Towards Completely Integrated Parsing and Inferencing*

Christopher K. Riesbeck and Charles E. Martin
Yale University

ABSTRACT

Our goal is the complete integration of natural language understanding with the rest of cognition. Two mechanisms that we have developed and implemented to achieve this goal are: (1) the Direct Memory Access Parsing (DMAP) algorithm, based on the notion of lexically-guided memory search and concept refinement, and (2) an inference-triggering process based on the notion of concept refinement failures. Together, these two mechanisms form a tightly-integrated system of parsing and inferencing, with no artificial boundaries between them.

Introduction

Our goal is the complete integration of natural language understanding with the rest of the cognitive system. The benefits of full integration are obvious: such a parser could take full advantage of whatever knowledge was present in memory (or, at least, could take as much advantage as any other memory process could), and other memory processes could make full and immediate use of linguistic input, without waiting for a final interpretation to be formed.

This paper describes two mechanisms that we have developed and implemented to achieve this goal. First, our parsing algorithm is a process of lexically-guided memory search. That is, patterns of words and concepts guide a general memory search process towards relevant memory structures, like lighthouses guiding a ship into harbor. We call this *direct memory access parsing* (DMAP). Second, our inference processes are triggered by *specialization failure structures*, generated by the parser, to record problems in building new instances of memory structures. Together, these two mechanisms form a tightly-integrated system of parsing and inferencing, with no artificial boundaries between them.

Our memory structures are frame-like objects called *Memory Organization Packets* (MOPs), organized into the standard part-whole (*packaging*) and class-subclass (*abstraction*) hierarchies [Schank 1982]. We integrate parsing knowledge into memory by attaching linguistic templates to these memory structures, in a manner reminscent of the Teachable Language Comprehender [Quillian 1969]. These templates, called *concept sequences*, are patterns of words and concepts. Attached to Milton Friedman, for example, is the lexical phrase "Milton Friedman". Attached to the general concept of a communication event is the concept sequence "[actor of communication] says [object of communication]." Any memory structure can have one or more concept sequences, plus every structure implicitly inherits the sequences attached to abstractions of that structure.

The dictionary in DMAP, which we call the *concept lexicon*, is simply a set of pointers from words and concepts to the concept sequences they appear in. For example, "Milton" has a pointer to the sequence "Milton Friedman" which is attached to the person concept MILTON-FRIEDMAN, and the concept HUMAN has a pointer to the sequence "[actor of communication] says [object of communication]." Obviously, "Milton" might point to every person the system knows named Milton, and HUMAN might point to every action described with a sequence involving a human. Our current concept lexicon only has a pointer from a word or concept to a concept sequence if the word or concept appears at the beginning of the sequence. Even so, the DMAP model depends on the use of

---

*This work was funded in part by the Air Force Office of Scientific Research under contract F49620-82-K-0010.

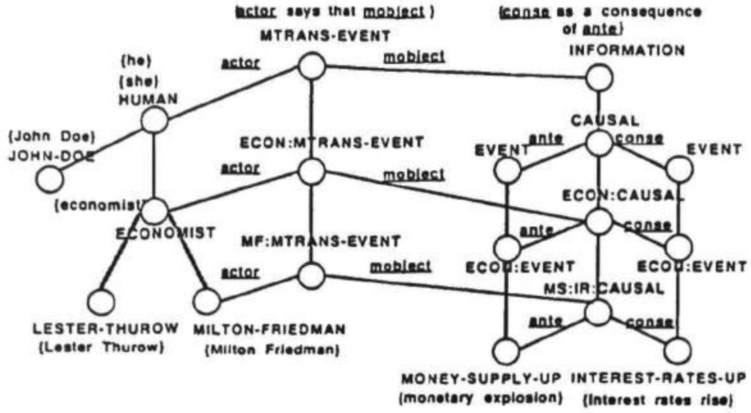

Figure 1: A portion of the DMAP memory.

parallel activation and intersection to resolve the basic combinatorial explosion, as is presumed in a number of other recent models [Small et al. 1982] [Hahn and Reimer 1983] [Granger et al. 1984] [Waltz and Pollack 1984][Charniak unpb].

