# PARAMAX SYSTEMS CORPORATION: DESCRIPTION OF THE PARAMAX SYSTEM USED FOR MUC-4

*Carl Weir and Barry Silk* [1]
Valley Forge Labs
Paramax Systems Corporation
Paoli, Pennsylvania
weir@prc.unisys.com
215-648-2369

## INTRODUCTION

This paper describes the Paramax MUC-4 data extraction system, a system with an analysis component implemented in CLIPS that provides a level of text understanding that falls somewhere between what is possible with conventional information retrieval techniques and deep linguistic analysis.

The Paramax MUC-4 system is a reimplementation of ideas originally introduced in the KBIRD system, which was evaluated in MUC-3 [4, 5]. There are plans to incorporate linguistic analysis techniques into the system, but the current effort is focussed on maximizing the data extraction capabilities of simpler methods. The Paramax MUC-4 system is depicted in Figure 1.

## APPROACH AND SYSTEM DESCRIPTION

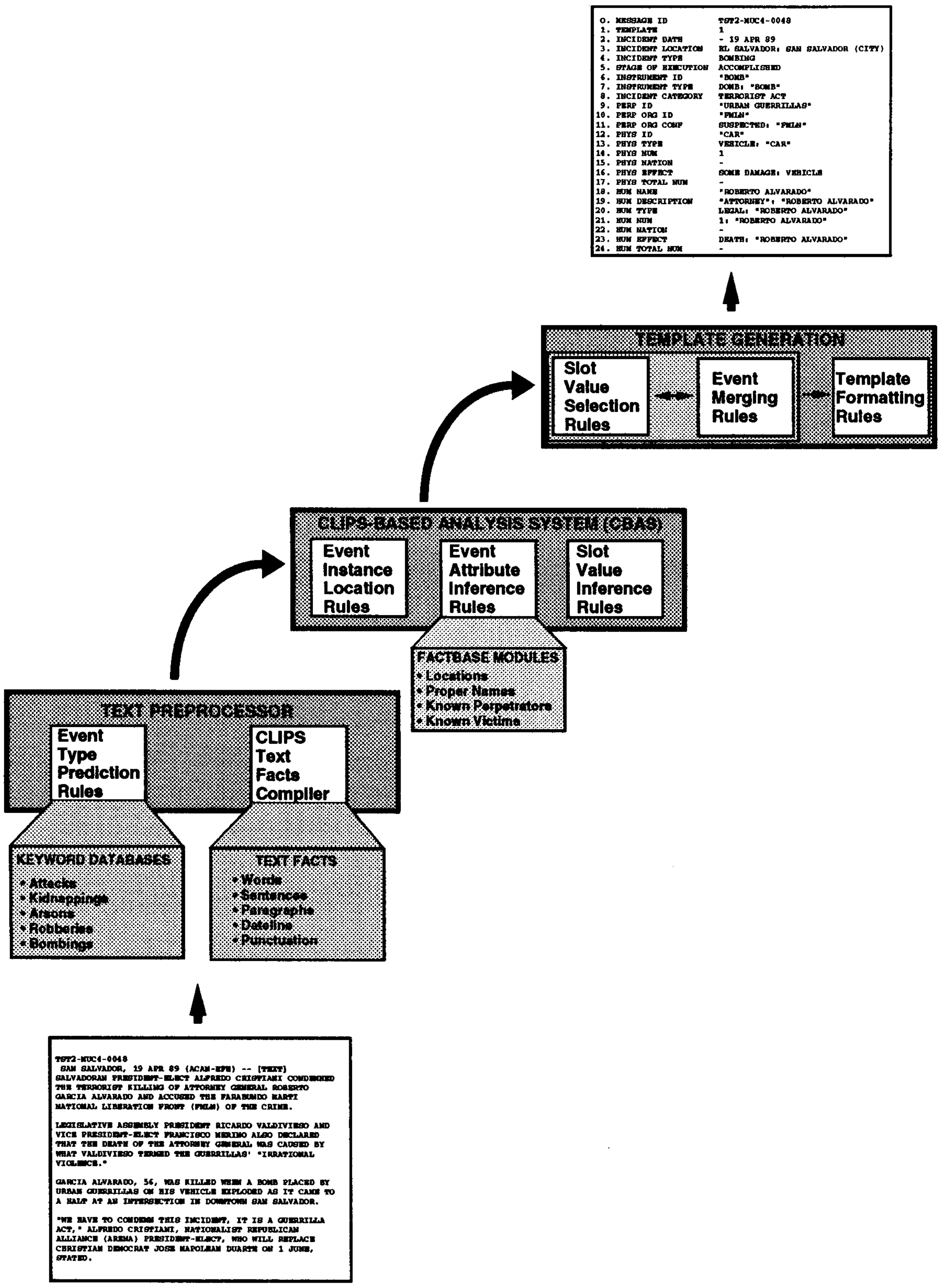
The Paramax MUC-4 system's architecture consists of three main processing components. An initial *preprocessing component* generates a set of CLIPS facts in which words, sentences, and paragraphs are identified. This component also computes the likelihood of relevant types of events being described in a given text. A core, rule-based *analysis component* implemented in CLIPS extracts whatever information can be inferred on the basis of the CLIPS facts generated by the preprocessor and the system's application-specific rule base. Finally, a third, *template generation* component sorts through all of the facts that have been inferred, merging descriptions of identical events and generating collections of templates in an appropriate format.

