ORGANIZATION AND RETRIEVAL IN A CONCEPTUAL MEMORY FOR EVENTS
OR
CON51, WHERE ARE YOU?

Janet L. Kolodner
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

In the course of a normal day, people effortlessly recall past events and episodes from their lives. A reasonable goal in the design of computer programs is to construct a memory with that same capability. To facilitate such retrieval of events from a computer memory, we must first specify a reasonable memory organization. We must then design updating and retrieval processes to build up and access that information. This paper will present such a theory, and will describe a computer program named CYRUS which implements the theory.

1.' Introduction

How can a large memory be searched efficiently? This is one of the biggest problems facing the designers of large systems, yet one which has been sorely neglected. While there has been a lot of work done on problems of knowledge representation and inference processes, the assumption of most research projects has been that the necessary Information would be available when needed without search. Without a well-structured memory organization, however, this will not always be true. This paper addresses the problem of organizing and retrieving events from a computer memory. There are three major questions to answer:

1. What are the prooesses for retrieving events from memory?
2. What memory organization do the retrieval processes imply?
3. What are the processes for adding new events to memory, and how does memory organization change as new events are added?

These questions cannot be independently answered. The organization of memory constrains the types of retrieval and updating processes the memory oan have. On the other hand, memory organization, and therefore procedures for adding information to memory, must be based on retrieval requirements.

In considering these problems, I will present CYRUS, a computer system that retrieves events from memory and automatloally reorganizes Itself as new events are added to it. CYRUS (Computerised Yale Retrieval and Updating System) stores and retrieves episodes in the lives of former U. S. Secretaries of State Cyrus Vance and Edmund Muskie. CYRUS can answer questions posed to it in English about the events it stores.

CYRUS can be seen as both a model of human memory and an intelligent information retrieval system. As an information retrieval system, however, it bears little resemblance to current systems. Information retrieval systems have based their memory organizations on keywords and not on conceptual categories. Because they do not organize their contents according to similarities of meaning, they cannot apply meaning-based heuristics for retrieval or for category reorganization.

Within psychology, long term memory has been described as a reconstructive process (e.g., [9]. [5]). The processes that have been described by psychologists, however, have been described in very general terms, independent of a memory organization or a description of the knowledge guiding the processes. This research, which explains a memory organization and particular retrieval processes, can be thought of as complementary to the psychological research.

2, The nature of a long term memory for events

There are a number of features of human memory which are desirable in a computer system designed to retrieve events. People learn new things every day, but they do not get slower at remembering [8]. In a computer system, too, <u>retrieval should not slow down significantly as new events are added to memory</u>. This requirement constrains both the retrieval processes and the memory organization. It has been central to development of the memory processes that will be presented.

Retrieval from any category must be able to happen without enumeration of the category. Within computer science, the traditional solution to this problem is to index items within categories. In adopting this solution, we must specify how to choose indices for an event in a particular category. If events are indexed in a category by all of their features which are salient to the category, then specification of an Indexed feature will enable direct retrieval of items with that feature.

The richer the indexing, however, the more spaoe is needed for storage. <u>Indexing must be controlled so that memory does not grow exponentially</u>. Similarities between events oan be used to control indexing. If memory keeps track of the similarities between events within a category, then indexing can be limited to the differences between events. It is the differences which will discriminate events from each other. A long term memory should be able to maintain itself. It must be able to ooapute indloes for new events, creating new conceptual categories when necessary and building up required generalized infomation.

Slnoe people seem to be good at remembering, we have turned to people to try to find efficient algorithms for retrieval and organisation. In previous research ([9] (71), It has been proposed thst people remember by reconstructlng what must have happened. The retrieval process can be seen as a propose of specifying and elaborating contexts for" search. In terms of the computer Implementstion, to retrieve any particular target event, it Is necessary to (1) specify a memory category that the event might be found in, and (2) compute differentiating features of the event within thst cstegory. In order to explain this process In more detail. It Is first neoesssry to give sn overview of the memory organisation.

3* An overview of the organisation

The memory organisation we are assuming Is bssed on conceptual categories for events. These categories index similar episodes scoordlng to their differences and keep track of their similarities. These categories will be referred to ss Episodic Memory Organisation Psokets (E-MOPs), or generlcslly ss "MOPs". These structures ere related to Schank's [6] HOP* and to Lebowits's [4] S-MOPs, but the concerns in defining MOPs snd S-MOPs were different than those in defining E-MOPs.

