triggered. An interpreter function operating on a KLONE entity causes the invocation of all procedures inherited by or directly attached to that entity by ihooks whose situations match the intent of that function. Situations include things like "Individuate", "Modify", "Create", "Remove", etc. In addition to a general situation, an ihook specifies when in the execution of the interpreter function it is to be invoked (PRE-, POST-, or WHEN-).

3. USE OF KLONE IN THE NATURAL LANGUAGE SYSTEM

The previous section described the features of KLONE in general terms. Here we illustrate how they facilitate the performance of our natural language system. (Figure 1 above sketched the places within the system of the various KLONE knowledge bases discussed here.) We will

discuss the use of a syntactic taxonomy to constrain parsing and index semantic interpretation rules, and structures used in the syntactic/discourse interface to express the literal semantic content of an utterance.

The parser uses KLONE to describe potential syntactic structures. A taxonomy of syntactic constituent descriptions, with Concepts like PHRASE, NOUN-PHRASE, LOCATION-PP, and PERSON-WORD, is used to express how phrases are built from their constituents. The taxonomy also serves as a discrimination net, allowing common features of constituent types to be expressed in a single place, and distinguishing features to cause branching into separate subnets.

Two benefits accrue from this organization of knowledge. First, shallow semantic constraints are expressed in the Roles and SD's of Concepts like LOCATION-PP. For example, the preObject of a LOCATION-PP must be a PLACE-NOUN. A description of "on AI" (as in "book on AI") as a LOCATION-PP could not be constructed since AI does not satisfy the value restriction for the head role. Such constraints help rule out misleading parse paths, in the manner of a semantic grammar [4], by refusing to construct semantically anomalous constituent descriptions. In conjunction with the general (ATN) grammar of English, this is a powerful guidance mechanism which helps parsing proceed close to deterministically [2].

Second, the syntactic taxonomy serves as a structure on which to hang semantic projection rules. Since the taxonomy is an inheritance structure, the description of a given syntactic constituent inherits all semantic interpretation rules appropriate for each of the more general constituent types that it specializes, and can have its own special-purpose rules as well. In the example above, simply by virtue of its placement in the taxonomy, the Concept for "on AI" would inherit rules relevant to PP's in general and to SUBJECT-PP's in particular, but not those appropriate to LOCATION-PP's. Interpretation per se is achieved using the attached procedure facility, with semantic projection rules expressed as functions attached to Roles of the syntactic Concepts. The functions specify how to translate pieces of syntactic structure into "deeper" Concepts and Roles. For example, the subject of a SHOW-PHRASE might map into the agent of a DISPLAY action.

The mapping rules are triggered automatically by the KLONE interpreter. This is facilitated by the interpreter's "pushing down" a Concept to the most specific place it can be considered to belong in the taxonomy (using only "analytic", definitional constraints). Figure 3 illustrates schematically the way a Concept can descend to the most specific level implied by its internal description. The Concept being added to the network is an NP whose head is "ARC" and whose modifier is "PUSH" (NP#0023). It is initially considered a direct (Generic) subConcept of the Concept for its basic syntactic type (NP). Its Role structure, however, implies that it in fact belongs in a more restricted subclass of NP's, that is, TYPED-ARC-NP (an NP whose head is an ARC-NOUN and whose modifier is an ARC-TYPE-WORD). The interpreter, on the basis of only definitional constraints expressed in the network, places the new Concept below its "most specific subsumer" -- the proper place for it in the taxonomy. The process proceeds incrementally, with each new piece of the constituent possibly causing further descent. In this case, NP#0023 would initially only have its head Role specified, and on that basis, it would be placed under ARC-NP (which is "an NP whose head is an ARC-NOUN"). Then the parser would add the modifier specification, causing the Concept's descent to the resting place shown in the right half of Figure 3. When the constituent whose description is being added to the network is "popped" in the parser, its KLONE description