### The Preprocessing Component

The preprocessing component is implemented in a programming language specifically designed for processing textual data called PERL.[2] During preprocessing, words, sentences, paragraphs and punctuation marks are delimited. In addition, every word in the text is checked against a set of keyword profiles. These profiles are pairs of words and conditional probabilities representing types of events: arsons, attacks, bombings, kidnappings, robberies, and murders. Although murders are not treated as a basic type of event in the MUC-4 task, the system nevertheless attempts to identify them in order to predict significant deaths. If a word occurring in a text also occurs in one of the event profiles with a probability greater than 55%, then descriptions of instances of that type of event are asserted to likely occur in the text. Because the MUC-4 training sample is small, the list of words determined to be highly predictive of a given type of event is hand-edited to eliminate entries that have been assigned high scores because of sparse data.

The preprocessor's average elapsed processing time per text in the MUC-4 TST3 data set of 100 texts is less than 3 seconds. Total elapsed preprocessing time for the data set is 4 minutes, 8 seconds. It took approximately three days to write the preprocessor, and another three or four days to create and tune the event profiles it uses.

[1] Barry Silk is a U.S. government employee on sabbatical at Paramax.
[2] PERL is available at no cost; it can be FTPed from prep.ai.mit.edu.



```
 0. MESSAGE ID         TST2-MUC4-0048
 1. TEMPLATE           1
 2. INCIDENT DATE      - 19 APR 89
 3. INCIDENT LOCATION  EL SALVADOR: SAN SALVADOR (CITY)
 4. INCIDENT TYPE      BOMBING
 5. STAGE OF EXECUTION ACCOMPLISHED
 6. INSTRUMENT ID      "BOMB"
 7. INSTRUMENT TYPE    BOMB: "BOMB"
 8. INCIDENT CATEGORY  TERRORIST ACT
 9. PERP ID            "URBAN GUERRILLAS"
10. PERP ORG ID        "FMLN"
11. PERP ORG CONF      SUSPECTED: "FMLN"
12. PHYS ID            "CAR"
13. PHYS TYPE          VEHICLE: "CAR"
14. PHYS NUM           1
15. PHYS NATION        -
16. PHYS EFFECT        SOME DAMAGE: VEHICLE
17. PHYS TOTAL NUM     -
18. HUM NAME           "ROBERTO ALVARADO"
19. HUM DESCRIPTION    "ATTORNEY": "ROBERTO ALVARADO"
20. HUM TYPE           LEGAL: "ROBERTO ALVARADO"
21. HUM NUM            1: "ROBERTO ALVARADO"
22. HUM NATION         -
23. HUM EFFECT         DEATH: "ROBERTO ALVARADO"
24. HUM TOTAL NUM      -
```

```
TST2-MUC4-0048
 SAN SALVADOR, 19 APR 89 (ACAN-EFE) -- [TEXT]
SALVADORAN PRESIDENT-ELECT ALFREDO CRISTIANI CONDEMNED
THE TERRORIST KILLING OF ATTORNEY GENERAL ROBERTO
GARCIA ALVARADO AND ACCUSED THE FARABUNDO MARTI
NATIONAL LIBERATION FRONT (FMLN) OF THE CRIME.

LEGISLATIVE ASSEMBLY PRESIDENT RICARDO VALDIVIESO AND
VICE PRESIDENT-ELECT FRANCISCO MERINO ALSO DECLARED
THAT THE DEATH OF THE ATTORNEY GENERAL WAS CAUSED BY
WHAT VALDIVIESO TERMED THE GUERRILLAS' "IRRATIONAL
VIOLENCE."

GARCIA ALVARADO, 56, WAS KILLED WHEN A BOMB PLACED BY
URBAN GUERRILLAS ON HIS VEHICLE EXPLODED AS IT CAME TO
A HALT AT AN INTERSECTION IN DOWNTOWN SAN SALVADOR.

"WE HAVE TO CONDEMN THIS INCIDENT, IT IS A GUERRILLA
ACT," ALFREDO CRISTIANI, NATIONALIST REPUBLICAN
ALLIANCE (ARENA) PRESIDENT-ELECT, WHO WILL REPLACE
CHRISTIAN DEMOCRAT JOSE NAPOLEAN DUARTE ON 1 JUNE,
STATED.
```