Sn E-MOP is s net In which esch node is either an E-MOP or an event. Each E-MOP has two important aspects — (1) generalited information characterising its episodes, snd (2) tree-like structures thst index those episodes by their differences. An E-MOP's norms Include information describing its events, suoh as their usual participants, locations, and topics, and their ususl reletionships to other events.

An E-MOP's indices can index either individual episodes or speolsllsed E-MOPs. When an E-MOP holds only one episode with s particular index, that index will point to the individual episode. When two or more episodes in an E-MOP shsre the same feature, its corresponding index will point to s specislised sub-MOP (with the structure just described) which organises the events with thst feature. In this way, MOP/sub-MOP hierarchies are formed. The MOP below is part of CYRUS' "diplomatic meetings* E-MOP. "Diplomatic meetings'* holds generslised information about "diplomatic meetings", while MOP2 snd MOP3 index "meetings with Begin" snd "meetings about the Camp David Aooords" respectively.

Indexing is two-tiered, where the first tier indexes types of features, and the second indexes vslues for the festures themselves. Thus, by following the index for "participants", snd from there following the index for Begin, the sub-MOP organising "meetings with Begin" osn be found* Following indices for "topic" and from there the index for "SALT", one arrives st the individual event EV2, the only meeting about SALT indexed in this MOP.

This orgsnlsatlon provides rioh oross-indexing of events in memory. Specification of any discriminating set of event features within *an* E-MOP allows retrieval of the corresponding event. Using s rlohly indexed organisation suoh as this, enumeration *of* s memory category should never be neoesssry for retrievsl. Instead, retrieval strategies osn be used to expend on question components, thereby inferring relevent psths through the memory structures. In this wsy, sesroh is directed only to ostegorles snd sub-ostegorles whose events are relevant.

```
                   'diplomatic meetings'

norms:  the actor is Cyrus Vance
 (MOP1) participants are foreign diplomats
        topics are international contracts
        participants talked to each other
        goal was to resolve disputed contract
diffs:           /                         \
        participants                  topic
      ../  |    \                ..../  |  \..
    ./  (1)|     \              /       |     \  (2)
   /    Dayan  Gromyko      SALT        |    Jerusalem
   |      |      |           |          |        |
   |     EV4    EV2         EV2         |       EV3
(3)|                               (4)  |  (MOP3)
Begin  (MOP2)                   Camp David Accords
 norms:                           norms:
   partic include Begin             topic is the CDA
   topic concerns Israel            partic are Israeli
      and Arabs                     specializatn of MOP1
   specialization of MOP1         diffs:     |
 diffs:        |                             | partic
               | topic              (7)     / \ (8)
 (5)         / \  (6)                 Begin   Dayan
 Jerusalem    Camp David Accords  (MOP4)     |
     |             (MOP4)                    EV4
    EV3
```

*. setrlevlmg an event from an E-MOP

Given appropriate features of sn event, its retrievsl from an E-MOP israther trivial. Conalder, for example the following question:

> (Q1): Have you ever attended a diplomstic meeting about the Camp David Aooords with Daysn?

The answer to (00) cn be found either by traversing the indioes for "has Dayen as s participant" or by first following the indioes for "has topic the Camp David Aooords", snd then from there traversing those for "has Dayan as s participant". Either way, EVS would be found.

This retrieval process oan be characterized as • trsverssl prooess, s process of following appropriate indioes down s tree until sn event is found. An event to be retrieved from an E-MOP is called a target event.

Rich cross-indexins of events in E-MOPs tables directed seerch of memory. To ensure dlreoted sesroh, the first step of trsverssl must be specifiestlon of paths to traverae, or seleotion of indioes for traversal.

Index seleotion is bssed on festures speolfied in the target event. Indices ohoeen for trsverssl to find any target event should be features that would have been chosen ss indioes for thst event if it had previously been Indexed in the E-MOP. Thus, the same Index selection process is used both when adding an event to memory and during retrieval. Index selection will be discussed in section 5.1.

