View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by NASA Technical Reports Server

N89-26581

An English Language Interface For Constrained Domains

Brenda J. Page
Unisys Corporation
4325 Forbes Boulevard
Lanham, Maryland 20706

The Multi-Satellite Operations Control Center (MSOCC) Jargon Interpreter (MJI) demonstrates an English language interface for a constrained domain. A constrained domain is defined as one with a small and well delineated set of actions and objects. The set of actions chosen for the MJI is from the domain of MSOCC Applications Executive (MAE) Systems Test and Operations Language (STOL) directives and contains directives for signing a crt on or off, calling up or clearing a display page, starting or stopping a procedure, and controlling history recording. The set of objects chosen consists of crts, display pages, STOL procedures, and history files. Translation from English sentences to STOL directives is done in two phases. In the first phase, an augmented transition net (ATN) parser and dictionary are used for determining grammatically correct parsings of input sentences. In the second phase, grammatically typed sentences are submitted to a forward-chaining rule-based system for interpretation and translation into equivalent MAE STOL directives. Tests of the MJI show that it is able to translate individual clearly stated sentences into the subset of directives selected for the prototype. This approach to an English language interface may be used for similarly constrained situations by modifying the MJI's dictionary and rules to reflect the change of domain.

The work described in this paper was done under contract to the National Aeronautics and Space Administration Goddard Control Center Systems Branch.

PRECEDING PAGE BLANK NOT FILMED

Introduction

One of the issues in the Multi-Satellite Operations Control Center (MSOCC) environment is the existence of multiple System Test and Operations Language (STOL) dialects. The implementations of STOL in the various MSOCC control centers, such as the past developments of the Earth Radiation Budget Satellite (ERBS) and the Solar Maximum Mission (SMM) projects and the recent developments of the MSOCC Applications Executive (MAE) for the Cosmic Background Explorer (COBE) and Gamma Ray Observatory (GRO) projects and the Data Operations Control System (DOCS) project, have been done independently giving rise to different STOL dialects. The existence of multiple STOL dialects prevents MSOCC users from easily manipulating multiple systems. Artificial intelligence (AI) techniques, such as augmented transition nets (ATNs) and rule-based systems, have been used to build a prototype for a man-machine interface (MMI) common to all MSOCC systems. This prototype, called the MSOCC Jargon Interpreter (MJI), is an English language interpreter. The objective of the MJI is to demonstrate a solution which can alleviate the user s need to know the syntax of any STOL dialect and allow him to concentrate on directing the system to perform the functions for which it was designed.

Since the same activities are preformed over and over in MSOCC, a subset of English language, which we have called MSOCC jargon, is used over and over again to describe those actions. Some of the common actions include starting or stopping objects such as crts, display pages, STOL procedures, and history files. For the prototype, the domain is constrained to the STOL directives (STOL, PAGE, START, KILLPROC, and HISTORY) to perform these actions. The STOL dialect produced by the MJI is the one common to COBE, GRO, and other projects based on MAE.

The MJI contains three major components (figure 1). The first part of the MJI is a context free grammar (cfg) ATN parser which accepts and parses imperative sentences. Sentences are checked by the parser to make sure they are grammatically correct. The parser also identifies sentences' verbs, direct objects, and prepositional phrases and types each word in the sentences as a verb, determiner, adjective, noun, preposition, or conjunction. A dictionary lists the words accepted by the MJI and the parts of speech they can fill.

The second component of the MJI is a rule-based system. Facts from typed sentences are placed into a working knowledge base. An inference engine uses the working knowledge base and sets of rules to determine the meanings of sentences and produce equivalent STOL directives.

The third component of the MJI is an explanation system which allows the user to see the way sentences are grammatically typed by the parser and which rules are used by the inference engine to create STOL directives.