**Figure 1**: A flowchart of the Paramax MUC-4 system.

## A CLIPS-Based Analysis System (CBAS)

After the text preprocessor has tokenized a text and has predicted which types of events are likely to be described, the task of extracting information about instances of the predicted event types falls to a CLIPS-based analysis system called CBAS. CLIPS is a forward-chaining system developed at NASA's Johnson Space Center that has received a lot of attention lately because of its speed and low cost. Rule-based systems similar to CLIPS have been used before to implement data extraction systems; two well-known implementations of this sort are the Carnegie Group's *Text Categorization Shell* [3] and the ADS *Rubric* system, which is a subcomponent of the *Codex* system evaluated at MUC-3 [1].

**CBAS rule format.** An important feature of CBAS rules is that the facts which the rules infer are associated with specific regions of text in very much the same way that edges in a parser's well-formed substring table are assigned to specific regions of an input string. In the KBIRD system referred to earlier, the region assigned to an inferred fact is automatically computed to be the maximal cumulative span of the regions of text associated with each expression in the antecedent of the rule making the inference. In CBAS, this region must be explicitly specified. Giving up the automated computation of the region provides more flexibility.

Unlike typical parsers, which contain an implicit constraint that adjacent constituents in a rule must be realized by contiguous strings of text in the input, CBAS requires all constraints to be explicitly encoded. In the earlier KBIRD system, constraints were realized using operators expressing relationships among the text regions associated with facts. In CBAS, relationships are expressed via constraints on attributes of facts rather than with operators taking facts as arguments.

The following example of a CBAS rule states that if the text preprocessor has predicted the likely occurrence of a bombing event and if the lexical item *dynamited* occurs in the text being analysed, then a bombing event instance is to be predicted at the same location as the occurrence of the lexical item:

```
(defrule bombing-dynamited
 (control-fact (phase events))
 (probable_event (type "BOMBING"))
 (word (lex "DYNAMITED")(pp ?P)(ss ?S)(left ?L)(right ?R))
 => (assert (bombing (id =(gensym))(key "DYNAMITED")(pp ?P)(ss ?S)(left ?L)(right ?R))))
```

The following two CBAS rules illustrate how rudimentary phrases can be constructed. In this case, proper names are recognized through examination of a database of known names, and proper names that are next to one another are concatenated together to form compound phrases.

```
(defrule assert-proper-name
 (control-fact (phase hum-proper))
 (word (lex ?LEX)(pp ?P)(ss ?S)(left ?L)(right ?R))
 (known_name (lex ?LEX))
 =>
 (assert (proper_name (lex ?LEX)(pp ?P)(ss ?S)(left ?L)(right ?R))))

(defrule assert-proper-name-phrase
  (control-fact (phase hum-proper))
  ?f1 <- (proper_name (lex ?LEX1)(pp ?P)(ss ?S)(left ?L1)(right ?R1))
  ?f2 <- (proper_name (lex ?LEX2)(pp ?P)(ss ?S)(left ?R1)(right ?R2))
  =>
  (assert (proper_name (lex =(format nil "%s %s" ?LEX1 ?LEX2))(pp ?P)(ss ?S)(left ?L1)(right ?R2)))
  (retract ?f1)
  (retract ?f2))
```

**Processing Phases/Rule modules.** Following standard practice in forward-chaining system development, the antecedent portions of CBAS rules include references to *control fact* statements. These control facts are asserted and retracted during the processing of a text to enable or disable portions of the Rete network constructed out of the system's rule base.[3] Through the use of control facts, different inferencing phases can be defined and the rules associated with these phases constitute separate modules, some of which may be used in other domains and/or applications. Three different types of rule modules arise:

**Event attribute rule modules.** Modules of this sort consist of rules which infer facts describing possible properties of events without associating those properties with any specific event instance. For example, a rule module exists whose only goal is to identify proper name expressions. Similarly, a rule module exists that identifies possible physical targets.