Traversal is a <u>recursive process</u> involving choloe of Indloes and traversal of those Indices. It stops when an event is found, or when there are no additional specified Indices to be traversed. Thus, if there *Br%* multiple paths to a target event, it will be retrieved from the shortest path with all of its Indloes specified in the target event. We can think of traversal as a breadth-first search which implements parallel traversal of all appropriate indices.

4.1 The need for elaboration

Retrieval of a target concept which specifies an event feature or combination of features which are both <u>Indexed</u> and <u>unique</u> can be done easily through traversal. When a target concept specifies an unindexed feature or does not specify a unique combination of features, the traversal algorithm presented above will fail, as in the example below:

(Q2): Have you ever attended a meeting in Jerusalem?

Answering this question requires that a meeting in Jerusalem be found. Using the algorithm and MOP above to answer (Q2) , there are no features specified by the question that can be used directly to find an event. The traversal process described must abort, even though there are events in the E-MOP which might have occurred in Jerusalem. Although we have outlawed enumeration in the normal case, there must be a way of retrieving events from an E-MOP even when indexed features are not specified in the question.

The traversal process can be continued if plausible Indloes can be computed. Since M0P1 Indexes events by "topic" and "participants", if either of those could be inferred for a meeting in Jerusalem, then actual meetings could be retrieved from the MOP. We call the process of specifying additional features of a target event <u>elaboration</u>.

The prooesses CYRUS uses for elaborating on a retrieval specification are called <u>instantiation strategies</u>. They use information specified in a target concept and information associated with the E-MOPs the target conoept fits into to better specify target ooncept features. The following is one of CYRUS* instantiation rules:

Infer-Participants

Infer participating people by retrieving representatives of specified organizations, members of known groups, representatives of known countries, or persons associated with known organisations, groups, or countries.

Sinoe a "diplomatic meeting" is a political event whose participants are political dignitaries, this rule can be used to infer that possible participants in a "meeting in Jerusalem" would have been important Israeli dignitaries. A similar strategy associated with "topios" would allow the "Camp David Acoords" to be inferred as a possible topic for such a meeting.

CYRUS uses these strategies to elaborate "a meeting in Jerusalem" in the following way:

"a meeting in Jerusalem with Begin, Day an, or Weismarm, with probable topic the Camp David Acoords"

It then attempts traversal of the appropriate Indloes. EV4 would be found in this way and would then be checked to make sure it had happened in Jerusalem. MOPM would also be reached during traversal, and additional elaboration would be done to traverse its indices. We oan envision the entire process as follows:

Traversal/Elaboration

```
                    +----------------+
                    | select indices | <--------------+
                    +----------------+                |
                       /          \  done             |
              "more info       +---\--------------+   |
               needed"      -->| traverse indices |   |
     +-------/-----+   /       +------------------+   |
     | elaboration |  /         found /       \ found |
     +-------------+ /          event/        \ E-MOP |
         /     \    /        +-----/----------+ +-------+
   failed      +---+         | check features |
      /        success       +----------------+
     /                        no match /    \ match
"not found"                           /      \
                           "not found"     return event
```

4.2 Context construction

In order to search memory, a <u>category for search</u> must first be chosen — in CYRUS, an E-MOP. Strategies in CYRUS which choose initial categories for search are called <u>component-to-oontext instantiation strategies</u>. These strategies use event information associated with question components to infer plausible E-MOPs. Consider, for example, the following question:

(Q3): Who have you talked to about SALT?

In answering this question, CYRUS uses information about SALT to infer the context of a "political meeting*[1], and searches memory for "political meetings with topic SALT". It is able to make this Inferenoe because "SALT" is an "international oontraet" and "international contracts" are known to be the topios of "political meetings". In this case, the strategy "Infer-from-Topio" does that work:

Infer-from-Topic

If the event is an MTRANS, and its topic (MOBJECT) is specified, then predlot oommunlcatory contexts associated with the topic.

4.3 Alternate-context aearoh

In using elaboration to answer (Q2), we ststed that'EV* would be found, and that MOP4 would be found, but additional elaboration would be needed to traverse its indicts. Suppose that there was not enough information available to do the additional elaboration neoessary for traversal of HOP*, or alternatively, that the only meeting whloh had taken place in Jerusalem waa the one indexed in "diplomatic meetings" as S "meeting about Jerusalem". In either of those osses, elsborstion would not suffloe to snswer the question.

