Laying a Foundation for Software Engineering of Knowledge Bases in Spacecraft Ground Systems

Mala Mehrotra
Pragati Synergetic Research Inc.
145 Stonelake Ct
Yorktown VA. 23693
mm@norfolk.infi.net

Sergio Alvarado
Aerospace Corporation
P.O. Box 92957
Los Angeles, CA 90009
alvarado@rush.aero.org

Ross Wainwright
Air Force Research Laboratory
3550 Aberdeen Ave SE
Kirtland AFB, NM 87117
ross.wainwright@vs.afrl.af.mil

Abstract

In spacecraft telemetry expert systems technology is being used to manage the complexity generated by the increasing number of complex measurands. However, an uncontrolled proliferation of rules in an expert system can lead to maintenance and management problems of the system. A semi-automated tool, such as Pragati's MVP-CA (Multi-ViewPoint Clustering Analysis) tool, can provide a valuable aid for comprehension, maintenance, verification, validation, integration and evolution of these expert systems by structuring a large knowledge base in various meaningful ways. The MVP-CA tool “mines” the knowledge existent in telemetry rule bases by exploiting the similarity across rules. This knowledge can serve as a handle to verify and validate the knowledge in the existing system as well as to formulate new rule sets for future mission planning activities.

Introduction

The increased number and complexity of spacecraft mission measurands and the evolution of ground systems architectures that support multiple configurable roles have emphasized the need to alleviate the mission operator workload. Rule-based expert systems are a common technology used to manage this complexity; yet a rule set created for a particular mission is often developed in a stand-alone, ad hoc manner. The consequence of this practice is that rule-based systems are redeveloped each time the system changes [Alvarado 1998]. Moreover, due to the critical nature of these applications, much more stringent standards have to be imposed now on their ability to provide reliable decisions in a timely and accurate manner. Pragati's Multi-ViewPoint-Clustering Analysis (MVP-CA) tool provides a framework for clustering large, homogeneous knowledge-based systems from multiple perspectives [Mehrotra & Wild 1995]. It is a semi-automated tool allowing the user to focus attention on different aspects of the problem, thus providing a valuable aid for comprehension, maintenance, verification and validation (V&V), integration and evolution of knowledge-based systems.

The MVP-CA tool has recently been adapted for clustering telemetry knowledge bases. We present here some preliminary results of applying the MVP-CA tool on some telemetry expert systems. In particular, results exposing verification and validation (V&V) problems in the rule bases are presented. We will also briefly discuss our next step of extracting reusable components in a systematic manner by proposing an integration of the MVP-CA tool with case-based reasoning (CBR) technology. Issues relating to indexing, retrieval and adaptation of the rule sets can be addressed effectively when the two technologies are integrated.

Motivation

Expert systems are increasingly being used as intelligent information specialists in cyberspace, both for civilian and military applications. In spacecraft telemetry, expert systems technology is used to manage the complexity generated by the greater number of complex measurands [Lindsay 1998]. Spacecraft satellite telemetry (sub) systems have a unique characteristic in that they usually have multiple configurable roles; hence, there are similar rule bases in existence for different subsystems. As new missions get planned the number of such rule bases with similar structures keeps growing. Also, as new knowledge evolves due to new technology in the market, these systems have to be adapted to incorporate/reflect the changes in technology. Each mission has its own rule set to be applied and each one of them has the potential to grow into a monolithically large unmanageable system. The phenomenon of “add a rule each time” to take care of different situations in any expert system, leads very quickly to an uncontrolled proliferation of rules in the expert system. Due to the data-driven nature of expert systems, as the number of rules of an expert system increase, the number of possible interactions between the rules increases exponentially. The complexity of each pattern in a rule compounds the problem of management of rules even further. Documentation has the danger of becoming obsolete very quickly, as software developers do not always have the necessary discipline to keep updating their documentation. Furthermore, defining any requirements or specifications up front in such a rapid prototyping and iterative development environment, even though they are desirable, becomes an impractical and moot question. Even if they were specified, as any software, conventional or knowledge-based becomes more complex, common errors are bound to occur through misunderstandings of specifications and requirements [Bellman & Walter 1988]. It is therefore desirable to have an analysis tool that exposes a developer to the current software architecture and semantics of the knowledge base in such a dynamically changing development environment, so that the knowledge base can be comprehended at various levels of detail. To achieve this goal, the knowledge in the system has to be suitably abstracted, structured, and otherwise clustered in a manner that facilitates software engineering activities [Jacob & Froscher 1990, Landauer 1990]. Hence**, by exposing the knowledge contained in the knowledge-based** system through the Multi-ViewPoint Clustering Analysis tool, we formulate a basis for addressing reusability, maintainability, and reliability issues for such systems.

