Check for updates

PVDaCS: A Prototype Knowledge-Based Expert System for Certification of Spacecraft Data

Cathleen Wharton
Department of Computer Science and Institute of Cognitive Science
Campus Box 430
University of Colorado at Boulder
Boulder CO 80309-0430

Patricia J. Shiroma
Karen E. Simmons
Laboratory for Atmospheric and Space Physics
Campus Box 392
University of Colorado at Boulder
Boulder CO 80309-0392

Abstract

On line data management techniques to certify spacecraft information are mandated by increasing telemetry rates [SiC87]. Knowledge-based expert systems offer the ability to certify data electronically without the need for time consuming human interaction. We have explored issues of automatic certification by designing a knowledge-based expert system to certify data from a scientific instrument, the Orbiter Ultraviolet Spectrometer (OUVS), on an operating NASA planetary spacecraft, Pioneer Venus (PV) [Ste80]. The resulting rule-based system, called PVDaCS (Pioneer Venus Data Certification System), is a functional prototype demonstrating the concepts of a larger system design. A key element of the system design is the representation of an expert's knowledge through the usage of well ordered sequences. PVDaCS produces a certification value derived from expert knowledge and an analysis of the instrument's operation. Results of system performance are presented.

Introduction

The PVDaCS prototype currently investigates a part of a larger data management plan. In particular, the system explores data certification techniques. *Science data certification* indicates the verification of the integrity of the data before its use in science analysis. Several factors have been identified as certification criteria: verification of instrument functionality, verification of scientific data integrity, and the quality, quantity, and continuity of the telemetered information. Certification of PV OUVS science data involves the comparison of information from several sources to confirm that the instrument is operating according to the Principal Investigator's requests. This information includes: requested instrument commands, reported instrument commands, instrument engineering status data, and spectral science data. The instrument operates at about 1000 bits per second for roughly eight hours out of each twenty four hour orbit around Venus. The data are telemetered to the ground after each orbit and a command comparison report is produced.

In order to certify data, the human expert utilizes information from a wide variety of sources. Requested commands, reported commands from the telemetry file, a computer generated image of the science data, and a plot of photon counts recorded over time are evaluated by the expert to determine whether the different sets of information agree. Visual evaluation of images generated from the science data proves to be a great advantage in manual command verification. The human expert inspects the image of the science data to determine the wavelength the instrument observed. Once wavelength has been determined, the expert can infer those commands that were in effect. Images the human expert uses in the analysis of science data are divided into periods of orbit time. Orbit time is measured relative to periapsis. Periapsis occurs when the spacecraft is closest to the planet.

The human expert designs the set of commands to achieve a specific science goal for the orbit. Inherent to the design is a relation between the groups of commands, time with respect to periapsis, and the expected instrument response. The PVDaCS *knowledge-based expert system*[4] accomplishes certification by mapping this expert knowledge into sequences of commands relative to orbital time.

Knowledge Representation

The PVDaCS knowledge base explicitly represents the knowledge inherent to command sequences by borrowing upon the technique employed by chess grandmasters called data chunking. The idea of data chunking is to form a judgement from disjoint information by grouping that information into a logical whole. This technique is commonly developed by chess grandmasters who recognize meaning in the perceived structure associated with chessboard configurations [NeS72, ChS73]. "Black's castled position: King, Rook, Bishop, Knight, and three pawns", for instance, is a well known chess configuration defined by the specific board locations of seven playing pieces [NeS72]. In PVDaCS the meaningful information chunks are the orderly sequences of instrument commands.

PVDaCS uses a procedural knowledge base to represent this meaning inherent to the sequences of OUVS commands. A set of contiguous instrument commands that serve a common purpose should be grouped together. Sequences are based on this concept. The Startup Sequence, used to restart the OUVS instrument in each orbit, is an instantiation of this notion. Each command in this sequence may be used in many other types of sequences to achieve other instrument configurations. However, when used in the specific order defined by the Startup Sequence the commands reinitialize the configuration. PVDaCS uses this idea of command sequences to determine if reported data are either missing or incoherent.

Sequences are also hierarchical in structure. For instance, a complete orbital sequence may be comprised of a Startup Sequence, an Inbound Image Sequence, a Periapsis Sequence, and an Outbound Image Sequence. An Inbound Image Sequence may in turn contain an instrument Calibration Sequence and a 3-Color Sequence.