When DMAP reads a sentence, such as "Milton Friedman says that high interest rates are the consequence of the monetary explosion," the concept lexicon leads it to concept sequences attached to memory structures, such as "economist claims economic causal connection," (ECON:MTRANS-EVENT [1] in Figure 1), and "rise in money supply causes rise in interest rates" (MS:IR:CAUSAL). Filling out these sequences activates the associated memory structures.

Because the parsing process, as described in the next section, pushes activation down to the most specific memory structures available, exactly which memory structures the parser settles on depends on which ones are already in memory. If this claim of Friedman's has been seen before, then seeing it again, as originally stated, or paraphrased, will guide the parser to the previously built memory structure MF:MTRANS-EVENT.

If MF:MTRANS-EVENT is not already in memory, then the parser is stops at a more general level, such as ECON:MTRANS-EVENT. When the parser cannot finding a more specific structure, either because there are none, or because the ones that exist do not match the input, it activates a specialization failure structure. One such structure is "actor exception", which means that the input event partially matches some existing memory structure, but the actors are different. For example, if the system has already seen the Milton Friedman sentence, and now reads "John Doe blames the large increase in money supply for the rise in interest rates," it will find the Milton Friedman structure, but be unable to specialize to it because of the mismatch in actors.

Specialization failure structures, like other memory structures, are organized by part-whole and class-subclass relationships. The responses attached to these failure structures are *reconciliation* processes that propose resolutions to the failure. Resolutions are proposed by activating potential explanation patterns (XPs) [Schank 1986].[2] A routine domain-specific XP for explaining why two economists say the same thing is "they belong to the same economic camp." The XP may or may not be applicable, depending on what the system already knows, i.e., what other memory structures it has that package and abstract Thurow and Friedman.

In the rest of this paper, we first discuss the details of direct memory access parsing. We will then describe two classes of specialization failure structures, and how they're activated during the parsing process.

Direct Memory Access Parsing

This section describes the current implementation of the DMAP interpreter. The interpreter uses a marker-passing architecture to identify relevant memory structures from the input text and the expectations in memory. Two kinds of markers are used in the system: *activation* markers, which cap-

---

[1] MTRANS is our primitive marker for communication events [Schank 1975].

[2] Specialization failure structures are similar in spirit to the Exception MOPs proposed in [Riesbeck 1981].

ture information about the input text and the current selection of relevant memory structures, and *prediction* markers, which indicate which memory structures may be expected to become relevant.[3]

The structure of memory

Figure 1 is a simplified portion of the DMAP interpreter's memory to represent the communicative act of the following text.

> Milton Friedman: Interest rates will rise as an inevitable consequence of the monetary explosion we've experienced over the past year.[4]

The central structure to this portion of memory is the MOP MF:MTRANS-EVENT, which packages MILTON-FRIEDMAN and MS:IR:CAUSAL via the actor and mobject roles. The concept sequence {actor says that mobject} is associated with the more general MTRANS-EVENT MOP. This concept sequence is the linguistic template used to recognize the memory structure MTRANS-EVENT. It is a pattern of words (e.g., "says") and concepts in the form of packaging relationships from which concepts are referenced (e.g., actor references HUMAN from MTRANS-EVENT, but MILTON-FRIEDMAN from MF:MTRANS-EVENT).[5]

Concept activation

Memory structures are activated by placing activation markers on them. Activation markers are created in two situations.

- *System input:* when an input word is read by the interpreter, an activation marker is created and placed on the associated lexical item in memory.
- *Concept sequence recognition:* when every element of a concept sequence has been activated, an activation marker is created and placed on the associated memory structure.

Activation markers are passed up the class-subclass abstraction hierarchy from their associated structures. This is a recursive process; all structures which receive an activation marker continue to pass it on to their own abstractions. When a memory structure receives an activation marker, that memory structure is said to have been *activated*; the activation marker contains a pointer to the originally activated memory structure.