Multi-ViewPoint Cluster Analysis (MVP-CA)

Existing approaches to structuring systems are limited in a major way. They only provide a single viewpoint of a system. We believe that **no one single structuring viewpoint is sufficient to comprehend a complex system**. In this paper we show the feasibility of applying Pragati's Multi-ViewPoint-Clustering Analysis (MVP-CA) methodology on satellite telemetry rule-based systems for V&V and reusability. MVP-CA framework also has the potential to be extended to incorporate case-based retrieval and adaptation technology for reusability of clusters generated through the MVP-CA tool.

Analysis of Existing Rule Sets

We have analyzed the following three telemetry rule bases:

- Spacecraft Environmental Anomalies (SEA-ES)
- X-Ray Timing Explorer (XTE)
- Unexpected Events System Rule Base (UES).

A brief discussion of the results follows.

Spacecraft Environmental Anomalies (SEA-ES)

Spacecraft Environmental Anomalies (SEA-ES) is an expert system developed by The Aerospace Corporation, Space and Environment Technology Center for use in the diagnosis of satellite anomalies caused by the space environment. The satellite anomalies to be detected by the rule base ranges from surface charging, bulk charging, single-event effects, total radiation dose, and space-plasma effects. Various parameters play a role in the determination of these anomalies such as, orbit of the satellite, the local plasma and radiation environment, satellite-exposure time, hardness of the circuits and their components etc.

```
5   ACCUM_FLUEN=V_H => CS=BLK_CRG CF60

201 RCRNC AND PRDCTY=OF_H_PENETRATING_FLUX AND
        ACCUM_FLUEN=H OR ACCUM_FLUEN=V_H => CS=BLK_CRG
        CF60
 .......................................................................

6   ACCUM_FLUEN=H => CS=BLK_CRG CF20

202 ACCUM_FLUEN=H OR ACCUM_FLUEN=V_H AND SAME_ORBIT  =>
        CS=BLK_CRG CF20
 .......................................................................
22 MAG_STATE_RECENT=DISTURBED => CS=BLK_CRG CF10

40 RCRNC AND PRDCTY=MAGNETICALLY_DISTURBED AND
        MAG_STATE_RECENT=DISTURBED => CS=BLK_CRG CF10
```

Figure 1: Rule pairs showing redundancies

```
19  RCRNC AND LT_RECUR => CS!=BLK_CRG CF20

110 RCRNC AND LT_RECUR => CS!=TOTAL_DOSE
.......................................................................

47   JUL_DATE IS KNOWN => PR_STMT7 AND DBASE-REPORT
         QUAL JDATE  > JSTART AND JDATE < JSTOP AND
         DATE_REPORT

51    JUL_DATE IS KNOWN => JSTART=JUL_DATE-2 AND
         JSTOP=JUL_DATE+2
163   JUL_DATE IS KNOWN => JSTART_FLARE=JUL_DATE-5 AND
      JSTOP_FLARE=JUL_DATE +1

.......................................................................
132 ORBIT=GEOSYNC => ALT=GEOSYNCHRONOUS AND
         INCLIN=EQTRL

144 ORBIT=GEOSYNC =>D_RT_10=5000000 AND
         D_RT_20=1700000 AND D_RT_50=220000 AND
         D_RT_100=30000 AND D_RT_200=1333
```

Figure 2: SEAES rules with inconsistencies

```
Pattern(s) : INCL became stable in group:
Rule#      Rule Description
  138  INCL BT -5 5 => INCLIN = EQTRL
  139  INCL BT 5 30 V INCL BT -5 -30 => INCLIN = L_INCLIN
  140  INCL BT 30 60 V INCL BT -30 -60 => INCLIN = I_INCLIN
  141  INCL BT 60 80 V INCL BT -60 -80 => INCLIN = H_INCLIN
  142  INCL BT 80 90 V INCL BT -80 -90 => INCLIN = PLR
  173   PERIGEE BT 96 145 ^ APOGEE BT 320 480 ^ INCL BT 80 100 =>
            ORBIT = LOW1
  175   PERIGEE BT 200 300 ^ APOGEE BT 200 300 ^ INCL BT 45 70 =>
            ORBIT = STS57L
  177   PERIGEE BT 280 420 ^ APOGEE BT 280 420 ^ INCL BT 45 70 =>
            ORBIT = STS57H
  179   PERIGEE BT 480 720 ^ APOGEE BT 480 720 ^ INCL BT 45 70 =>
            ORBIT = ERBS
  174   PERIGEE BT 135 205 ^ APOGEE BT 185 276 ^ INCL BT 85 105 =>
            ORBIT = LOW2
  180   PERIGEE BT 630 770 ^ APOGEE BT 630 770 ^ INCL BT 90 105 =>
            ORBIT = L^SAT
  181   PERIGEE BT 735 900 ^ APOGEE BT 750 920 ^ INCL BT 90 110 =>
            ORBIT = DMSP
  183  PERIGEE BT 800 980 ^ APOGEE BT 820 1000 ^ INCL BT 90 110 =>
            ORBIT = IRAS
  176   PERIGEE BT 240 360 ^ APOGEE BT 240 360 ^ INCL BT 20 35 =>
            ORBIT = STS28L
 178    PERIGEE BT 400 600  ^ APOGEE BT 400 600  ^ INCL BT 20 35 =>
        ORBIT = STS28H

                          . . .
```