ORIGINAL PAGE IS
OF POOR QUALITY

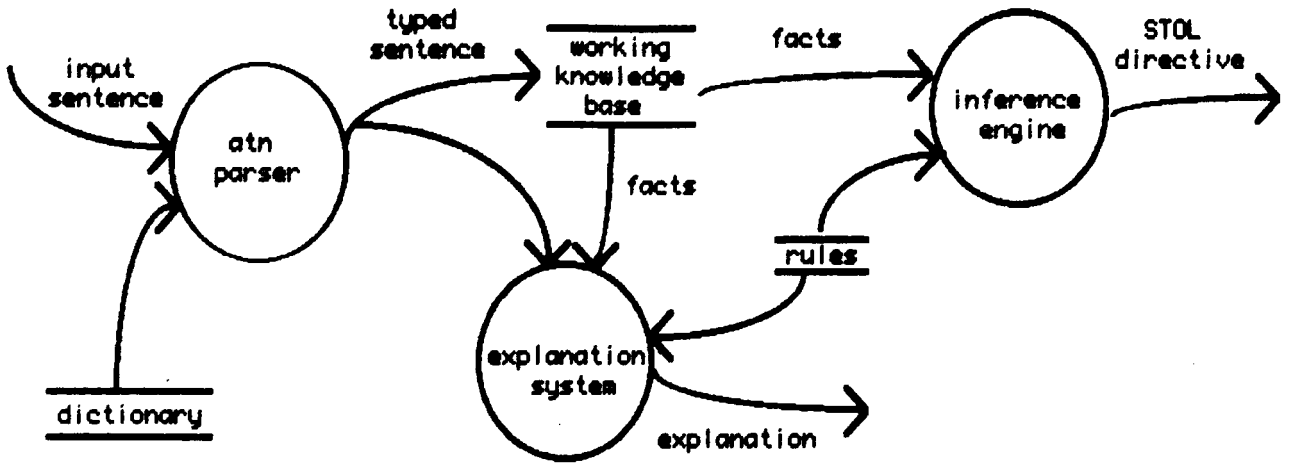

Figure 1. MJI components

The MJI is divided into three parts. The first is an ATN parser which accepts input sentences and grammatically types each word according to its usage (verb, noun, etc.) in the sentence. Typed sentences are then sent to an inference engine which uses rules and a working knowledge base to produce STOL directives. The explanation system allows the user to see the way sentences are grammatically typed and which rules are used to create STOL directives.

The parser

Because of the semantics of the MSOCC environment, only present tense imperative sentences containing a verb, a direct object target of the verb, and prepositional phrases modifying the action taken on the direct object are accepted by the parser. This restriction supports the type of work done in MSOCC. Commands to a system are made with the expectation that they will be carried out immediately. There is no stated subject in the imperative sentences accepted by the prototype - the subject is implied to be the target system.

The knowledge representation scheme chosen to represent context free imperative sentences is an ATN in conjunction with a dictionary. For grammatically typed sentences produced by the ATN, the knowledge representation scheme chosen is frames. Frames contain slots for values and are placed into a hierarchy.

An ATN is a set of states and transitions between the states used to parse an input string to determine whether or not the string is legal under the grammar defined by the ATN. The knowledge of the construct of a legal sentence is kept in the form of the ATN. Figure 2 shows part of the ATN used in the MJI. ATNs differ from traditional nets, which

also consist of stated and transitions, in that they have the capability to take notes as they are traversed and refer to the notes as the traversal continues or when it is done. The note taking feature is used to fill frames containing a sentence's structure of verb, direct object and prepositional phrases.

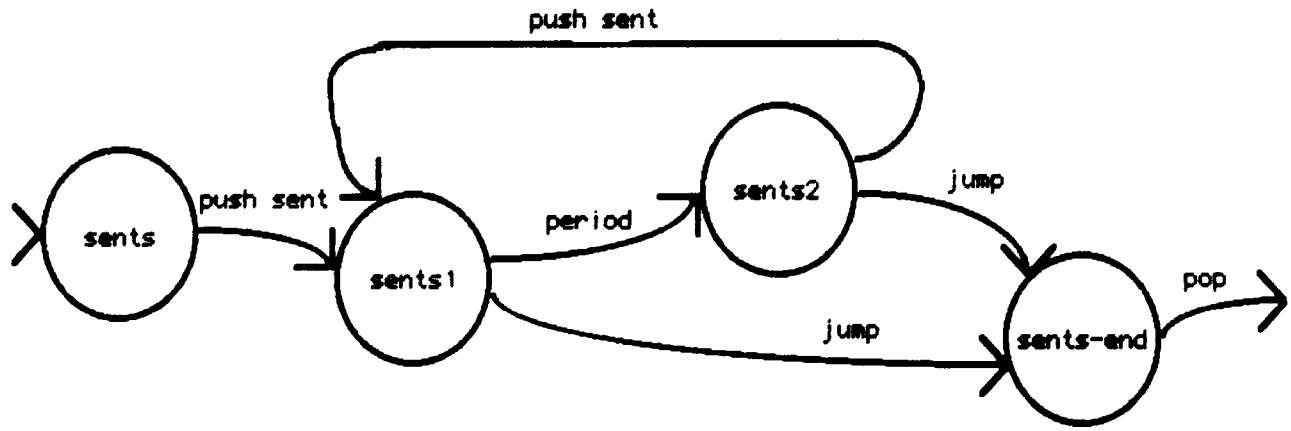

Figure 2. ATN nodes

The sentences ATN allows multiple sentences to be entered separated by periods. Each individual sentence is checked by the sentence ATN via the "push sent" transition. When all sentences are processed, a jump is made to sents-end which returns via a pop. Table 1 shows the definitions for these nodes and figure 3 shows part of the sentence ATN.