For example, an activation marker associated with the memory structure MONEY-SUPPLY-UP will be passed to ECON-EVENT, which in turn passes the marker to EVENT. All of these structures are activated, while the activation marker keeps a pointer to MONEY-SUPPLY-UP.

Concept prediction

The *concept lexicon* indexes concept sequences under the memory structures referenced by the first elements of the sequences. For example, the MILTON-FRIEDMAN structure's concept sequence {Milton Friedman} is indexed under the lexical item "Milton", and the MTRANS-EVENT MOP's concept sequence {actor says that mobject} is indexed under the structure HUMAN, since HUMAN is referenced by the actor role of MTRANS-EVENT. These indicies are not shown on the diagram.

Prediction markers represent concept sequences which are in the process of being recognized. Whenever a memory structure is activated, prediction markers are created for all the concept sequences indexed by that memory structure through the concept lexicon. A prediction marker is a structured object which records its associated *concept sequence*; the *current element* of the concept sequence (either a packaging role or a lexical item); the *reference structure* which will be activated if the concept sequence is recognized—initially this is the structure associated with the concept sequence; and the *target* of the prediction (either a memory structure or a lexical item).

A prediction marker is always located at the target memory structure, which is derived from the reference structure and the current element. A prediction marker is passed through memory whenever its reference structure or current element is changed; this takes place through two concurrent processes.

---

[3] The markers of the system are structured objects which would be unacceptable in current connectionist theories [Feldman and Ballard 1982] [Waltz and Pollack 1984].

[4] *The New York Times*, August 4, 1983.

[5] The EVENT and ECON:EVENT MOPs are shown twice in the diagram to make the packaging relationships clear. There is only one structure for each in memory.