Figure 3: Candidate reusable SEAES rule cluster

At a logical level, clustering this rule base with the MVP-CA tool exposed a few inconsistencies as well as a few redundancies. An inconsistent condition exists if two rules have the same antecedent but different conclusions. In a forward chaining system, this creates an anomalous condition. A redundant condition is detected when the consequents of two rules are the same, but one of the antecedents has conditional clauses that are a subset of the other. In other words, a general and a more specific rule are asserted in the same rule base. This gives rise to a dependency on the conflict-resolution scheme of the inference engine, on its criteria to resolve such a conflict at run-time. This makes the system vulnerable to unpredictable behavior when transitioning to other expert system shells/platforms.

Figure 1 and Figure 2 shows rule pairs exhibiting redundancies and inconsistencies as flagged by the MVP-CA tool. In this, and later figures, the "=>" symbol separates the antecedent from the consequent.

A representative stable group for the concept of inclination (INCL) is presented in Figure 3. It clearly shows the relationship between different orbit types, inclination types, perigee and apogee. Concepts such as *inclination* are a supporting domain concept in this rule base, that the MVP-CA tool allows us to identify through the clustering of rules. These rule sets can become viable candidates for potential reuse.

X-Ray Timing Explorer (XTE)

X-Ray Timing Explorer (XTE) is an expert system written in GenSAA (Generic Spacecraft Analyst Assistant, which is a superset of Clips. GenSAA was built by NASA to serve as a development and application environment for building expert systems at various NASA control centers. XTE is a health and safety monitoring rule base, checking the various onboard subsystems on the satellite, such as attitude and control system, power subsystem, thermal subsystem, solar array subsystem, spacecraft data subsystem, transponder subsystem, and many others.

As was iterated earlier, a rule base needs to be first cleansed before any reusable component identification can be done. In XTE we discovered through the clustering, an interesting rule-name duplication error. The process of discovery was partly automatic and partly manual. The two sets of clustering shown in Figure 4, on the *status check aspect* and on the *limit check aspects*, in the various subsystems were obtained through the MVP-CA clustering.

The two rules, Rule # 33 and Rule # 35, dealing with, TAM, the three-axis-magnetometer, appeared dubious as they had the same rule names. However, when we inspected the rule contents for Rule # 35 as shown in Figure 5, we found that all allusions inside it were for PCA, proportional counter array! The allusions to PCA in the rule's content have been highlighted for bringing the readers attention to the inadvertent rule-naming mistake. The process of discovering this anomaly was eased by the fact that we had a very similar rule set performing the limit check on the same subsystems. A manual inter-cluster comparison made the error quite obvious to us. It was an important revelation from two standpoints. Firstly, in Clips if two rules have the same rule name one simply masks the existence of the other. Secondly, since these rules were to be a part of the operational system in NASA's Mission Operations Control Room, it was very

Rule#	Rule Description
23	sa_status_check
27	xpndr_status_check
29	gsace_status_check
33	***tam_status_check***
35	***tam_status_check***
25	sds_status_check
31	rwa_status_check

Rule#	Rule Description
24	sa_limit_check
28	xpndr_limit_check
30	gsace_limit_check
34	tam_limit_check
36	pca_limit_check
26	sds_limit_check
32	rwa_limit_check