[4]The term *knowledge-based expert system* indicates that the system achieves expert-level performance through an explicit knowledge representation of such expertise. This contrasts with the terms *expert system* and *knowledge-based system* which respectively are used to indicate that only expert-level performance is achieved or that knowledge is explicitly represented without regard to expertise [BMS86].

```
RULE STARTUP_SEQUENCE
{
 &ORDER   (RELATION  RELATIONSHIP=|COMMAND_ORDER|);
 &POFF    (REQ_REP_CMD_PAIR  CMD=|0026|;
           INDEX = &ORDER.ARG1);
 &PON     (REQ_REP_CMD_PAIR  CMD=|0025|;
           INDEX = &ORDER.ARG2);
 &ACMD1   (REQ_REP_CMD_PAIR  INDEX = &ORDER.ARG3);
 &BCMD1   (REQ_REP_CMD_PAIR  INDEX = &ORDER.ARG4);
 &BUFFC   (REQ_REP_CMD_PAIR  INDEX = &ORDER.ARG5);
 &ACMD2   (REQ_REP_CMD_PAIR  CMD=&ACMD1.CMD;
           INDEX = &ORDER.ARG6);
 &ACMD3   (REQ_REP_CMD_PAIR  CMD=&ACMD1.CMD;
           INDEX = &ORDER.ARG7);
 &PERIOD  (PERIOD  TYPE = |ANY_PERIOD|);
-->
 -- report recognized startup sequence
 -- calculate sequence probability value with respect to time tolerances
};
```

Figure 1: Startup Sequence in Procedural Form.

Figure 1 is an example of the PVDaCS representation scheme. It indicates the seven commands that define a Startup Sequence. This sequence of commands must be executed within a two minute period of time. The first two commands, a Power-Off command and a Power-On command (&POFF and &PON) reset the instrument. Following these commands, the instrument is issued an A-command (&ACMD1) to set the wavelength position and a B-command (&BCMD1) to establish the data format used by the OUVS. Next, a Buffer-Clear command (&BUFFC) is issued to ensure that any data from a previous orbit are removed. Finally, two more A-commands (&ACMD2 and &ACMD3) are issued to ensure that the previous A-command is in effect. (Note that these last two A-commands must be identical to the first A-command). Taken together, these seven contiguous commands form a chunk comprising the PV OUVS instrument Startup Sequence.

Certification Process

PVDaCS certifies data through a three stage process. Each stage of processing corresponds to one of the knowledge types: commands, instrument status, or science data. The first stage compares the logs of requested and reported commands to determine if the requested commands were issued to the instrument. Pairs of requested and reported commands are generated to make this determination. Each pair must fall into a specified time boundary. Accumulation of command pairs fosters the instantiation of sequences. Commands that make up a sequence may be further constrained to occur within a specific orbital period. The command knowledge base defines five orbital periods: Inbound Background Inbound Image, Periapsis, Outbound Image, and Outbound Background. The second stage verifies that the result of the commands executed agrees with the engineering status information. This information expressly reveals knowledge of the instrument's configuration. In the third stage, additional information about the configuration is inferred from the science data. This is accomplished by looking for discontinuities in photon count rates. The times associated with sharp count rate changes should correspond to the execution of commands. Telemetry quality, quantity and continuity values are used at each stage to evaluate the integrity of all information sets.

As a detailed example of stage one processing, consider the following. Figure 1 illustrates the standard Startup Sequence. Initially, requested and reported command pairs (REQ_REP_CMD_PAIR) are found. Processing continues until a set of command pairs is recognized as a known sequence (&POFF, &PON, etc.). Sequence identification requires a specific ordering of these command pairs (e.g. &ORDER.ARG1 must precede &ORDER.ARG2) and their occurrence within a specific time period (&PERIOD). Recognition of the complete sequence causes the rule (STARTUP_SEQUENCE) to fire. This results in the calculation of a sequence probability value with respect to time tolerances and a report of its recognition status.

To coordinate results and to allow for concurrent evaluation between each stage of processing a set of three running instrument configuration elements is used. The configuration elements are updated by firing production rules and are used to resolve data set conflicts. Agreement between data sets establishes certification while disagreement indicates the need to determine if data are misreported or incorrect. The final certification value for an orbit is derived with respect to individual sequence probabilities and agreement between information sources. This certification value represents the integrity of the science data.

Architecture

The sequencing techniques described above have many architectural implications for the system. The inference engine, for example, can be separated from the knowledge base, a distinct advantage recommended by Davis [Dav82]. This separation allows the architectural components to be highly modular and to maintain a loose coupling. The PVDaCS system is shown in Figure 2. Incomplete modules are within the shaded area.

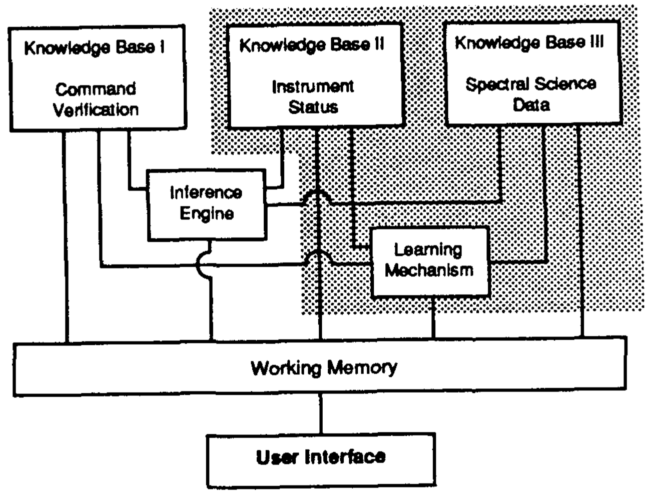

Figure 2: Pioneer Venus Data Certification System

The complete PVDaCS architecture contains the basic components of an ideal knowledge-based expert system [HWL83]. There is an inference engine, a learning component, three distinct knowledge bases, a user interface, and a working memory. The working memory is used for the maintenance of the three running instrument configuration elements, intermediate calculations, and unprocessed portions of input and output data. The three knowledge bases handle the different types of data: command verification, instrument status, and spectral science. Each knowledge base contains both procedural and declarative knowledge. The knowledge base concerned with command verification is quite small, consisting of 51 rules from the following categories: 14 command I/O rules, 20 sequence rules, 14 command comparison rules, and 3 general knowledge base rules. PVDaCS also contains 25 types of facts and 7 user-defined OPS83 functions for numerical calculations, string parsing, and the display of error messages.

