1995113239

N95-19645

An Expert System for Configuring a Network for a Milstar Terminal

34080
p. 9

Melissa J. Mahoney
Dr. Elizabeth J. Wilson

Raytheon Company
1001 Boston Post Road
Marlboro, MA 01572

melissa@tif352.ed.ray.com
bwilson@sud2.ed.ray.com

Abstract

This paper describes a rule-based expert system which assists the user in configuring a network for Air Force terminals using the Milstar satellite system. The network configuration expert system approach uses CLIPS. The complexity of network configuration is discussed, and the methods used to model it are described.

1. Introduction

Milstar COMPLEXITY

Milstar is a flexible, robust satellite communications system which provides world-wide communication capability to a variety of users for a wide range of tactical and strategic applications. Milstar is interoperable requiring simultaneous access by Air Force, Navy and Army terminals. The transmit/receive capability of these terminals ranges from small airborne units with limited transmission power to stationary sites with large fixed antennas.

The wide range of applications of the Milstar system includes services capable of supporting status reporting, critical mission commands, point-to-point calls, and networks. The same system that provides routine weather reports must also allow top-level mission coordination among all services. This diversity results in traffic which is unpredictable and variable. The system must accommodate both voice and data services which can be transferred at different rates simultaneously. In order to protect itself from jamming and interference, the Milstar waveform includes a number of signal processing features, such as frequency hopping, symbol hop redundancy, and interleaving, some of which are among the network parameters defined at service setup.

The configuration of each terminal is accomplished through the loading of adaptation data. This complex data set provides all the terminal specific and mission specific parameters to allow the terminal to access the satellite under the mission conditions with the necessary resources to establish and maintain communication. Network definitions are a subset of this adaptation data.

Defining a network for a terminal is a challenging task requiring a large body of knowledge, much of it heuristic. The systems engineers who have been on the Milstar program for many years have amassed this knowledge; many of them have been reassigned to other programs. Since the recent launch of the first Milstar satellite, testing activity has been very high and expertise in this area is not readily available. When testing ends and Milstar becomes fully operational, the terminals will be operated by personnel whose Milstar knowledge is considerably less than that of the systems engineers. To solve the problem of having a user with limited knowledge take advantage of the many network configuration capabilities of the Milstar system, one of the systems engineers proposed developing an expert system to encapsulate the vast amount of knowledge required to successfully design network definitions accurately and reliably.

NETWORK COMPLEXITY

A Milstar network consists of 82 parameters which are interrelated. Some relationships are well-understood, but many are obscure. The few existing domain experts spend a significant amount of time constructing network definitions starting from a high-level mission description usually provided by a commanding authority. Not only should the network allow communication, but it should do so in a manner which makes optimal use of satellite resources. The expert, then, is faced with an optimization problem: to define a network whose parameters are consistent with each other while minimizing the use of payload resources.

The complexity of network configuration is compounded by the number and locations of the terminals which will participate in the network. Additional terminals add more constraints to the problem space. The resource requirements of different terminals may vary, and their locations may be dynamic, as in the case of airborne terminals.

The complexity further increases when configuring a terminal to participate in several networks simultaneously. Certain resources cannot be shared among a terminal's active networks.

KNOWLEDGE ACQUISITION

Several system engineers were available as knowledge sources. Initially, they were asked to review each of the network parameters and to describe obvious relationships between them. Remarks about rules, exceptions, and values were recorded and coded into prototype rules. These rules were reviewed with the experts and modified as needed to confirm that the information had been accurately translated.

In addition to interviewing the domain experts, scores of documents were read. Tables and charts were constructed to focus on the relationships between only several parameters at a time. Gradually, a hierarchy of knowledge was formed.

The knowledge base was partitioned into functional areas (e.g., hardware parameters, mission-level parameters, terminal-specific parameters, and service-specific parameters). A knowledge engineer was assigned to each one. When it was deemed that most of the knowledge had been acquired in one of these areas, the entire knowledge engineering team of four would check the functional area's rules with respect to each network type (of which there are 11). This cross-checking resulted in more rules which captured the exceptions and special cases associated with the more general rules belonging to the functional area.

In-house operations and maintenance training courses provided the knowledge engineers with an opportunity to learn Milstar as an Air Force terminal operator. This provided the hands-on training that taught lessons not documented in specifications.

2. The Expert System

SCOPE

The scope of the task was to develop an expert system to assist the user in configuring a network definition for a single terminal. This scope represents the necessary first step in later building an extended system which will address the issues of multiple terminals and multiple networks.