**Event instance rule modules.** Modules of this sort consist of rules which locate instances of types of events. There is only one module of this sort in the MUC-4 implementation, but multiple modules of this sort could exist. For example, it might be desirable to have separate modules for each event type.

**Slot value rule modules.** Modules of this sort consist of rules which infer possible slot values for templates describing event instances. Facts inferred by rules in this type of module are weighted.

**Key CBAS features.** The most significant features of CBAS are the following:

- CBAS is implemented in CLIPS, which can be acquired at little or no cost.[4]
- The implementation is reasonably fast, with an average processing time per text in the MUC-4 TST3 data set of 1 minute, 46 seconds. Total processing time for all 100 messages in the data set is 2 hours, 57 minutes.[5]
- The implementation was built quickly, in about 2 months with less than 4 person-months of effort.[6]
- The CBAS rule formalism is easy to learn and requires no linguistic expertise. Most of the rules written for the MUC-4 prototype were written by someone with no background in linguistics.

## A Template Generator

After the CBAS component has inferred whatever information can be extracted from a given text, a third module is responsible for generating templates based on the factbase that has been created. Three tasks are performed by the template generator:

1. It selects the actual templates that should be produced as output.
2. It chooses among candidate slot fillers if more than one filler has been found.
3. It prints the actual templates in the proper format.

[3] A Rete network is a data structure commonly used to encode information in forward-chaining systems. See Forgy [2] for an explanation of Rete networks.

[4] It can be acquired at no cost for NASA and USAF projects and at a marginal cost for all other uses ($490.00, including a three volume set of documentation).

[5] Total processing time including non-CLIPS processing (PERL preprocessing and PROLOG template generation) is $3\frac{1}{2}$ hours for the TST3 data set.

[6] We did not remember to actually *write down* the date we started. We have estimated two months because we know that CLIPS was installed at our site on 7 April 92 and we know that we stopped development on 30 May 1992. We had access to some old rulebases that were created for the MUC-3 KBIRD system, but these were expressed in a sufficiently different rule formalism to require a total rewrite. We estimate less than 4 person-months of effort because only two individuals were involved, and both individuals were required to perform other tasks in addition to their work on MUC-4.

**Template Selection.** The process of determining which template structures to build out of the facts inferred by CBAS begins by determining if any events at all have been predicted. If no event has been predicted, then an "irrelevant template" is created. If several events of the same type have been created, the template generator will attempt to merge them using a set of heuristics which hypothesize that two event descriptions refer to the same event. Some of the general heuristics used for merging events of the same class are the following:

- Merge two events if there is a significant overlap in the text regions found by the *event locator rules.*
- Merge two events if they share human targets whose scores are above a certain threshold.
- Merge two events if they share physical targets whose scores are above a certain threshold.

**Slot Filler Selection.** After merging events, the template generator must select the final slot filler values. The CBAS rules which propose slot fillers attach a score (an integer between 0 and 100) to each candidate which represents the system's confidence in that value. If multiple candidate fillers exist for a given template, several general heuristics are used to select among them:

- Candidate slot values with scores below a given threshold are dropped from consideration.
- A set of synonymous expressions are dropped in favor of their canonical expression.
- If one candidate expression is a substring of another, then the shorter one is dropped.
- A generic description (e.g., *vehicles*) is dropped in favor of one or more subsumed ones (e.g., *ambulance, truck*).
- If a slot can only take a single value then the candidate receiving the highest value is selected.

The template generator is fast, with an average processing time per text in the MUC-4 TST3 data set of 11 seconds. Total processing time for all 100 texts in the MUC-4 TST3 data set was 17 minutes, 44 seconds.

## AN EXTENDED EXAMPLE

In this section, we illustrate in a more concrete fashion how the Paramax MUC-4 system goes about processing messages by examining in detail what happens during the processing of a specific text, message TST2-MUC4-0048, in the MUC-4 corpus.[7] Our discussion will proceed through the three stages of text processing that have been identified.

### First Stage: Text Preprocessing

Figure 2 contains a sampling of the CLIPS text facts created during the preprocessing stage for TST2-MUC4-0048. The `msg_location`, `msg_date`, and `msg_src` facts are extracted from the text dateline. The `word` facts identify the locations of lexical items, and the `ss` and `pp` facts identify the locations of sentence and paragraph boundaries, respectively. For this message, three types of events were predicted: attacks, bombings, and murders.

### Second Stage: CLIPS-Based Analysis

At the start of the second stage of processing, the set of CLIPS facts generated during the preprocessing stage are asserted to the CBAS factbase. Once this is done, the forward-chaining engine is invoked to extract information.