An ATN compiler is used to compile ATN node definitions (table 1) into code, each node becoming a procedure. The ATNs are compiled such that traversal proceeds in a depth first search. When a node is executed, it places all next nodes which can legally be reached on the front of a queue. A control mechanism pops the next node off the queue so it can be executed.

The ordering of links leaving a state plays an important role in determining the efficiency of the parser. The links are ordered so the one most likely to lead to a solution is checked first. For example, in the sentence ATN (figure 3) the first part of a sentence can be either a verb or a prepositional phrase. Since the occurrence of imperative sentences beginning with a prepositional phrase is predicted to be less than imperatives beginning with a verb, the verb path is always checked first.

ORIGINAL PAGE IS
OF POOR QUALITY

Table 1. ATN nodes

The ATN compiler processes each ATN node definition creating procedures. Figure 2 shows the ATN corresponding to these definitions.

```
(atn-node sents
      (push sent sents1))
            ; Push sent ATN so it may be traversed.

(atn-node sents1
      (cat period t sents2 nil)
            ; If the next word is a period, consume the word and move to sents2.
      (jump sents-end))
            ; Move to sents-end.

(atn-node sents2
      (push sent sents1)
            ; Push sent ATN so it may be traversed.
      (jump sents-end))
            ; Move to sents-end.

(atn-node sents-end
      (popit
            ; Pop up a level after the following statements are executed.
      (progn
        (addr new-regs ss-reg
          (build-sentences-frame (getr new-regs s-reg)))
            ; Build a sentences frame and place it in a register.
        (setq good-grammar
          (append good-grammar (getr new-regs ss-reg)))
           ; Add the new frame to a global.
        (remr new-regs s-reg))))
            ; Remove the register containing the frame created by the sent ATN.
```

As a sentence is parsed its parts - verb, direct object, and prepositional phrases - are identified and placed into sentence structure frames (figure 4). A verb is the basic requirement for a sentence; a direct object is not always necessary and neither are prepositional phrases. When a verb is precessed by the parser, it's sense is looked up in a dictionary and placed in the verb frame. Verb senses allow different verbs with similar meanings to be grouped under one term. Within the direct object phrase, there is an optional determiner, one or more optional adjectives, and one or more nouns. The only conjunction for connecting adjectives or nouns is 'and.' Multiple sentences may be entered at once as long as they are separated by periods. Figure 5 shows some legal and illegal sentences that can be entered and figure 6 shows a sentence placed in sentence structure frames.