Events, however, occur in the contexts of other events snd refer to those related events, an event can be found by finding an epiaode it was relsted to. When s relsted event is found, its context csn be sesrched for the target. Thus, to search for s meeting in Jerusalem, it might be appropriate to recall a trip to Israel or negotiations involving Isrsel whloh might hsve included such a meeting in its sequenoe of events. Since trips snd negotistlons are less common than diplomatic meetings, they might be easier to retrieve [7].

Context-to-context instantiation strategies are used to construct slternste oontexts f5F sesrch. In order for possible related oontexts to be Inferred from s target event, E-MOPs must specify both the types of episodes (other E-MOPs) they are often related to, and how those episodes sre relsted, i.e., how their roles correspond to those of the relsted E-MOP. A diplomatic trip related to a meeting in Jerusalem, for example, can be inferred to have been to Isrsel.

Sesrch strstegles direct search for alternate contexts. They hsve the following steps:

1. choose a context to be searched for, snd csll appropriate instantiation strstegles to construct and elaborate that context
2. retrieve corresponding events from memory
3. sesrch for the target event in the surrounding contexts of the events retrieved

In sesrchlng for "diplomatic meetings In Jerusalem", a searoh strstegy whloh searches for episodes sn event could have been mart of would (1) construct contexts for "diplomatic trlpa to Israel," and "negotiations concerning an Israeli Issue," (2) traverse memory searching for eeoh of those, and (3) searoh the sequenoe of events of each epiaode found for an appropriate diplomatic meeting. The following la an example from CTRUS Which illustrates the retrieval process:

```
>Who hsve you discussed SALT with?
    inferring political meeting — sM-MEETING
 sesrchlng memory for question concept
    additional information needed
      meeting could have oocurred In USSR or USA
        searching for meeting in USSR
        sesrchlng for meeting in USA
 searching memory for episodes meeting could
         have occurred in
          searching for I-NEGOTIATE
            searching I NEGOTIATE for meeting
          sesrchlng for conferences about SALT
          aearohlng for diplomatic trlpa to USSR
       sesrohlng memory for standard types of meetings
         sesrohlng for diplomatic meetings about SALT
         searching for briefings about SALT
         searching for public relations meetings

CARTER, BREZHNEV, GRONYKO, OTHER AMERICAN AND
RUSSIAN DIPLOMATS, AND MUSTAFA KHALIL.
```

4.4 Summary

The following chart illustrates the entire retrieval process:

The Retrieval Process

```
     +------------------------------+
     | choose a context for search  | <--------+
     +------------------------------+          |
                  \ /                          |
       +-----------------------+               |
       | traversal/elaboration |               |
       +-----------------------+               |
          /             \                      |
      succeed          fail                    |
        /       +---------\--------------+     |
  return        | alternate context search |   |
  events        +--------------------------+   |
                  succeed /        \ fail      |
                        /           \----------+
                  return events
```

This retrlevsl process trades speed of retrieval for memory spaoe. It is fsster than enumeration would be, but needs more memory. It is also less accurate than enumeration would be. If cues necessary to choose an appropriate oontext are not available at the time of retrieval, then retrieval will fall. Similarly, if the knowledge necessary to do appropriate elaboration is not available, then retrieval will fall. What this soheme does silow Is (1) retrieval in most oases that is faster than enumeration and (2) retrieval of generalised information in exactly the same way actual events are found.

5. Maintaining memory orgsnixstion

In order for retrlevsl to work consistently as new events ^9 added to memory, memory'a organisation must be maintained. When a new event la added to an E-MOP, it must be Indexed so that it can later be retrieved. New E-MOPs must be created aa neoessery, and generalised information neoesssry to aid retrieval must be built up.

The firat step in adding a new event to an E-MOP la to ohoose appropriate feat urea of the event for indexing. Each feature ohosen oan have one of four relatlonahlps to the E-MOP [2]:

1. There is nothing yet Indexed in the E-MOP with that feature

2. There is one other item with that feature indexed in the E-MOP
3. There is an E-MOP Indexed by that feature
4. it is on of the E-MOP's norms.