SYSTEM COMPONENTS

The expert system consists of three major components: a graphical user interface (GUI), a rule and fact base, and an interface between those two components. The GUI performed the more trivial consistency checks between the input parameters. For example, if a voice network is being configured, then the only valid data rate for the network is 2400 bps. The GUI ghosts out all other choices. The GUI also checks the syntax and ranges of user-entered data.

The interface code controls the flow of the expert system. It reads in the user-supplied input items, translates them into facts, loads the appropriate rules; when the rules have finished firing, the interface code passes their results to the GUI. This process continues as the user moves between the screens.

Rules and facts were used to check the more complicated relationships between the parameters. Rules were not used to perform trivial tasks (e.g., syntax checking) nor inappropriate tasks (e.g., controlling the flow).

SOFTWARE TOOLS

CLIPS was chosen for this project because of its forward-chaining inference engine, and its ability to run as both a standalone and an embedded application. Other benefits were its object-oriented design techniques and its level of customer support.

The rules and facts were developed, tested and refined on a Sun workstation under UNIX. They were ported to an IBM 486/66, the target environment. A point-and-click graphical user interface was developed for the PC using Visual Basic. Borland C++ was used to write the interface code between the GUI and the CLIPS modules.

Currently, the expert system contains about 100 rules and 1000 facts.

OBJECTS

CLIPS Object-Oriented Language (COOL) was used where appropriate.

In the problem domain, terminals are represented as objects having attributes such as a terminal id, an antenna size, a latitude/longitude location, and a port usage table. The port usage table is made up of 34 ports. A port is an object which has attributes such as data rate and i/o hardware type. Terminals are objects to prepare for future expansion to the analysis of a network definition for multiple terminals.

CLIPS offers a query system to match instances and perform actions on a set of instances. For example, CLIPS can quickly find the terminal requiring the most robust downlink modulation by keying off each terminal's antenna size and satellite downlink antenna type. Satisfying the needs of this terminal would be a constraint to ensure reliable communication among a population of terminals.

TEMPLATES

Valid parameter relationships are defined using templates and asserted as facts during system initialization. For example, only a subset of uplink time slots are valid with each combination of data rate, uplink modulation, and uplink mode. Figure 1a shows a table relating these parameters. Figure 1b shows a template modeled after this table. Figure 1c shows a deffacts structure which will assert the facts contained in the table by using the template.

RULES

Several rules exist for most of the 82 parameters. The left-hand side of each rule specifies a certain combination of patterns which contain some bounded values (i.e., network parameters that already have values assigned). The right-hand side of each rule performs three actions when it fires. First, it assigns a suggested parameter value. Second, it creates a multifield containing other valid, but less optimal, parameter values. And, third, it creates an explanation associated with the parameter. The GUI highlights the suggested value, but allows the user to change it to one of the other valid choices. Figure 2 shows a rule which fires when the expert system does not find any valid uplink time slots. [* The problem of refiring needed to be addressed. Refiring occurred because when a rule modifies a fact, the fact is implicitly retracted and then reasserted. Since the fact is "new", the rule fires again immediately. This problem was solved by including a check in the antecedent whether the valid choices multifield is currently the same as what the rule will set it to be. If the fields are identical, then the rule will not refire.]

EXPLANATIONS

The explanation associated with each parameter reflects the left-hand side of the rule which ultimately assigned the parameter's value. An explanation template contains the text, and a flag to indicate whether a positive rule fired (i.e., a valid parameter value was suggested), or a negative rule fired (i.e., the expert system could find no valid value for the parameter based upon preconditions). In the latter case, the GUI immediately notifies the user that the expert system is unable to proceed, and it states a recommendation to fix the problem. Normally the user is instructed to back up to a certain parameter whose current value is imposing the unsatisfiable constraint. The explanation facility is invoked by pointing the mouse on the parameter and clicking the right button.

OUTPUT

After the user has proceeded through the screens and values have been assigned to all network parameters, s/he may choose various output media and formats. In production, the completed network will be converted to hexadecimal format and downloaded into the terminal's database which is then distributed to the field.

FUTURE EXTENSIONS

As stated earlier, a terminal class was established to allow for future expansion to the analysis of a network with respect to multiple terminals. Taking this idea one step further would involve creating a network class, and making a network definition an instance of it. Multiple networks could then be considered simultaneously when writing rules to suggest values for certain parameters. For example, a terminal cannot participate in two networks which use the same uplink time slot. Advancing the expert system to this level of consistency checking would improve its utility significantly. CLIPS promotes a modular approach to design which allows the developer to plan for future extensions like these.