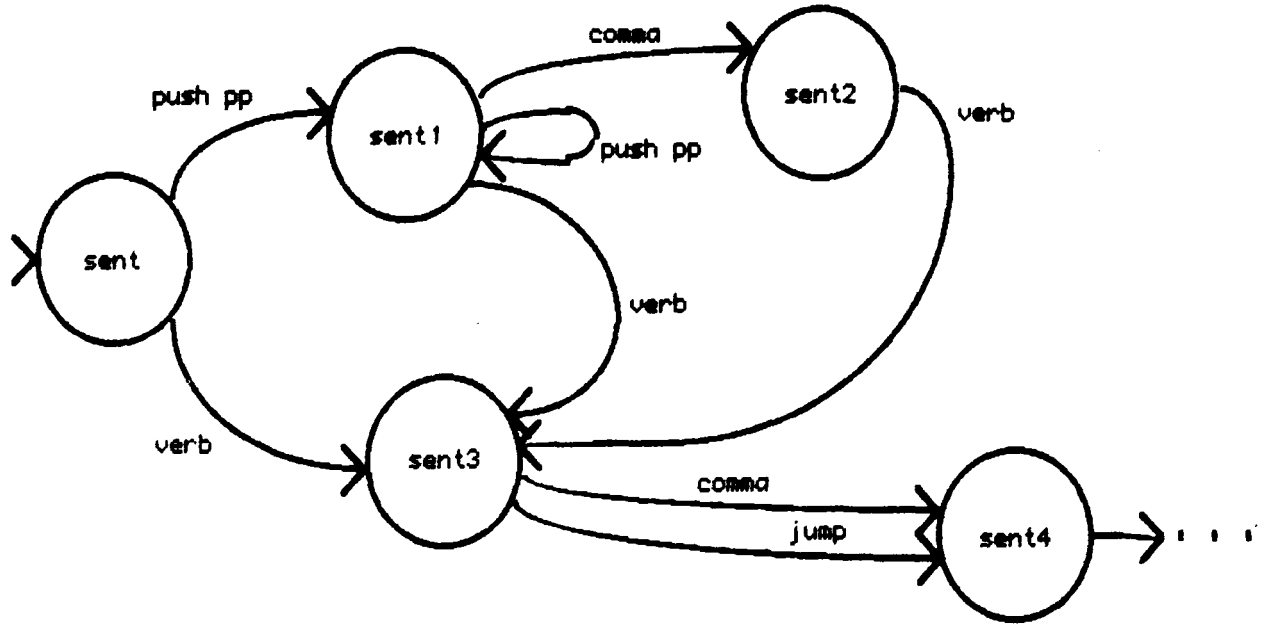

Figure 3. Sentence ATN (partial)

The nodes for all ATNs are ordered so that the path most likely to be successful is tried first. In the setences ATN, the verb transition is checked before a preposition transition. Transitions to next states are placed on a LIFO stack which is later popped to determine the next node to execute. So the information about the "push pp" transition from node sent is placed on the stack before information about the "verb" transition. When the stack is popped the "verb" transition is on top and tried first.

The knowledge about words is kept in a dictionary which lists the words allowed by the MJI and the parts of speech they may fill. Figure 7 contains some dictionary entries. The dictionary contains two special entries - *number and *char-string. *number is used to allow numbers within sentences. Whenever a number is specified in a sentence, the *number entry is used for data dictionary lookups. The other special entry is *char-string. Any time a word cannot be found in the dictionary and the word is not a number, the *char-string entry is used for the word. This feature is necessary to allow for words which cannot be entered into the dictionary ahead of time - such as the name of a newly created wildcard page or STOL procedure.

Because many words in the English language can fill the role of more than one part of speech, it is possible for a single sentence to have more than one grammatically correct interpretation. In the MJI, the parser outputs a single grammatically correct interpretation of a sentence. If the interpretation produced cannot be changed into directives, another grammatically correct interpretation is produced (if one exists). If a sentence is ambiguous, the first parsing produced is not necessarily the correct parsing.