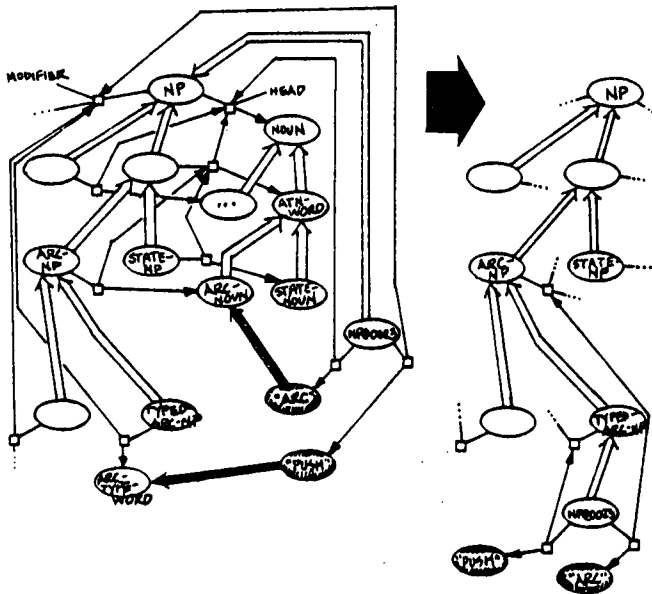


Figure 3. Automatic Concept descent.

is individuated -- causing the invocation of all "WHEN-Superdividuated" attached procedures inherited through superConcept Cables. These procedures cause an interpretation for the constituent to be built on the basis of the interpretations of component parts of the syntactic description.

This literal semantic interpretation of a phrase -- also a KLONE structure -- is the "input" to the discourse component. An important element of this interface between the syntactic processor and the discourse component is that the parser/interpreter commits itself only to information explicitly present in the input phrase, and leaves all inference about quantifier scope, etc. to the discourse expert. Two kinds of representational structures support this. The Concept DSET (for "determined set") is used extensively to capture sets implicit in noun phrases and clauses. DSETs use the inherent multiplicity of RoleSets to group together several entities under a single Concept, and associate determiners (definite/indefinite, quantifiers, etc.) with such a set of entities. A DSET can express the characteristics of a set of entities without enumerating them explicitly, or even indicating how many members the set is expected to have. RoleValueMaps allow constraints between DSETs to be expressed in a general way -- a RoleValueMap expresses a subset or equality relation between two RoleSets. Such relations can be constructed without knowing in advance the cardinality of the sets or any of their members.

Figure 4 illustrates the use of these structures to express the intent of the sentence, "Show me states S/NP, S/AUX, and S/DCL". DSET#0035 represents the interpretation of the noun phrase, "the states S/NP, S/AUX, and S/DCL". The generic DSET Concept has two Roles, member and determiner. The member Role can be filled multiply, and therein lies the "settedness" of the DSET. DSET#0035 has a particularized version of the

* RoleSets in this figure are drawn as squares with circles around them. RoleSets with filled-in circles are a special kind of particularized RoleSet that can occur only in Individual Concepts. The RoleValueMap is pictured as a diamond.

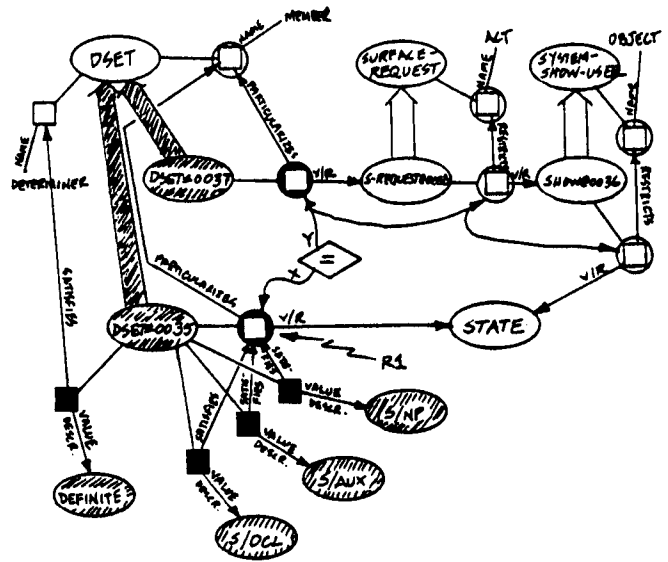


Figure 4. KLONE description of "Show me states S/NP, S/AUX, and S/DCL".