When there is not already an index for a feature (1), a new index is built, and the event is indexed at that point. When there is one other event with a particular feature (2), a new E-MOP is formed based on the similarities between the new event and the previous one with that feature, and the two events are indexed in that E-MOP. When there is already an E-MOP Indexed by a particular feature (3), the new event is integrated into that E-MOP. That integration includes refining the E-MOP's generalized information and indexing the event. If the feature is one of the E-MOP's norms (4), no indexing is done. The remainder of this section will describe index selection, and building up generalised information during processes associated with (2) and (3) above.

5.1 Index selection

Index selection is part of both retrieval and memory update. It takes an event and an E-MOP as input, and produces a subset of the event's features to be used for indexing the event in the MOP (during update) or for traversal (during retrieval).

Index selection in a particular E-MOP depends on the MOP's norms. In general, to maintain discrlmlnlblllty between events, normal aspects of a situation should not be indexed, while weird and different aspects of a situation should. Indexing by a norm would supply memory with unneeded redundancy, and violate economy of storage. Differences between events, on the other hand, differentiate them from each other, providing <u>discrimlnablllty</u>. Organising events according to differences allows events to be recognixed individually. If a unique difference from a norm is specified in a retrieval key, the event that corresponds to that specification can be retrieved.

Another important property indices should have is <u>predictive power</u>. A feature which is predictive often co-occurs with some other event feature. The nationality of participants in a diplomatic meeting, for example, Is usually the same as one party to the oontract being discussed. Thus, in a "diplomatic meetings" MOP, the nationality of participants can help predict the meeting's topic, and la a good predictive feature for indexing.

Predictions that a particular feature or set of features can make are used during retrieval for elaboration. During retrieval, specification of the value of a predictive property will allow inferenoe of the properties it predicts. If the feature "participants are Russian" oo-ooours with "topic is usually arms limitations", then knowing a meeting was with a Russian will allow Inferenoe of the meeting's topic

Of course, we oan*t tell for sure, the first time we see a particular feature, whether or not it will later be predictive. Predlctlveness of features, however, can be judged by previous experience. If a <u>type</u> of <u>index</u> (e.g., nationality of participants, sides of a contract) has been useful previously for similar events, then there is a good ohance it will be useful for the current event. This implies that as new events are added to memory, the relative predictive power of different types of indices must be tracked.

The predictive power of a feature <u>depends on</u> the <u>context</u> in <u>which</u> H <u>is found</u>. Thus, E-MOP indices must make <u>context-related predictions</u>, i.e., predictions about MOP-specific features. These criteria suggest the following algorithm for index selection:

Index Selection

```
+-----------------------------------------------+
| Select the types of features from the event   |
| that have been predictive previously.         |
+-----------------------------------------------+
                      \ /
  +---------------------------------------------+
  | Get rid of                                  |
  |   (1) features known to be non-predictive   |
  |   (2) E-MOP norms                           |
  +---------------------------------------------+
```

Following is the information CYRUS uses to choose indices when adding events to its "diplomatic meetings" MOP.

'diplomatic meetings'

norms:
participants: diplomats of countries involved in contract being discussed
location: conference room in capital city of country of an important participant
topic: international contract
duration: one to two hours

predictive:
political roles of the participants
classes those roles fit into
nationalities of the participants
occupations of the participants
political leanings of participants
topic, place, participants
sides of the topic
issue underlying the topic (e.g., peace)

non-predictive features:
participants' occupation is for. min.
participants' occupation is head of state

Using this knowledge, the first step of the index selection process would choose the following as potentially predictive features of EV1:

EV1: Cyrus Vance has a meeting with Andrei Gromyko in Russia about SALT.

1. **the meeting is with a foreign minister**
2. **the meeting is with a diplomat**
3. **the meeting is with a Russian**
4. **the meeting is with a Communist**

5. the meeting is about SALT
6. the topic concerns the U.S. and Russia
7. the underlying topic is arms limitations
8. the meeting is with Gromyko
9. the meeting is in Russia

Taking into account the norms and non-predictive aspects of the "diplomatic meetings" MOP above, only features 3 through 9 would remain as plausible indices for EV1 after the second step.

5.2 Generalization

Generalization is important to both the retrieval and updating processes. It is the process which builds up an E-MOP's norms. Retrieval strategies are guided by generalized knowledge. During memory update, a MOP's norms contrain later indexing and creation of new E-MOPs, thus preventing combinatorial explosion of indices.