[7]Appendix F contains the text and answer key templates for TST2-MUC4-0048.

```
(msg_location (id "SAN SALVADOR"))
(msg_date (day "19")(month "APR")(year "89"))
(msg_src (lex "ACAN-EFE"))
(word (pp 1) (ss 1) (left 0) (right 1) (lex "SALVADORAN"))
(word (pp 1) (ss 1) (left 1) (right 2) (lex "PRESIDENT-ELECT"))
(word (pp 1) (ss 1) (left 2) (right 3) (lex "ALFREDO"))
(word (pp 1) (ss 1) (left 3) (right 4) (lex "CRISTIANI"))
...
(word (pp 2) (ss 2) (left 26) (right 27) (lex "LEGISLATIVE"))
(word (pp 2) (ss 2) (left 27) (right 28) (lex "ASSEMBLY"))
(word (pp 2) (ss 2) (left 28) (right 29) (lex "PRESIDENT"))
(word (pp 2) (ss 2) (left 29) (right 30) (lex "RICARDO"))
...
(ss (num 1) (pp 1) (left 0) (right 26))
(ss (num 2) (pp 1) (left 26) (right 84))
(pp (num 1) (left 0) (right 26))
(pp (num 2) (left 26) (right 55))
...
(predicted_event_type (type "ATTACK"))
(predicted_event_type (type "BOMBING"))
(predicted_event_type (type "MURDER"))
```

**Figure 2**: Some of the preprocessing output for Message TST2-MUC4-0048.

Since linguistic analysis techniques are not incorporated into the system, information about subconstituent structure is limited. Constraints of this sort on the system's data extraction capabilities are *inherent constraints*. The system also contains *accidental constraints*, which are caused by engineering flaws in the system and not by inherent limitations in the extraction methods being employed.

It is often not obvious that a data extraction problem is a consequence of an inherent constraint or an accidental one, or if it is a consequence of some combination of inherent and accidental constraints. From the perspective of the MUC-4 conference, detecting inherent constraints is more interesting than detecting accidental constraints. Keeping this in mind, performance characteristics of the CBAS analysis component will now be described by examining some of the slot values extracted by the system for message TST2-MUC4-0048.

**Event instance location.** Because of the event type predictions made during the preprocessing stage, the system is primed to look for instances of attacks, murders, and bombings in TST2-MUC4-0048. In the CBAS system, each reference to a given type of event detected in a text is treated as if it is referring to a different event instance. It is up to the template generation component to determine that the templates describing different event instances are in fact describing the same event. A key problem with the MUC-4 implementation is that this identification process is not working well, leading to an excessive number of spurious values.

For TST2-MUC4-0048, the MUC-4 CBAS system generated 4 response templates. The first template describes a terrorist act identified as an attack, but for which a bomb is correctly identified as an instrument. Given the presence of a bomb as an instrument, the failure to identify the event as a bombing is clearly the consequence of an indequately developed rule base and not the result of any inherent constraint in the methodology being used.

It is important to keep in mind that many event instances are created during the data extraction process, and that the goal is to identify instances of the same actual event and merge their template structures. A number of bombing event instances were generated for this message, as expected given the number of references to bombs and explosions. However, since the event instances leading to the generation of the first response template were identified as attacks, their template structures were not merged with those of bombing instances. A total of three other spurious templates all of which were identified as descriptions of bombings, were generated for the sample text.