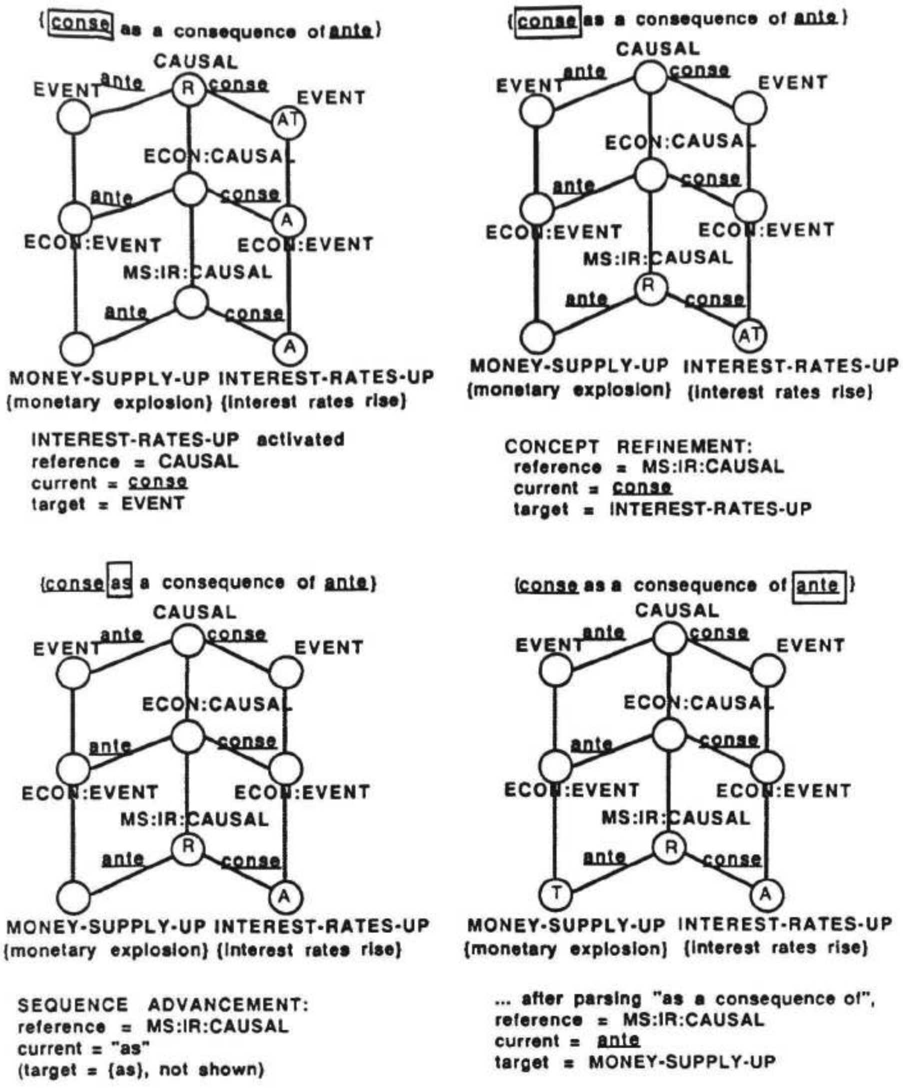

Figure 2: How prediction markers are passed.

- *Concept refinement:* This occurs when the target of a prediction marker whose current element is a packaging role is activated. Since the reference structure will generally package an abstraction of the originally activated structure, the reference structure can be replaced by a specialization which explicitly packages the original activation.

- *Prediction satisfaction:* when the target of a prediction marker is activated, the current element of the concept sequence is satisfied by that activation. The current element is replaced by the next element in the concept sequence. (If there are no further elements, then the concept sequence has been recognized, and the reference structure is activated.)

For example, the concept sequence {conse as a consequence of ante} is associated with the structure CAUSAL; if INTEREST-RATES-UP is activated, the subsequent activation of EVENT will index this concept sequence through the concept lexicon, and a new prediction marker will be produced. Concept refinement will change the reference structure from CAUSAL to MS:IR:CAUSAL (the most specific memory structure which packages the activated memory structures), and prediction satisfaction will change the current element to the "as" lexical item. The new target is this lexical item, and the prediction marker is passed to this memory structure. After reading the phrase "as a consequence of", the current element will become the mobject role, which references MONEY-SUPPLY-UP from the reference structure MS:IR:CAUSAL. MONEY-SUPPLY-UP becomes the new target. This "trace" of the parse is depicted in Figure 2.

Specialization Failure Structures

We've now seen how the parsing process works when the net result is simple recognition. A key event that the DMAP interpreter watches for is when an activation marker meets a prediction marker. This causes a specialization process to occur, which spreads the prediction markers down the abstraction hierarchy. If the prediction markers can be pushed all the way down to the level of the input activation markers, then the parsing process has reconciled the input completely with some existing memory structure. Thus, given the memory in Figure 1, "Milton Friedman says that high interest rates are a consequence of the monetary explosion" activates MF:MTRANS-EVENT and that node packages nodes with input activation markers.

If MF:MTRANS-EVENT did not already exist in memory, then the parser would activate the specialization failure structure MISSING-SPECIALIZATION, which represents the situation where the parser can't specialize a memory structure because no specialization of that structure exists. In this case, the parser can't specialize ECON:MTRANS-EVENT, even though Milton Friedman and his argument are more specialized than ECONOMIST and ECON:CAUSAL.

The MISSING-SPECIALIZATION structure packages the general structure and the specialized fillers, e.g., ECON:MTRANS-EVENT and the specialized fillers MILTON-FRIEDMAN and MS:IR:CAUSAL. The reconciliation strategy attached to MISSING-SPECIALIZATION is simple: instantiate a new specialization of the existing memory structure to package the specific items. We call this reconciliation ROTE-MEMORY because it simply adds the items seen to memory. If MF:MTRANS-EVENT was not already in memory, then ROTE-MEMORY would create it.

ROTE-MEMORY is invoked only in the simple situation where you know things of a certain type can occur, and one of them does. The input matches completely a general pattern and there is no more specific version of the pattern to compare the input with.

Of more interest is the case where the input *partially* matches some existing structure. For example, suppose that MF:MTRANS-EVENT is in memory and the parser reads "John Doe blames the large increase in the money supply for the rise in interest rates." When this is parsed, the parser is unable to specialize from ECON:MTRANS-EVENT to MF:MTRANS-EVENT because of the mismatch between the actor of the input, i.e., John Doe, and the actor of MF:MTRANS-EVENT, i.e., Milton Friedman.