5.2.1 Initial generalization

Initial generalization happens when the second event with a particular feature is added to an E-MOP. At that point, there are two events available for comparison -- the current one and the one already in memory. A new E-MOP is created, and the common features of the two events are added to the norms of the new E-MOP.

Consider two trips that Vance might have gone on to the Middle East: one to Israel, and one to Egypt, both to negotiate Arab-Israeli peace. Suppose that in both he talked to the head of state of the country he was visiting and was treated to a state dinner. If both of those events were indexed as "trips to the Middle East" in the "diplomatic trips" MOP, their similarities and differences would be reflected in the norms and indices of the newly-created MOP as follows:

```
        'diplomatic trips to the Middle East'

norms:  destination is the Middle East
        purpose is to negotiate Arab-Israeli peace
        includes meeting with head of state
        includes state dinner
        specialization of "diplomatic trip"
differences:          |
                 destination
                   /      \
               Israel     Egypt
```

5.2.2 Adjusting the certainty of a generalization

The first two events added to an E-MOP are special. They initiate the set of generalizations that will be used in future indexing and retrieval. Some initial generalizations are more reasonable than others, however. All meetings indexed in the E-MOP above, for example, will have the feature "destination is the Middle East" since that is the index for this sub-MOP in "diplomatic trip". Probably, these trips will continue to have the purpose of negotiating Arab-Israeli peace, at least as long as there is no peace there. We would not, however, expect that every trip to the Middle East will include a state dinner. There may also be attributes of trips to the Middle East not common to both of these trips. As additional meetings are added to the E-MOP, the unreasonable generalizations must be discovered and removed from the MOP. In addition, new events must be monitored to see if additional generalized information can be extracted from them.

5.2.3 Making additional generalizations

Because the first two events indexed in an E-MOP might fail to imply a particular generalization, MOP norms and indices must be monitored after initial generalization. In CYRUS, after an E-MOP reaches a reasonable size (6), CYRUS checks each sub-MOP referred to by incoming events to see if any of them index a large majority of the events in the E-MOP. If one does, CYRUS collapses it and merges its generalizations with those of the parent E-MOP, thus adding to the generalized information associated with that MOP.

5.2.4 Recovery from bad generalizations

Recovery from bad generalizations is more complex. When new information and events contradict a previously made generalization, that generalization must be removed as one of the E-MOP's norms.

This raises a special problem. While a feature is a norm of an E-MOP, events can never be indexed by that feature. In addition, if a feature is one of an E-MOP's norms, then some events ought to have it as a feature. Because events were not indexed by that feature, however, it would be impossible to go back and find all events supporting the generalization.

Generalization removal, then, can have grave implications in retrieval. If a retrieval specification specified a feature that had been removed as a generalization, but which had not yet been indexed, then the retrieval processes would not be able to find any trace in memory that an event with that feature had ever been processed. It would have to conclude that there had never been such an event in memory.

The solution to this is to create an index to an empty sub-MOP each time a feature is removed from a MOP's generalized information, and in addition, to mark the feature as having been "once generalized". That way, the retrieval functions will be able to come back with the message "there may be events with this description, but I can't find particular ones", instead of failing completely if no distinct event could be found. During later indexing, that sub-MOP will be treated like any other.

We can now point out an important need for search strategies as part of the retrieval process. Although false generalization on one E-MOP might keep a particular event from being well-indexed, related events might have been more richly indexed. If that is the case, they will be easier to retrieve than the event whose features had been falsely generalized. Finding a "once-generalized" E-MOP during traversal, then, should signal that search strategies will be particularly appropriate.

6. CYRUS Itself

CYRUS has two data bases — one each for former U.S. Secretaries of State Cyrus Vanoe and Edmund Muskle. CYRUS takes oonceptual representations of episodes as Input. Thus, representations of episodes must be built before sending them to CYRUS. CYRUS has two modes of receiving representations of stories. In one mode, the stories are analyzed and the representations encoded by the human reader before being integrated into CYRUS' memory. In its second mode of operation, CYRUS is connected to FRUMP [1] to form a complete information retrieval system called Cyfr [31. FRUMP reads stories from the UPI news wire, and sends conceptual summaries of stories about Muskle and Vance to CYRUS. CYRUS then adds the new events to its memory and answers questions about them. CYRUS' Muskle memory has been built up entirely from FRUMP-processed stories. Its Vanoe memory is built partially of FRUMP-processed stories and partially of stories encoded by hand.