ORIGINAL PAGE IS
OF POOR QUALITY

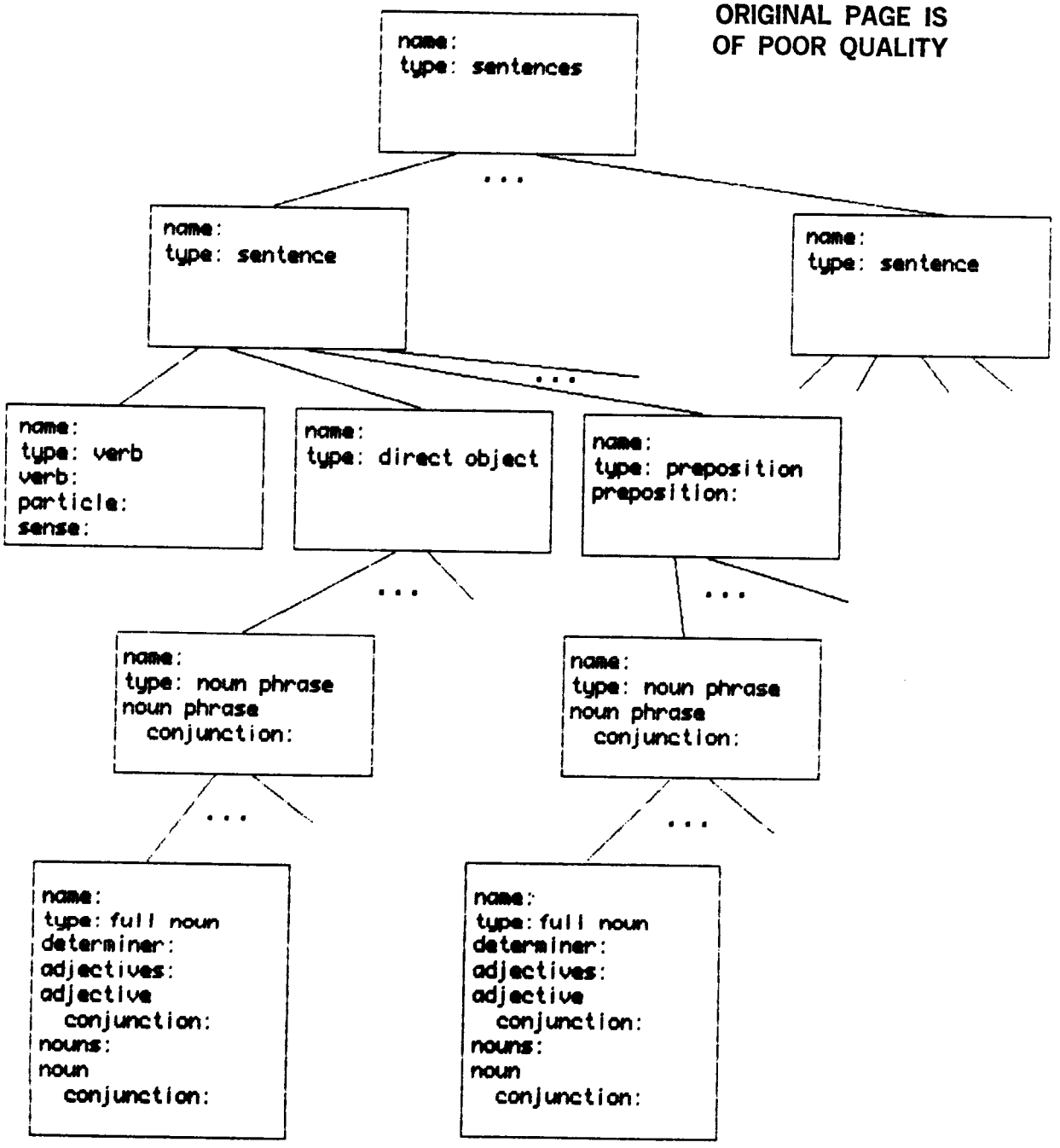

Figure 4. Blank sentence structure

As input is processed by the ATN parser it is placed in sentence structure frames.

Legal sentences

sentence	directive(s) created
Sign on.	STOL ON
Bring up the event page on crt 7.	PAGE EVENT 7
With parameter x, begin procedure test.	START TEST (X)
Bring up the NCC page and procedure test.	PAGE NCC START TEST
Start recording history data.	HISTORY ON

Illegal sentences

sentence	reason
Bring	no direct object
Start the	incomplete direct object
Get the event page with	incomplete prepositional phrase
Graph test	graph is not in the dictionary

Figure 5. Legal and illegal sentences

Examples of sentences accepted and not accepted by the ATN parser are listed.

The rule-based system

Once a sentence is successfully parsed and placed in sentence structure frames, it's meaning is determined by submitting the frames to a rule-based system (figure 8). First pertinent information from the frames, such as the verb sense and direct object adjectives and nouns, is moved into a working knowledge base of fact name and value pairs (figure 9).

The inference engine is forward-chaining and does a depth first search for a solution. It examines rules to see if any can be fired. When the conditions in the antecedents of rules are met by the current set of facts in the working knowledge base, the actions specified in the consequents of the rules are executed. The consequents are used to add facts to or modify facts in the working knowledge base. The consequents may also specify procedures which are called for their side effects of modifying the working knowledge base or printing results for the user. The inference engine keeps examining rules until a solution is found or no more rules can be fired. Figure 10 contains some rules.

The rules are split into several contexts each containing rules for specific functions. For example, the context PAGE-CHECK contains the rule page-check-1 and page-check-2 for determining if the direct object is a page and MAKE-START-DIR contains the rules make-