This specialization failure is called ECONOMIC-MTRANS:ACTOR-EXCEPTION. It packages together: a *reference structure* that couldn't be specialized, a *potential reference structure* (a specialization of the reference structure) that the parser could not reach because the *activated structure* which satisfied the actor role of the prediction marker is not an abstraction of the *anomalous feature structure* packaged by the actor role of the potential reference structure. In the John Doe example these would be ECON:MTRANS-EVENT, MF:MTRANS-EVENT, JOHN-DOE, and MILTON-FRIEDMAN, respectively.

The general situation of two people saying the same thing can be explained in many ways, but we are interested here in explanations specific to economists and their public pronouncements about the economy. That is why we focus on ECONOMIC-MTRANS:ACTOR-EXCEPTION, rather than the more general MTRANS:ACTOR-EXCEPTION.[6] One reconciliation strategy for ECONOMIC-MTRANS:ACTOR-EXCEPTION is CREATE-ECONOMIST-CAMP. This strategy takes the following actions.

- Create a new *economist camp* memory structure which is a specialization of the least abstract generalization of the activated and anomalous feature structures, and make these features specializations of it.
- Create a new *camp mtrans event MOP* which is a specialization of the reference structure; the new mop packages the economist camp

---

[6] Since these failure structures are organized in the regular hierarchical memory format, strategies at the level of MTRANS:ACTOR-EXCEPTION are accessible if more specific strategies are ineffective in resolving the anomaly.

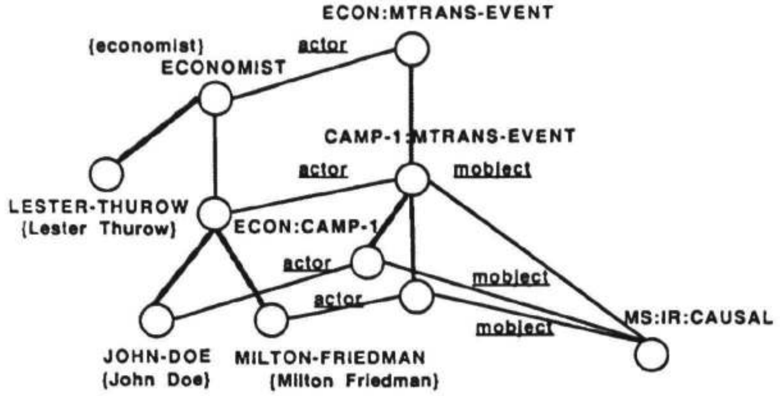

Figure 3: Building new structures.

structure and the other activations of the input.

- Add the potential reference structure and the structure created by the ROTE-MEMORY strategy as specializations of the new camp mtrans event MOP.

In this example, the strategy creates as a specialization of ECONOMIST an economist camp MOP (call it ECON:CAMP-1) and makes JOHN-DOE and MILTON-FRIEDMAN specializations of it. It packages this new MOP by a new camp mtrans MOP, CAMP-1:MTRANS-EVENT, which is a specialization of ECON:MTRANS-EVENT, and packages MF:MTRANS-EVENT and the new John Doe mtrans event created by ROTE-MEMORY as specializations of it. CAMP-1:MTRANS-EVENT has the interpretation that the new economic camp claims that that increased money supply is causing high interest rates. Figure 3 shows the net result.

A topic for future research is how the system might learn a name for ECON:CAMP-1, such as "monetarist," while reading text.

CREATE-ECONOMIST-CAMP would apply if we knew nothing else about Friedman and Doe except these two statements. Suppose however that the concept of "monetarist" is already in memory, Friedman and Doe are instances of it, Lester Thurow is known to be in another camp, and the parser reads "Lester Thurow blames the rise in interest rates on the increased money supply." The appropriate failure structure in this case is ECONOMIC-MTRANS:CAMP-EXCEPTION, which is a specialization of ECONOMIC-MTRANS:ACTOR-EXCEPTION that represents a failure to specialize because of a camp mismatch, rather than an actor mismatch.