| # | Description | Filler |
|---|---|---|
| 0. | MESSAGE: ID | TST2-MUC4-0048 |
| 1. | MESSAGE: TEMPLATE | 1 |
| 2. | INCIDENT: DATE | 14 APR 89 |
| 3. | INCIDENT: LOCATION | EL SALVADOR: SAN SALVADOR (DEPARTMENT): SAN SALVADOR (CITY) |
| 4. | INCIDENT: TYPE | ATTACK |
| 5. | INCIDENT: STAGE OF EXECUTION | ACCOMPLISHED |
| 6. | INCIDENT: INSTRUMENT ID | "BOMB" |
| 7. | INCIDENT: INSTRUMENT TYPE | BOMB: "BOMB" |
| 8. | PERP: INCIDENT CATEGORY | TERRORIST ACT |
| 9. | PERP: INDIVIDUAL ID | "AN INDIVIDUAL" |
| 10. | PERP: ORG ID | "FARABUNDO MARTI NATIONAL LIBERATION FRONT" |
| 11. | PERP: ORG CONF | CLAIMED OR ADMITTED: "FARABUNDO MARTI NATIONAL LIBERATION FRONT" |
| 12. | PHYS TGT: ID | "HOME" |
|  |  | "CAR" |
|  |  | "ARMORED VEHICLE" |
| 13. | PHYS TGT: TYPE | CIVILIAN RESIDENCE: "HOME" |
|  |  | TRANSPORT VEHICLE: "CAR" |
|  |  | TRANSPORT VEHICLE: "ARMORED VEHICLE" |
| 14. | PHYS TGT: NUMBER | 1: "HOME" |
|  |  | 1: "CAR" |
|  |  | 1: "ARMORED VEHICLE" |
| 15. | PHYS TGT: FOREIGN NATION | - |
| 16. | PHYS TGT: EFFECT OF INCIDENT | SOME DAMAGE: "ARMORED VEHICLE" |
|  |  | SOME DAMAGE: "CAR" |
|  |  | SOME DAMAGE: "HOME" |
| 17. | PHYS TGT: TOTAL NUMBER | - |
| 18. | HUM TGT: NAME | "ROBERTO GARCIA ALVARADO" |
|  |  | "RICARDO VALDIVIESO" |
|  |  | "ALFREDO CRISTIANI" |
| 19. | HUM TGT: DESCRIPTION | "LEADER" |
|  |  | "JUSTICE" |
|  |  | "DRIVER" |
|  |  | "BODYGUARDS" |
|  |  | "URBAN TERRORISTS VICE PRESIDENT-ELECT" |
|  |  | "INDIVIDUAL" |
|  |  | "SALVADORAN PRESIDENT-ELECT" |
|  |  | "ATTORNEY GENERAL": "ROBERTO GARCIA ALVARADO" |
|  |  | "PRESIDENT": "RICARDO VALDIVIESO" |
|  |  | "PRESIDENT": "ALFREDO CRISTIANI" |
| 20. | HUM TGT: TYPE | POLITICAL FIGURE: "LEADER" |
|  |  | LEGAL OR JUDICIAL: "JUSTICE" |
|  |  | CIVILIAN: "DRIVER" |
|  |  | SECURITY GUARD: "BODYGUARDS" |
|  |  | GOVERNMENT OFFICIAL: "URBAN TERRORISTS VICE PRESIDENT-ELECT" |
|  |  | CIVILIAN: "INDIVIDUAL" |
|  |  | GOVERNMENT OFFICIAL: "SALVADORAN PRESIDENT-ELECT" |
|  |  | LEGAL OR JUDICIAL: "ROBERTO GARCIA ALVARADO" |
|  |  | GOVERNMENT OFFICIAL: "RICARDO VALDIVIESO" |
|  |  | GOVERNMENT OFFICIAL: "ALFREDO CRISTIANI" |
| 21. | HUM TGT: NUMBER | 1: "LEADER" |
|  |  | 1: "JUSTICE" |
|  |  | 1: "DRIVER" |
|  |  | 2: "BODYGUARDS" |
|  |  | PLURAL: "URBAN TERRORISTS VICE PRESIDENT-ELECT" |
|  |  | 1: "INDIVIDUAL" |
|  |  | 1: "SALVADORAN PRESIDENT-ELECT" |
|  |  | 1: "ROBERTO GARCIA ALVARADO" |
|  |  | 1: "RICARDO VALDIVIESO" |
|  |  | 1: "ALFREDO CRISTIANI" |
| 22. | HUM TGT: FOREIGN NATION | - |
| 23. | HUM TGT: EFFECT OF INCIDENT | NO INJURY: "LEADER" |
|  |  | NO INJURY: "JUSTICE" |
|  |  | NO INJURY: "DRIVER" |
|  |  | NO INJURY: "BODYGUARDS" |
|  |  | DEATH: "SALVADORAN PRESIDENT-ELECT" |
|  |  | NO INJURY: "ROBERTO GARCIA ALVARADO" |
|  |  | DEATH: "ALFREDO CRISTIANI" |
|  |  | DEATH: "RICARDO VALDIVIESO" |
| 24. | HUM TGT: TOTAL NUMBER | - |

**Figure 3:** The first response template generated by the Paramax MUC-4 system for TST2-MUC4-0048.

**Extracting perpetrator ids.** In the first response template generated for TST2-MUC4-0048, the string "AN INDIVIDUAL" was recorded as the perpetrator id, based on the occurrence of this phrase in the the clause "AN INDIVIDUAL PLACED A BOMB ON THE ROOF OF THE ARMORED VEHICLE". The correct string fill, "URBAN GUERRILLAS" was also extracted by CBAS, but the former value was assigned a higher score. The following ground clauses were generated by CBAS to represent these strings:

```
potential_ind_perpetrator('TERRORISTS','"URBAN GUERRILLAS"').
potential_ind_perpetrator('UNKNOWN','"AN INDIVIDUAL"').
```