ORIGINAL PAGE IS
OF POOR QUALITY

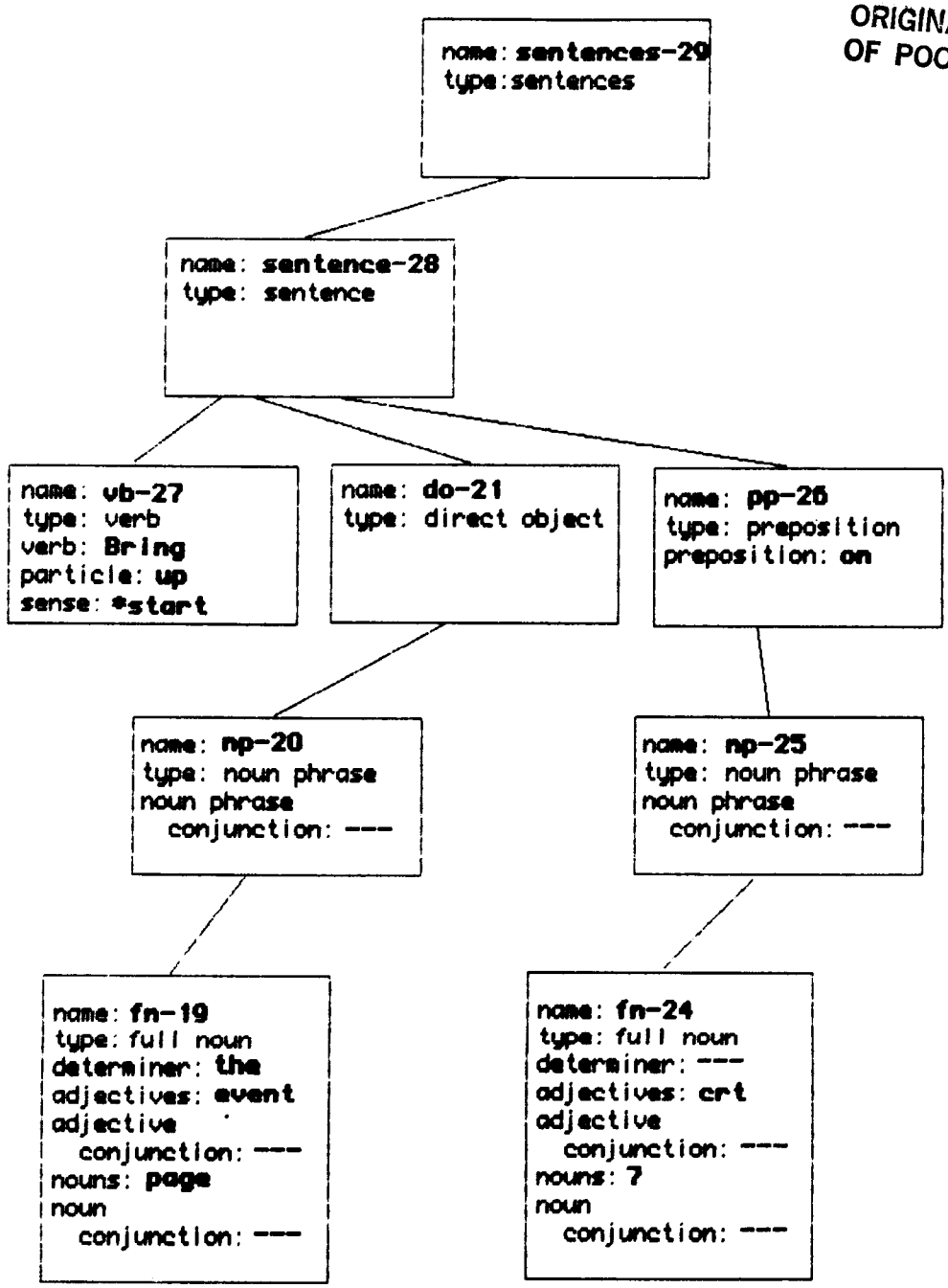

Figure 6. Filled sentence structure frames

The sentence "Bring up the event page on crt 7" is placed into sentence structure frames as it is parsed.

word	category	value(s)
begin	word-class	verb
begin	sense	*start
clear	word-class	verb
clear	sense	*clear
crt	word-class	noun, adjective
crt	number	singular
crts	word-class	noun, adjective
crts	number	plural
*char-string	word-class	noun, adjective
*char-string	number	singular, plural
*number	word-class	noun, adjective
*number	number	singular, plural

Figure 7. Dictionary entries

Each word in the dictionary has a word class entry indicating the part of speech the word may fill. Verbs also have a sense entry and nouns and adjectives have a number entry indicating whether the word is singular or plural.