The following is a story CyFr has processed about Muskle. FRUMP produoed the summary, and sent Its oonceptual representation to CYRUS. After adding the events to its memory, CYRUS answered the questions:

Carter begins going from the United States to Italy and Yugoslavia to talk. Secretary of State Edmund Muskle will go from the United States to Asia this month to have talks with ASEAN. Muskle will have talks with NATO in Ankara in June.

****

```
>Have you been to Europe recently?
 YES, MOST RECENTLY LAST MONTH.
>Why did you go there?
 TO TALK TO ANDREI GROMYKO.
>Are you going to Asia?
 YES, THIS MONTH.
>Who will you talk to?
 WITH NATO IN ANKARA, TURKEY.
```

The Vanoe and the Muskle memories start out the same, but after adding events to the two data bases, their organizations differ in four ways: (1) The indices are different. (2) The types of indloes are different. While the Vance E-MOP has topic Indloes and larger episode indices, the Muskle E-MOP haa neither of those. (3) The norms of their corresponding E-MOPs are different, (i.e., different generalisations have been made). (4) The Vanoe E-MOP indexes mostly sub-MOPs, and the Muskle E-MOP Indexes mostly individual events.

Three factors oontribute to these differences. First, the experienoes the two men have had are different. This is the reason for differences between Indloes in corresponding E-MOPs. Second, the data entered into the Vanoe data base is much more detailed than that entered into the Muskle memory. This faotor aooounts for the differences in the types of Indloes in the two memories. Because the Muskle memory is not usually aware of the topics of Muskle's meetings, for example, it oannot index them by aspeots of their topios.

The third faotor whioh aooounts for differences between the two memories is the degree of similarity between the events. The first ten events added to the Vance E-MOP, for example, were very similar to each other. Eight of them were meetings about the Camp David Accords. On the other hand, except for three meetings with Gromyko the meetings entered into the Muskle data base had very different participants and locations.

This factor accounts for differences (3) and (4) above. The more similarities between events in an E-MOP, the more filled out the MOP's generalized information can be. In addition, the more similar events in an E-MOP are, the more sub-MOPs will be indexed in the MOP than individual events. The extent of new category creation, then, is a function of the degree of similarity between items added to an E-MOP, and not on the number of items it organizes.

CYRUS Itself is no longer being developed as a computer system. Many of the problems illuminated by work on CYRUS are being investigated in the context of other research projects. The generality of CYRUS* retrieval and organizational strategies is being examined in two new areas — a world affairs expert and a medical diagnosis program. We are investigating more sophisticated methods for index selection and new category creation, and memory reorganization based on retrieval. The behavior of reconstructive memory as a memory organization for expert domains is also being explored.

7. Referenoes

[1] DeJong, G. F. (1979). Prediction and substantiation: a new approach to natural language processing. Cognitive Science, Vol. 3. PP. 251-273.

[2] Kolodner, J. L. (1980a). Organizing memory and keeping it organized. In Proceedings of the First Annual National Conference on Artificial Intelligence, Stanford, CA.

[33 Kolodner, J. L. (1980b). Retrieval and organizational strategies in conceptual memory: A computer model. Research Report #187. Dept. of Computer Science, Yale University, New Haven, CT.

[4] Lebowltz, M. (1980) Generalization and memory in an Integrated understanding system. Research Report #186. Dept. of Computer Science. Yale University, New Haven, CT.

[5] Norman, D. A. and Bobrow, D. G. (1977). Descriptions: a basis for memory acquisition and retrieval. Report #7703. Center for Human Information Processing, La Jolla, California.

[6] Schank, R. C. (1980). Language and memory. Cognitive Science. Vol. 4, No. 3.

[71 Schank, R. C. and Kolodner, J. L. (1979). Retrieving Information from an Episodic Memory Research Report #159. Department of Computer Solenee, Yale University, also in IJCAI-79.

[8] Smith, E. E., Adams, N., and Sohorr, D. (1978). Fact Retrieval and the Paradox of Interference, Cognitive Psychology. Vol 10, pp.438-464,

[9] Williams, M. D. (1978). The prooess of retrieval from very long term memory. Center for Human Information ProoessIng Memo CHIP-75, La Jolla, California.