Unfortunately, the type specification for "URBAN GUERRILLAS" should have been output as 'TERRORIST', not 'TERRORISTS'. Had this been done, then the following rule would have assigned a higher score to "URBAN GUERRILLAS".

```
(defrule perp-attack-events-3
  (control-fact (phase perp-events))
  (actual_event  (id ?id) (class "ATTACK") (ss ?ss))
  (potential_ind_perpetrator (type "TERRORIST") (name ?p) (ss ?ss))
  =>
  (assert (tmp (id ?id)(slot slot09)(val ?p)(prob 85))))
```

Since no rules exist for assigning probabilities to possible perpetrators of type "TERRORISTS", the following CBAS rule assigned a relatively low score to "URBAN GUERRILLAS" as a possible fill for slot 9:

```
(defrule perp-all-events-1
  (control-fact (phase perp-events))
  (actual_event (id ?id)(ss ?ss))
  (potential_ind_perpetrator (name ?p)(ss ?ss))
  =>
  (assert (tmp (id ?id)(slot slot09)(val ?p)(prob 50))))
```

The following rule assigned a much higher weight to "AN INDIVIDUAL":

```
(defrule perp-attack-events-6
  (control-fact (phase perp-events))
  (actual_event  (id ?id) (class "ATTACK") (ss ?ss))
  (potential_ind_perpetrator (type "UNKNOWN") (name ?p) (ss ?ss))
  =>
  (assert (tmp (id ?id)(slot slot09)(val ?p)(prob 75))))
```

This type of error in the rule base could have been caught with a bit more staffing and/or development time and is characteristic of relatively simple bugs that have a significant cumulative impact on data extraction performance.

**Extracting physical targets.** Two separate problems are manifested in the slot values proposed for physical targets in the first response template generated for TST2-MUC4-0048.

The first problem is evidenced by the failure of the system to determine that "CAR" and "ARMORED VEHICLE" are co-referential. The ability to establish co-referentiality is minimal in the CBAS system, and this weakness is probably inherent in the methodology. However, in this particular case, the CBAS rule base could be expanded to treat "CAR" and "ARMORED VEHICLE" as synonyms and therefore submit only one of them as a reference to a physical target. Were this done, the problem that would then arise is the need to recognize when two separate objects are in fact being talked about.

The second problem evidenced in the extraction of physical targets is the presence of "HOME" as a possible value, along with "CAR" and "ARMORED VEHICLE". It turns out that the CBAS rule base proposed two separate attack event instances, one for the bomb attack on Alvarado's car, and one for the bomb attack on Merino's home. However, the template generator component, which is responsible for merging templates describing the same event, erroneously merged the templates describing these two separate attack instances, combining their physical target slot values. Not enough time was available to develop template merging heuristics.

**Extracting Human Targets.** A number of spurious human target values are proposed in the first response template. Many of these spurious values have arisen because of a poorly tuned rule base and not because of any inherent contraints in the analysis techniques employed. If finer-grained rule weightings were to be employed so that strings describing human targets which are located relatively near a region of text triggering the detection of an event instance are preferred over others, then many of these spurious values would not be present.

The human targets "RICARDO VALDIVIESO" "ROBERTO GARCIA ALVARADO", "LEADER", "JUSTICE", "DRIVER", and "BODYGUARDS" were all proposed with equal likelihood as possible slot values for an event instance detected through the occurrence of the word "KILLED" in the phrase "VIOLENCE THAT KILLED ATTORNEY GENERAL GARCIA." If a higher score were assigned to human targets immediately following "KILLED", then the other values could be eliminated. This could most certainly be done, and in fact was done in the KBIRD implementation described earlier. For this type of finer-grained score assignment, it would make sense to employ a part-of-speech tagger that is able to distinguish simple past tense forms from past participles used in passive constructions.[8]

It is less clear how to reintroduce "DRIVER" and "BODYGUARDS" as slot values, given that they will not be introduced if the suggested improvements in rule scoring are made. It is in the following two sentences that it is made clear that the driver and bodyguards were also targets: "ACCORDING TO THE POLICE GARCIA ALVARADO'S DRIVER, WHO ESCAPED UNSCATHED, THE ATTORNEY GENERAL WAS TRAVELING WITH TWO BODYGUARDS. ONE OF THEM WAS INJURED." Rules could certainly be written to capture these additional human targets. For example, rules could be sensitive to the existence of individuals who "ESCAPE UNSCATHED" and to groups of individuals, some of whom are injured and that are "TRAVELING WITH" a highly likely human target. But it is not clear if one would ever be able to write enough rules in a given domain to reach closure in extracting all human targets that are referred to in this sort of indirect fashion.