Figure 4: Rule name error in Rule #35 of XTE

```
Rule # 35
(defrule tam_status_check ""
        (LimitStatus PAPCU1TMP2T#XTE_DECOM  ?x1)
        (LimitStatus PAPCU2TMP2T#XTE_DECOM  ?x2)
        (LimitStatus PAPCU3TMP2T#XTE_DECOM  ?x3)
        (LimitStatus PAPCU4TMP2T#XTE_DECOM  ?x4)
        (LimitStatus PAPCU5TMP2T#XTE_DECOM  ?x5)
        ?o1 <- (Inferred PCA-Temp-Status ?cur_stat)
        (Inferred valid-telemetry  valid)
=>
        (bind ?stat (worst-status-test ?x1 ?x2 ?x3 ?x4 ?x5 0))
        (if (= ?stat 4) then (bind ?new_stat HighAlarm)
           else (if (= ?stat 3) then (bind ?new_stat LowAlarm)
                  else (if (= ?stat 2) then (bind ?new_stat HighWarning)
               else (if (= ?stat 1) then (bind ?new_stat LowWarning)
                      else (bind ?new_stat InLimits)))))
        (if (neq ?cur_stat ?new_stat) then
            (retract ?o1)
            (AssertFact  "Inferred PCA-Temp-Status" ?new_stat)
            (if (= (str-compare ?new_stat "Normal") 0)
                  then (SendMessage "MessageWindow" Status (str-cat "PCA
                         Temperatures changed from " ?cur_stat " to " ?new_stat))
               else (SendMessage "MessageWindow" Warning (str-cat "PCA
                              Temperatures changed from " ?cur_stat " to "
?new_stat)))
             (if (str-index "Alarm" ?new_stat)
```

Figure 5: XTE Rule with Naming Anomaly

important that such inadvertent human errors be exposed at the earliest possible time and certainly before the rule base became operational, because it can result in unexpected runtime behavior.

Unexpected Events System (UES)

Unexpected Events System (UES), the third expert system that we examined, was built as part of NASA's FUSE (Far Ultraviolet Spectroscopic Explorer) project, by Interface and Control Systems, located in Columbia, MD. UES was built using a commercial product, SCL (Spacecraft and Command Language), which has an autonomous system architecture that encompasses both flight and ground operations. It enables on-board and/or ground control operations by providing visibility for the state of each on-board sensor or system. The use of scripts and rules allows for various operational logic to be defined on the various sensors and subsystems. In addition, it has formatting directives to generate real-time commands. The UES system provided to us performed bit checks on a data stream through digital and analog mnemonics. The relational data files (RDL) provided some qualitative information on the mnemonics by way of specifying their range and types. Another set of derived mnemonics – known as pseudo-mnemonics existed in modal files, which defined the derived values of some of the mnemonics.

In applying the MVP-CA tool for this rule base we had to first adapt our tool to process rules which were making script calls. Scripts are the procedural part of the SCL environment and are a much-needed feature for performing any synchronous operations. The asynchronous operations are taken care of by rules. Our tool had to be adapted to store information about the subsystem and category in which the rules were active. MVP-CA tool was adapted to store the priority and activation information as well. Having adapted the tool to take as input these special types of rules, we found some conflicting rules, potentially redundant rules and some cases for reusable rule sets.

In the UES system, we found two rule pairs with the same name, performing exactly the opposite function as shown in Figure 6. The modal rule, for the detector 1 auxiliary power subsystem performs a check on mnemonic *i_det1auxpwrst* to be zero and issues the message that the telemetry *i_det1auxpwrst* is in mode 1. However, in the second rule, the same mnemonic, for the same value, sends a message that it is in mode 2. Moreover, since these rules have been developed semi-automatically through a standard user interface, it was easy to repeat the same mistake with detector 2, as was evidenced by another set of similar rules dealing with detector 2. We leave it to the domain expert to disambiguate these types of potentially dangerous and conflicting actions.

```
rule modal_im_det1auxpwrst -- rule 15
subsystem modal_i_det_t27hsk category rtmm
if change (i_det1auxpwrst) then
        if i_det1auxpwrst = 0 then
                msg "modal tlm i_det1auxpwrst now in mode
1"
        else if i_det1auxpwrst = 1 then
                  msg "modal tlm i_det1auxpwrst now in mode
2"
                          …
end modal_im_det1auxpwrst


rule modal_im_det1auxpwrst -- rule 16
subsystem  modal_i_det_t27hsk      category rtmm
if change (i_det1auxpwrst) then
        if i_det1auxpwrst = 1 then
                     msg  "modal tlm i_det1auxpwrst  now  in
mode 1"
        else if i_det1auxpwrst = 0 then
                     msg  "modal tlm i_det1auxpwrst  now  in
mode 2"
                              …
```