3. Conclusions

An expert system has proven to be a valuable tool in the configuration of Milstar terminal network parameter sets. The generation of these parameter sets is complex and primarily heuristic lending itself to an expert system approach. CLIPS and COOL has provided an effective and efficient development environment to capture the knowledge associated with the application domain and translate these relationships into a modular set of rules and facts. The decision to use rules vs. facts, CLIPS vs. C++, rules vs. GUI was made after careful consideration of the overall implementation strategy and the most efficient means to provide the expertise.

The complexity of the problem domain, variety of sources of expertise (design engineers, operational support, and program documentation), and broad and varying scope of the applications has made this project a significant knowledge acquisition challenge. The knowledge engineering team approach has been successful in translating the vast and subtle nuances of the implementation strategies and in developing a synergetic composite of the available expertise.

The combined use of rules, facts, objects, C++ code, and a graphical interface has provided a rich platform to capture the Milstar knowledge and transform this knowledge into a modular tool staged for expansion and improvement.

References

1. Basel, J., D'Atri, J. and Reed, R., "Air Force Agile Beam Management", Raytheon Company, Equipment Division, Marlborough, Massachusetts, June 15, 1989.

2. Giarratano, J. and Riley, G., EXPERT SYSTEMS: PRINCIPLES AND PROGRAMMING, ed. 2, PWS Publishing Company, Boston, Massachusetts, 1994.

3. Vachula, G., "Milstar System Data Management Concept", GMV:92:05, Raytheon Company, Equipment Division, Communication Systems Laboratory, Marlborough, Massachusetts, February 7, 1992.

Primary U/L Channels

U/L Modulation		Data Rate (bps)					
		2400	1200	600	300	150	75
HHR FSK	2	1,2,3,4	--	--	--	--	--
	4	5,6	1,2,3,4	--	--	--	--
	8	7	5,6	1,2,3,4	--	--	--
	16	--	7	5,6	1,2,3,4	--	--
	32	--	--	7	5,6	1,2,3,4	--
	64	--	--	--	7	5,6	1,2,3,4
	128	--	--	--	--	7	5,6
	256	--	--	--	--	--	7
LHR FSK	10	--	--	--	7	5,6	1,2,3,4
	20	--	--	--	--	7	5,6
	40	--	--	--	--	--	7

Secondary U/L Channels

U/L Modulation		Data Rate (bps)		
		300	150	75
HHR FSK	2	1,2,3,4	--	--
	4	5,6	1,2,3,4	--
	8	7	5,6	1,2,3,4
	16	--	7	5,6
	32	--	--	7

Figure 1a: Table Relating Uplink Time Slot, Uplink Mode, Uplink Modulation and Data Rate

```
; valid-dr-ts

; this template sets up associations of uplink mode,

; uplink modulation, and data rate/timeslot pairs

(deftemplate valid-dr-ts)
        (field ul-mode
                (type SYMBOL)
                (default ?NONE))
        (field ul-modulation
                (type SYMBOL)
                (default ?NONE))
        (multifield dr-ts
                (type ?VARIABLE)
                (default ?NONE))
```

Figure 1b: Template Corresponding to Table in Figure 1a

```
; this deffacts defines valid data rate and timeslot pairs

; for each combination of uplink mode and uplink modulation

(deffacts set-valid-dr-ts
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation HHRFSK2)
                            (dr-ts 2400 bps with time slots 1-4))
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation HHRFSK4)
                            (dr-ts 2400 bps with time slots 5-6
                            1200 bps with time slots 1-4))
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation HHRFSK8)
                            (dr-ts 2400 bps with time slot 7
                            1200 bps with time slots 5-6))
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation HHRFSK16)
                            (dr-ts 1200 bps with time slot 7
                            600 bps with time slots 5-6
                            150 bps with time slots 1-4))
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation HHRFSK32)
                            (dr-ts 600 bps with time slot 7
                            300 bps with time slots 5-6
                            150 bps with time slots 1-4))
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation HHRFSK64)
                            (dr-ts 300 bps with time slot 7
                            150 bps with time slots 5-6
                             75 bps with time slots 1-4))
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation HHRFSK128)
                            (dr-ts 150 bps with time slot 7
                            75 bps with time slots 5-6))
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation HHRFSK256)
                            (dr-ts 75 bps with time slot 7))
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation LHRFSK10)
                            (dr-ts 300 bps with time slot 7
                            150 bps with time slots 5-6
                             75 bps with time slots 1-4))
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation LHRFSK20)
                            (dr-ts 150 bps with time slot 7
                            75 bps with time slots 5-6))
        (valid-dr-ts        (ul-mode PRIMARY)
                            (ul-modulation LHRFSK40)
                            (dr-ts 75 bps with time slot 7))
        (valid-dr-ts        (ul-mode SECONDARY)
                            (ul-modulation HHRFSK2)
                            (dr-ts 300 bps with time slots 1-4))
```