ECONOMIC-MTRANS:CAMP-EXCEPTION, is an anomaly requiring more creativity in explanation than our simple routine XP strategies are intended to account for. Possible explanations are that Thurow is changing camps, or he's lying, or the monetarist position is becoming more acceptable to non-monetarists. Coming up with explanations at this level of complexity is under investigation [Schank 1986], but not in the DMAP framework.

Causal Chains

Although we focussed above on specialization failure structures and related strategies for adding economic communication events to memory, the same approach is being used to direct inferencing to build causal chains as well.

Building causal chains is a key part of economic reasoning. For example, one text that DMAP parses and reasons about is

> With high growth choked off by high interest rates, the deficit will be bigger, not smaller. [7]

To understand this text, a causal chain has to be found connecting interest rates to growth, and growth to the deficit.

When the parser reads "high growth choked off by high interest rates," it tries to specialize the economic causal "high interest rates cause reduced growth." If there is already is such a causal in memory, there is no problem, but if there isn't, then the specialization failures for economic causal structures are activated.

---

[7] Lester Thurow, *Newsweek*, September 21, 1983.

If there is no causal argument in memory mentioning interest rates or growth at all, the strategy ROTE-MEMORY will just add the input causal to memory. If there is instead a causal argument that says interest rates cause inflation, then the strategy attached to the specialization failure structure ECONOMIC-CAUSAL:CONSEQUENT-EXCEPTION will try to reconcile the input with memory by finding a second causal connecting increased inflation to reduced growth, hence building a causal chain.

Conclusions

We've discussed two techniques for integrating parsing and inferencing with a hierarchical memory. First, the direct memory access parsing algorithm attaches concept sequences to structures in memory and uses these sequences to guide memory search to the most specialized applicable packaging structures available. Second, specialization failures during parsing activate specialization failure structures, to which are attached reconciliation strategies that create new memory structures that will allow the input to be instantiated and installed into memory. The examples of reconciliation strategies given deal with one economist agreeing with another, and with causal chains. Our belief is that the reconciliation strategies are a significant part of the total inference processing that goes on during text understanding and evaluation.

Acknowledgments

The work described here is based on the joint efforts of the Direct Memory Access Parsing project, which consists of Charles E. Martin, Monique Barbançon, and Michael Factor.

References

Charniak, E. (Unpublished). *A Single-Semantic-Process Theory of Parsing.*

Feldman, J.A. and Ballard, D. (1982). Connectionist models and their properties. *Cognitive Science, 6* (3), 205–254.

Granger, R.H., Eiselt, K.P., and Holbrook, J.K. (1984). The parallel organization of lexical, syntactic, and pragmatic inference processes. In *Proceedings of the First Annual Workshop on Theoretical Issues in Conceptual Information Processing.* Atlanta, GA.

Hahn, U. and Reimer U. (1983, November). *Word expert parsing: An approach to text parsing with a distributed lexical grammar.* Bericht TOPIC 6/83. Universitat Konstanz, Konstanz, West Germany.

Quillian, M.R. (1969). The Teachable Language Comprehender: A Simulation Program and Theory of Language. *Communications of the ACM, 12* (8).

Riesbeck, C.K. (1981). Failure-driven Reminding for Incremental Learning. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence.* Vancouver, B.C..

Schank, R.C. (1975) *Conceptual Information Processing.* Amsterdam: North Holland/American Elsevier.

Schank, R.C. (1982) *Dynamic Memory: A Theory of Learning in Computers and People.* Cambridge University Press.

Schank, R.C. (1986) *Explanation Patterns: Understanding Mechanically and Creatively.* In preparation.

Small, S., Cottrell, G. and Shastri, L. (1982). Toward Connectionist Parsing. In *Proceedings of the AAAI-82.* Pittsburgh, PA..

Waltz, D.L. and Pollack, J.B. (1984). Phenomenologically plausible parsing. In *Proceedings of the AAAI-84.* Austin, Texas.