member Role: Role R1 represents the set of three states mentioned in the noun phrase, as a group. Thus, the Value Restriction of R1, STATE, applies to each member. The three IRoles of DSET#0035, connected by "Satisfies" links to the particularized member RoleSet, indicate that the particular states are the members of the set*.

The other DSET in the figure, DSET#0037, represents the clause-level structure of the sentence. The clause has been interpreted into something like "the user has performed what looks on the surface to be a request for the system to show the user some set of states". This captures several kinds of indeterminacy: (1) that the sentence may only be a request at the surface level ("Don't you know that pigs can't fly?" looks like a request to inform), (2) that there is more than one way to effect a "show" ("show" could mean redraw the entire display, change it slightly to include a new object, or simply highlight an existing one), (3) that it is not clear how many operations are actually being requested (showing three objects could take one, two, or three actions). Therefore, the interpretation uses Generic Concepts to describe the kind of events appearing in the surface form of the sentence and makes no commitment to the number of them requested. The only commitment to "quantificational" information is expressed by the RoleValueMap. Its two pointers, X (pointing to the member Role of DSET#0035) and Y** (pointing to the object of

* The Value Restriction, STATE, is redundant here, since the members of this particular set were explicitly specified (and are known to be states). In other cases, the information is more useful. For example, no IRoles would be constructed by the parser if the sentence were "Are there three states?"; only one would be constructed in "Show me state S/NP and its two nearest neighbors". On the other hand, no Value Restriction would be directly present on Role R1 if the noun phrase were just "S/NP, S/AUX, and S/DCL".

** Y is a chained pointer going first through the member Role of DSET#0037, then through the act Role of S-REQUEST#0038, and finally to the object Role of SHOW#0036. It is considered to refer to the set of IRoles expressing the objects of all SHOW events ultimately S-REQUESTed, when it is determined exactly how many there are to be (i.e. when the IRoles of

the requested act), indicate that the ultimate set of things to be shown, no matter how many particular SHOW events take place, must be the same as the set of members in the noun phrase DSET (namely, the three states).

As mentioned, semantic interpretation invokes the discourse expert. This program looks to a plan that it is hypothesizing its user to be following in order to interpret indirect speech acts. Following [1], the speech acts REQUEST, INFORM, INFORMREF, and INFORMIF are defined as producing certain effects by means of the hearer's recognition of the speaker's intention to produce these effects. Indirect speech act recognition proceeds by inferring what the user wants the system to think is his/her plan. Plan-recognition involves making inferences of the form, "the user did this action in order to produce that effect, which s/he wanted to enable him/her to do this (next) action".

Making inferences at the level of "intended plan recognition" is begun by analyzing the user's utterance as a "surface" speech act (SURFACE-REQUEST or SURFACE-INFORM) indicating what the utterance "looks like". By performing plan-recognition inferences whose plausibility is ascertained by using mutual beliefs, the system can, for instance, reason that what looked to be an INFORM of the user's goal is actually a REQUEST to include some portion of the ATN into the display. Thus, the second clause of the utterance, "No; I want to be able to see S/AUX," is analyzed as a REQUEST to INCLUDE S/AUX by the following chain of plan-recognition inferences:

The system believes

- (1) the user has performed a SURFACE-INFORM of his/her goal; thus
 - (2) the user intends for the system to believe that the user wants to be able to see S/AUX. Since this requires that S/AUX be visible,
 - (3) the user intends for the system to believe that the user wants the system to plan an action to make S/AUX visible. Because the "No" leads to an expectation that the user might want to modify the display, the system plans to INCLUDE S/AUX in the existing display, rather than DISPLAY S/AUX alone.
 - (4) Hence, the user intends for the system to believe that user wants the system to INCLUDE S/AUX.
 - (5) The user has performed a REQUEST to INCLUDE.
- The system responds by planning that action.