Figure 1c: Deffacts Representation of Table in Figure 1a

```
(valid-dr-ts      (u1-mode SECONDARY)
                  (u1-modulation HHRFSK4)
                  (dr-ts 300 bps with time slots 5-6
(valid-dr-ts      (u1-mode SECONDARY)
                  (u1-modulation HHRFSK8)
                  (dr-ts 300 bps with time slot 7
                  150 bps with time slots 5-6
                   75 bps with time slots 1-4))
(valid-dr-ts      (u1-mode SECONDARY)
                  (u1-modulation HHRFSK16)
                  (dr-ts 150 bps with time slot 7
                   75 bps with time slots 5-6))
(valid-dr-ts      (u1-mode SECONDARY)
                  (u1-modulation HHRFSK32)
                  (dr-ts 75 bps with time slot 7))
```

```
(defrule 12bults-009
"   this rule fires when the terminal's required u/1 modulation is more robust than the implied u/1 modulation
    using any u/1 time slot."
        (outnet  (name data-rate)
                 (value ?dr-value))          ;   get value of data-rate
        (outnet  (name u1-mode)
                 (value ?u1-mode))           ;   get value of u1-mode
        (outnet  (name u1-antenna-type)
                 (value ?u1-ant-type))       ;   get value of u1-ant-type
        (outnet  (name net-type)
                 (value ?net-type))          ;   get value of net-type
        ?fact <- (outnet   (name u1-time-slot)   ;   get address
                           (choices $?oldchoices))  ;   of time slot
        (test    (neq $?oldchoices =(mv-append nil)))   ;   don't refire (* see note)
        (terminal ?term-id)                  ;   get the terminal id
        (u1modmap          (ant-size =(get-ant-size  ?term-id)
                           (u1-ant ?ul-ant-type)
                           (u1-modulation ?ul-mod-terminal))
                                                        ;   get the minimum
                                                        ;   u1 mod this terminal
                                                        ;   can use given its antenna size
                                                        ;   and the u1 antenna type
        (outnet  (name u1-modulation)
                 (value ?u1-mod-type))                  ;   get the u/1 modulation type
        (valid-u1-quad     (data-rate ?dr-value)        ;   get all valid modulations
                           (u1-mode ?u1-mode)           ;   for this data rate
                           (u1-mod-type ?u1-mod-type)
                           (u1-mods $?all-valid-mods))
        (most-to-least u1 ?u1-mode $?allmods)           ;   get all modulations
        (test    (more-robustp ?u1-mod-terminal   (first $?all-valid-mods)    $?allmods))
                                                        ;   if the terminal's
                                                        ;   required robustness is
                                                        ;   more robust than the
                                                        ;   most robust implied
                                                        ;   modulation
        (valid-dr-ts       (u1-mode ?u1-mode)
                                   (u1-modulation ?u1-mod-terminal)
                                   (dr-ts $?all-dr-ts))    ;   get all pairs of valid data rates
                                                        ;   and timeslots for this modulation
        ?explain <- (explanation (param-name ul-time-slot)  ;   get previous explanation
                                 (text ?oldtext))       ;   associated with this
                                                        ;   parameter
```

Figure 2: Rule Using the Facts Shown in Figure 1c

```
=>

(modify ?fact     (suggest nil)                      ;   no suggested value
                  (choices =(mv-append nil)))        ;   no valid choices
(if (neq ?net-type MCE)                              ;   tailor the explanation
then                                                 ;   to the network type
(bind ?newtext
(str-implode
(mv-append
(str-explode "This terminal cannot reliably support this service's data rate u sing the current u/1 antenna
type.  U sing any time slot with this data rate would imply an insufficient u/1 modulation for this
terminal.  Either use a stronger u /1 antenna type, or lower the data rate.  To satisfy this terminal's u/1
modulation requirements, the valid data rate/time slot pairs are:"
$?all-dr-ts
)        ;end append
)        ;end implode
)        ;end bind

else
(bind ?newtext
(str-implode
(mv-append
(str-explode "This terminal cannot reliably support this service's data rate u sing the current u/1 antenna
type.  U sing any time slot with this data rate would imply an insufficient u/1 modulation for this
terminal.  Either use a stronger u /1 antenna type, or lower the data rate.  To satisfy this terminal's u/1
modulation requirements, the valid data rate/time slot pairs are:"
$?all-dr-ts
(str-explode".  Since this is an MCE network, time slot 4 should be used.")
)        ;end append
)        ;end implode
)        ;end bind
)        ;end if

(explain ?explain  ?old text  ?newtext YES)

)
```

Figure 2: Rule Using the Facts Shown in Figure 1c (cont'd.)