Figure 6: Anomalies in the UES rule base

Reusability of Rulesets

In the MVP-CA-based environment, it is envisioned that legacy expert systems can be clustered into rule sets of semantically related rules. Once we have a mechanism for decomposing the expert systems in various meaningful ways, relevant rule sets from different expert systems can be retrieved and assimilated through case-based retrieval (CBR) and analogical reasoning techniques [Wolverton & Roth 1994]. In fact, the sets of rules could be "wrapped" in such a manner that commercial CBR tools could be used to retrieve the relevant rule sets as and when required. Once the appropriate rule set has been retrieved through the Cluster Interface Definition (CID), they can be adapted for the new mission's functionality as needed. For the new evolving prototypes, providing insight into the continually changing models through the MVP-CA tool can prove to be a valuable aid in their transition to an operational stage. Such an environment could then support the orderly and reliable transition of evolving, complex, knowledge-based system software in the satellite telemetry domain, so that such systems can be reused for new scenarios.

This environment will focus on the issues of long-term maintenance, reusability and evolution of mission-specific rule sets in spacecraft telemetry systems. Preliminary investigation is currently under way to study how case-based retrieval and storage techniques could be used effectively for storing and retrieving such CIDs defining the rule clusters. Our research efforts address the possibility of providing a software environment which enables semi-automatic detection, storage, retrieval and adaptation of these rule sets so that reusability of existing rule sets can be addressed across missions in a systematic and disciplined manner [Turner & Mangan]. It is envisioned that some form of case-based storage and retrieval techniques will be incorporated into the MVP-CA methodology for reuse of rule sets, so that a full-scale prototype environment can be built. Such an environment will alleviate the developer from the tedious burden of manually inspecting large and complex legacy rule bases before building a new rule base for the next similar mission.

Even though our ideas are being applied to telemetry applications primarily, the methodology for reusability being advocated here can be transitioned to other knowledge-based applications areas such as, medical, forensics, civil engineering and others. Also the clustering methodology in the MVP-CA technology is not dependent on any particular knowledge representation scheme or the language of the knowledge-based system; hence, the MVP-CA methodology can be integrated into any environment that encapsulates domain knowledge in a regular form.

Conclusions

A key to any successful expert system development effort is effective knowledge engineering. Typically there is not a standard way to perform expert system knowledge engineering because it is usually dependent on available resources such as the availability of domain experts and the adequacy of documentation. The expert system developer could certainly make use of previous work on similar expert systems. Without a supporting tool, expert system reuse is at best opportunistic.

We have shown that the MVP-CA prototype tool is able to extract various views of expert systems through the clustering of rules. The rule clusters form a basis for understanding the system for verification and validation purposes because they are suggestive of various rule-models inherent in the software system. Clusters are also suggestive of reuse. Given the successful development of the MVP-CA tool, the expert system developer will be in the position to leverage the knowledge of existent expert systems for building new ones in a reliable and efficient manner.

Acknowledgements.

This research was supported by a Phase I SBIR Air Force contract F29601-98-C-0102. Our thanks to Walter Truszkowski, NASA Goddard Space Flight Center, and Allan Posner, Interface and Control Systems for providing us their knowledge bases and technical assistance.

References

S.J.Alvarado. An Evaluation of Object-Oriented Architecture Models for Satellite Ground Systems. *In Proceedings of the Ground System Architecture Workshop, GSAW98*. The Aerospace Corporation. El Segundo, CA. http://sunset.usc.edu/GSAW/1998

K. L. Bellman and D. O. Walter. Analyzing and correcting knowledge-based systems requires explicit models. *In AAAI-88 Workshop on Verification, Validation and Testing of Knowledge-Based Systems, July 1988.*

R. J. K. Jacob and J. N. Froscher. A software engineering methodology for rule-based systems. *IEEE Transactions on Knowledge and Data Engineering,* 2(2): 173-189, June 1990.

C. Landauer. Correctness principles for rule-based expert systems. *Journal of Expert Systems with Applications,* 1:291-316, 1990.

S. Lindsay. COTS AI Technologies for Satellite Control. *Proceedings of the Ground System Architecture Workshop, GSAW98*. The Aerospace Corporation. El Segundo, CA. http://sunset.usc.edu/GSAW/1998

M. Mehrotra and C. Wild. Analyzing knowledge-based systems using multi-viewpoint clustering analysis. *Journal of Systems and Software,* 29:235-249, 1995.

S.R. Turner and P. Mangan. Reusable Expert Systems. In *Proceedings of the Fourth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems*, 780-784. June 1991.

M. Wolverton and B. Hayes-Roth. Retrieving semantically distant analogies with knowledge-directed spreading activation. *In Proceedings, National Conference on Artificial Intelligence*, pages 56-61, 1994.