In addition to using Contexts to hold descriptions of beliefs and wants, the plan-recognition process makes extensive use of RoleValueMaps and DSETs (see Figure 4). Plan-recognition inferences proceed using just the clause-level structure and pay no attention to the particulars of the noun phrase interpretations. The system creates new DSETs for intermediate sets and equates them to previous ones by RoleValueMaps, as, for example, when it decides to do a SHOW whose object is to be the same as whatever was to be visible. At the end of plan-recognition the system may need to trace through the constructed RoleValueMaps to find all sets equivalent to a given one. For instance, when it determines that it needs to know which set of things to display, highlight, or include, it treats the equated RoleValueMaps as a set of rewrite rules, traces back to the original noun phrase DSET, and then tries to find the referent of that DSET*.

DSET#0037 are finally specified). Thus, if there are ultimately two SHOWs, one of one state and the other of two, the Y pointer implicitly refers to the set of all three states shown.

* The system only finds referents when necessary. This depends on the user's speech acts and the system's needs in understanding and complying with them. Thus, it is

Finally, not only are parse structures and semantic interpretations represented in KLONE, but the data base -- the ATN being discussed -- is as well (see Figure 2 above). Further, descriptions of how to display the ATN, and general descriptions of coordinate mappings and other display information are represented too. Commands to the display expert are expressed as Concepts involving actions like SHOW, CENTER, etc. whose "arguments" are descriptions of desired shapes, etc. Derivations of particular display forms from generic descriptions, or from mapping changes, are carried out by the attached procedure mechanism. Finally, once the particular shapes are decided upon, drawing is achieved by invoking "how to draw" procedures attached to display form Concepts. Once again, the taxonomic nature of the structured inheritance net allows domain structure to be expressed in a natural and useful way.

Acknowledgements

The prototype natural language system was the result of a tremendous effort by several people: Rusty Bobrow was responsible for the parser and syntactic taxonomy, although his support in design and implementation of KLONE was as extensive and as important; Phil Cohen designed and built the discourse/speech act component that does all of the inference in the system; Jack Klovstad did the graphics, building on an existing system (AIPS) built by Norton Greenfield, Martin Yonke, Eugene Ciccarelli, and Frank Zdybel. Finally, Bill Woods built a pseudo-English input parser that allowed us to easily build complex KLONE structures with a minimum of effort. Many thanks to Phil Cohen, Candy Sidner, and Bonnie Webber for help with this paper.

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-77-C-0378.

References

- [1] Allen, James F. A Plan-based Approach to Speech Act Recognition. Technical Report No. 131/79. Toronto, Ontario: Dept. of Computer Science, University of Toronto, February 1979.
- [2] Bobrow, R. J. The RUS System. In Research in Natural Language Understanding: Quarterly Progress Report No. 3 (1 March 1978 to 31 May 1978). BBN Report No. 3878. Cambridge, MA: Bolt Beranek and Newman Inc., July 1978.
- [3] Brachman, R. J. A Structural Paradigm for Representing Knowledge. Ph.D. Dissertation, Harvard University, Cambridge, MA, May 1977. Also BBN Report No. 3605. Cambridge, MA: Bolt Beranek and Newman Inc., May 1978.
- [4] Burton, R. R. Semantic Grammar: An Engineering Technique for Constructing Natural Language Understanding Systems. BBN Report No. 3453. Cambridge, MA: Bolt Beranek and Newman Inc., December, 1976.
- [5] Woods, W. A., Kaplan, R. M., and Nash-Webber, B. The Lunar Sciences Natural Language Information System: Final Report. BBN Report No. 2378. Cambridge, MA: Bolt Beranek and Newman Inc., 1972.

intended that a naming speech act like "Call that the complement network" will not cause a search for the referent of "the complement network".