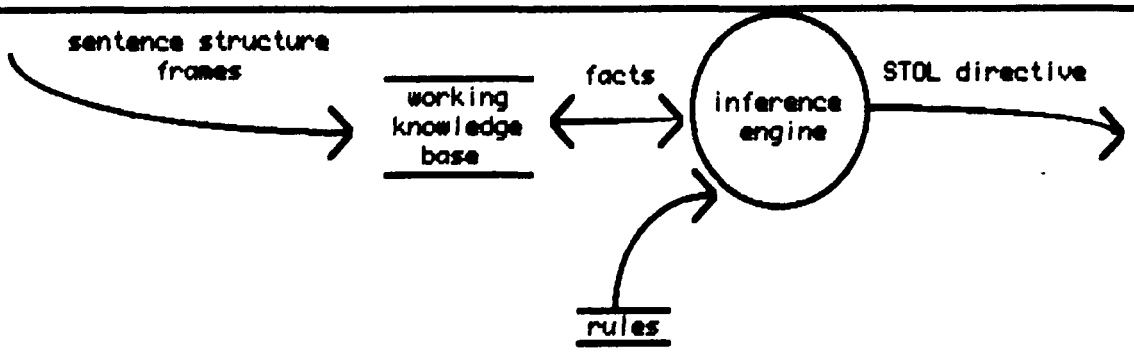

Figure 8. Rule-based system

Information from sentence structure frames is moved into the inference engine's working knowledge base. The engine then uses the working knowledge base to determine which rules should be fired to produce STOL directives.

start-dir-1 and make-start-dir-2 for making START directives. One of the facts in the working knowledge base tells the inference engine which rule context to use.

ORIGINAL PAGE IS
OF POOR QUALITY

Working knowledge base

fact name	fact value
SENSE	*START
ADJS	EVENT
NOUN	PAGE
PREP-PHRASES	*:|pp-26|
CURRENT-CONTEXT	WHICH-DO-CHECK1

Figure 9. Working knowledge base

The working knowledge base contains fact-value pairs. The facts making up the initial working knowledge base for the sentence "Bring up the event page on crt 7" are shown. This information is used to determine which rules may be fired (when the data in the working knowledge base make rules' antecedents true) by executing the rules' consequents (which may cause changes to the working knowledge base).

The explanation system

The explanation system allows the user to see the steps the MJI went through to translate a sentence into STOL directives. After an interpretation is produced, the user may enter 'explain.' The MJI will then describe how the sentence was grammatically typed and list the rules which were fired to obtain the directives. The rules are printed in natural language format with substitutions made from saved copies of the working knowledge base for values in antecedents and consequents. Only the rules which directly lead to a solution are described. Figure 11 shows the explanation of the sentence 'Bring up the event page on crt 7.'

In order to give an explanation, the MJI saves the sentence structure frame of the last sentence processed. The MJI also saves the names of the rules fired on the path to a solution along with copies of the working knowledge base at the time the rules' antecedents were checked.

Conclusions

The MSOCC Jargon Interpreter demonstrates the successful translation of English sentences into STOL directives. The MJI operates in a constrained domain consisting of clearly defined objects (crts, display pages, STOL procedures, and history files) and definite operations upon each of the objects specifiable by the sentences' verbs. Additions to MJI's rules and dictionary can expand the capacity of the interpreter allowing for more directives.

```
(setq page-check1
          ; If the direct object type is either *page-known-strings or
          ; *page-unknown-strings, move to the obj-is-page context.
     '(if ((one-of dir-obj-type (*page-known-strings
                                 *page-unknown-strings)))
       then
          ((rep-fact current-context obj-is-page))))

(setq page-check-2
          ; If the current context is page-check, then move to the finished
          ; context because processing is complete.
     '(if ((is current-context page-check))
       then
          ((rep-fact current-context finished))))

(setq make-start-dir-1
          ; If there are prepositional phrases in the sentence, then call
          ; the prep-is-param function for it's side effects of adding
          ; START parameters (if any exist) to the working knowledge base
          ; and move to the make-start-dir1 context.
     '(if ((is prep-phrases anything))
       then
          ((check-func prep-is-param (prep-phrases))
           (rep-fact current-context make-start-dir1))))

(setq make-start-dir-2
          ; If the current context is make-start-dir, call the
          ; create-start-dir procedure for it's side effects of printing
          ; START directives for the user and move to the finished context.
     '(if ((is current-context make-start-dir))
       then
          ((create-start-dir (dir-obj-list) nil)
           (rep-fact current-context finished))))
```

Figure 10. Rules

Shown are some of the rules used by the MJI.