The string "URBAN TERRORISTS VICE PRESIDENT-ELECT" arises as a reference to a human target because of a bug in one of the CBAS rules that builds up compound common noun phrases. In this case, the rule was not written to pay attention to sentence boundaries, and consequently the common noun "URBAN TERRORISTS" occurring at the end of one sentence and "VICE PRESIDENT-ELECT" occurring at the beginning of the next sentence were incorrectly identified as a single common noun expression. Correcting this type of error is easy.

The string "SALVADORAN PRESIDENT-ELECT" should have been recognized as a title of Alfredo Cristiani in the phrase "SALVADORAN PRESIDENT-ELECT ALFREDO CRISTIANI". A bug in one of the CBAS rules that recognizes titles next to proper names was to blame. If a common noun expression is recognized as a title, it is not proposed as a separate reference to an individual. The presense of Alfredo Christiani as a possible victim could have been eliminated by the improvement in scoring techniques mentioned earlier.

Finally, the string "INDIVIDUAL" arises as a reference to a human target. This error could have been eliminated by improved scoring techniques. Also, since "AN INDIVIDUAL" occurs as a perpetrator, improved template merging heuristics should have eliminated one of the values as a candidate filler.

[8]Unfortunately, we have been told that taggers generally perform poorly in discriminating passive past participles from simple past tense forms and participles used in past perfect constructions.

## Third Stage: Template Generation

Time and funding constraints made it impossible to implement a satisfactory template generation component for this system.

In the previous discussion of the analysis component's performance, references were made to cases in which the template generator merged template descriptions when it shouldn't have. In many cases, the opposite is true. For example, two of the four templates generated for TST2-MUC4-0048 are exact duplicates of one another. Moreover, the duplicates are relatively impoverished, containing no physical targets, and only one reference to "BODY" as a human target descriptor. With a little cooperation from the rule base to eliminate "BODY" as a human target, the generator should be able to justify not printing the underpopulated templates, and in any case, an existing heuristic for not printing exact duplicates of templates needs to be fixed.

## CONCLUSIONS

A key motivating factor in the design of the Paramax MUC-4 system has been a desire to exploit simple data extraction methods to the fullest extent possible. Another motivating factor has been the desire to build a relatively inexpensive system that individuals with no training whatsoever in linguistics could develop and maintain.

We do not believe that we have fully exploited the capabilities of non-linguistic data extraction methods and intend to continue to explore the simpler techniques used in the CBAS system. This intention is not based on a strongly-held desire to avoid the use of linguistic-analysis techniques in data extraction applications. Quite the contrary—our research group is actively involved in the development of linguistic analysis techniques and feels that they will ultimately be essential in data extraction applications. Our MUC-4 implementation is part of a long-term effort to better understand exactly when such techniques are in fact required.

Except for the template generator, which is written in Prolog, the Paramax MUC-4 system has been built using freeware (PERL and CLIPS). Each component is written in a different programming language, which is a strong indicator of its modularity.

Finally, we have demonstrated by example that the approach taken in the Paramax MUC-4 system does not require any linguistic expertise. A general knowledge of the domain is all that is required. Acquiring an understanding of the CBAS rule formalism and of CLIPS in general takes very little time—the two individuals that developed the MUC-4 system had no prior experience whatsoever with CLIPS.

## REFERENCES

[1] Laura Blumer Balcom and Richard M. Tong. Advanced decision systems: Description of the Codex system as used for MUC-3. In *Proceedings of the Third Message Understanding Conference*, pages 129–136, San Diego, May 1991. Morgan Kaufmann Publishers, Inc.

[2] Charles L. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, (1), 1982.

[3] Carnegie Group. Text categorization shell. Technical brief, Carnegie Group, Five PPG Place, Pittsburgh, PA 15222, 1989.

[4] Carl Weir, Tim Finin, Robin McEntire, and Barry Silk. Unisys: Description of the Unisys system used for MUC-3. In *Proceedings of the Third Message Understanding Conference*, pages 212–222, San Diego, May 1991. Morgan Kaufmann Publishers, Inc.

[5] Carl Weir, Robin McEntire, Barry Silk, and Tim Finin. Unisys: MUC-3 test results and analysis. In *Proceedings of the Third Message Understanding Conference*, pages 112–115, San Diego, May 1991. Morgan Kaufmann Publishers, Inc.