The combination of ATN parser and rule-based system can be used in similarly constrained domains. For example, the rules and dictionary of the MJI are being modified so it can be used as a front-end for a display creation prototype under development. The new prototype is a customized graphics editor which allows the creation of wildcard display pages for use in MSOCC.

```
Your original sentence was grammatically typed as follows.
  Verb
    verb: BRING particle: UP verb sense: *START
  Direct object
    determiner: (THE) adjective: (EVENT) nouns: (PAGE)
  Prepositional phrase
    preposition: ON adjectives: (CRT) nouns: (7)
Your original input was sent directly through the inference engine for translation
to STOL directives.
```

```
The current rule is WHICH-DO-CHECK1-2.
Since the value of variable CURRENT CONTEXT is WHICH-DO-CHECK1.
The old value of CURRENT-CONTEXT was replaced with WHICH-DO-CHECK2.
```

```
The current rule is WHICH-DO-CHECK2-1.
Since the value of variable SENSE is *START and is one of *CLEAR *CLOSE *DISPLAY
  *INIT *OPEN *PUT *RESET *START *STOP.
Since the value of variable NOUN is PAGE and is not null.
A function IS-PAGE was called with parameters
   ADJS:  EVENT
   NOUN:  PAGE
 to deduce the fact(s)
   DIR-OBJ-TYPE with value *PAGE-KNOWN-STRINGS
   DIR-OBJ-LIST with value EVENT.
The old value of CURRENT-CONTEXT was replaced with PAGE-CHECK.
```

```
The current rule is PAGE-CHECK-1.
Since the value of variable DIR-OBJ-TYPE is *PAGE-KNOWN-STRINGS and is one of
  *PAGE-KNOWN-STRINGS *PAGE-UNKNOWN-STRINGS.
The old value of CURRENT-CONTEXT was replaced with OBJ-IS-PAGE.
```

```
The current rule is OBJ-IS-PAGE-2.
Since the value of variable DIR-OBJ-TYPE is *PAGE-KNOWN-STRINGS.
Since the value of variable SENSE is *START and is one of *DISPLAY *INIT *OPEN
  *PUT *START.
The old value of CURRENT-CONTEXT was replaced with MAKE-PAGE-DIR.
```

(continued on next page)

Figure 11. Explanation example

When giving an explanation, the MJI first tells how the sentence was typed and then lists the rules executed to produce STOL directives. Substitutions from saved copies of the working knowledge base are made into the rules before they are displayed.

Acknowledgments

I would like to thank Henry Murray of NASA Goddard's Control Center Systems Branch and Mike Blackstone of Unisys for their support of the work described in this paper. I would also like to thank them and Joe Maynard of Unisys for reviewing this paper.

```
The current context is MAKE-PAGE-DIR-1.
Since the value of variable PREP-PHRASES is *:|pp-26| and is not null.
A function PREP-IS-CRT was called with parameters
    PREP-PHRASES:  *:|pp-26|
  to deduce the fact(s)
    PREP-OBJ-CRT with value *CRT-KNOWN-NUMBERS
    PREP-CRT-LIST with value 7.
A function PREP-IS-INTERVAL was called with parameters
    PREP-PHRASES:  *:|pp-10|
  but did not deduce any new facts.
The old value of CURRENT-CONTEXT was replaced with MAKE-PAGE-DIR1.
```

```
The current rule is MAKE-PAGE-DIR1-1.
Since the value of variable CURRENT-CONTEXT is MAKE-PAGE-DIR1.
A function CREATE-PAGE-DIR was called with parameters
    UP-PAGE
    DIR-OBJ-LIST:  EVENT
    PREP-CRT-LIST:  7
    PREP-INT-LIST:
  which created the STOL directive(s)
    PAGE EVENT 7
The old value of CURRENT-CONTEXT was replaced with FINISHED.
```

EXPLANATION COMPLETE

Figure 11. Explanation example (continued)

When giving an explanation, the MJI first tells how the sentence was typed and then lists the rules executed to produce STOL directives. Substitutions from saved copies of the working knowledge base are made into the rules before they are displayed.

References

Allen, J., Natural Language Understanding, Benjamin/Cummings, California, 1987.

Denning, P. J., J. B. Dennis and J. E. Qualitz, Machines, Languages, and Computation, Prentice-Hall, New Jersey, 1978.

UNISYS, Applicability Of Artificial Intelligence In The Multi-Satellite Control Center (MOSCC) Software Design Document, August 1988.

UNISYS, CCS-7SUG/0485, MSOCC Applications Executive (MAE) System Test And Operations Language (STOL) Programmers Guide, September, 1987.

Winograd, T., Language As A Cognitive Process Volume I: Syntax, Addison-Wesley, Massachusetts, 1983.

Winston, P. H., Artificial Intelligence, Addison-Wesley, Massachusetts, 1984.