PVDaCS' working memory serves as a blackboard through which all system components communicate. The inference engine determines which production will fire and applies it, thus changing the contents of working memory. The scheduling and interpretation process of the productions is accomplished via the OPS83 system shell and interpreter [For86a, For86b]. Modifications have been made to the conflict resolution scheme, changing the default MEA (means-ends analysis) algorithm to a primitive version of the LEX (lexicographic ordering) strategy [BFK85]. The change was made so as to de-emphasize the recency of working memory elements, while placing more emphasis on specificity. Input and output is accomplished through the application of production rules and the certification values are updated as needed. Sequence hierarchies are processed by using backward chained rules. The learning component examines working memory and from its content determines the necessary changes to any of the knowledge bases.

System modularity is further supported by the fact that PVDaCS is written primarily in the production system language OPS83[5]. Subsidiary routines written in the C language and IDL[6] are used for efficient I/O and to facilitate the use of existing routines concerned with the interpretation of scientific data.

OPS83 was selected due to its ability to interface to other languages and because it is a compiled language. Other advantages of using a production system language, such as OPS83, over more conventional programming languages like FORTRAN 77 have been recognized [NLK87]. Instead of the (basically) deterministic flow of control prescribed by conventional languages, the evaluation process can be both goal and data driven. Thus the process is not constrained solely by the program's structure. Since OPS83 can handle variability in rule granularity and size, a wide variety of PV orbits can be easily accommodated. OPS83 rules are allowed to fire independently and knowledge is represented homogeneously. As a result, rules can be added, removed, or augmented as needed. Consequently, only those rules relevant to a particular orbit are used in the certification process.

System Performance and Testing

Test criteria were established for PVDaCS to quantitatively measure system performance. Successive versions of the system will be compared to these benchmarks. Detailed performance evaluation is in progress. Preliminary results are presented. These measurements can be classified according to two categories: overall system performance and a comparison of PVDaCS to the existing Pioneer Venus method of command verification. A baseline orbit was established for the system performance tests. Systematic additions were made to control the dependent parameters. A comparison test was made using timed tests between PVDaCS and the current command verification software, PVCOMPARE.

Running on a VAX 11/785, PVDaCS test measurements are based on the number of rules fired per CPU second. It was expected that there should be a relationship between the number of commands in a given orbit and the amount of time it takes to evaluate that orbit. Preliminary tests show that a greater number of command sequences in an orbit causes more rules to fire during evaluation. Consequently, total processing time is longer for more extensive data sets. Conflict resolution also influences the amount of processing time.

The baseline orbit is the simplest and most common type of PV orbit to be evaluated by PVDaCS. The primary scientific objective for this orbit type is to observe the planet at one wavelength. The baseline orbit consists of 16 commands subdivided into 3 orbital periods. The Inbound Image sequence consists of a startup sequence of 7 commands. The Periapsis Sequence contains 2 commands and the Outbound Image Sequence consists of 7 commands. Synthesized orbits were constructed by using additional sets of 3-Color Sequences. (A 3-Color Sequence is comprised of a cycle of 3 A-commands to examine 3 different wavelengths. One A-command is issued every few minutes to alternate between the 3 wavelengths. After the third A-command has been executed, a new cycle begins.) Extensive PV command sets arise primarily from an increase in the number of sequences during the Inbound Image. Since the 3-Color Sequence is commonly used during in this period, its use in synthesized orbits is reasonable. Total execution time was calculated for each synthesized orbit and results were compared to the baseline orbit. The baseline orbit fired 68 rules and took 2.67 CPU seconds. In contrast, the synthesized orbit with the maximum number of commands, 98, caused 402 rules to fire requiring 8.21 CPU seconds. Fourteen synthesized orbits and ten actual orbits were examined. Each orbit was tested five times to derive an average CPU time.

A successful verification of an orbit data set occurs when PVDaCS flags all missing commands and correctly recognizes all command sequences. The system can successfully verify an orbit only if all of its sequences are part of the knowledge base. Since the current command knowledge base does not contain all known sequences, command sequences for some orbits still fail to be identified. Future refinements to PVDaCS, in particular the addition of the learning mechanism, will further enhance the knowledge base by increasing the number of known sequence rules.

---

[5]OPS83 is a trademark of Production Systems Technologies, Inc.

[6]IDL is an Interactive Data Language under copyright of Research Systems, Inc.

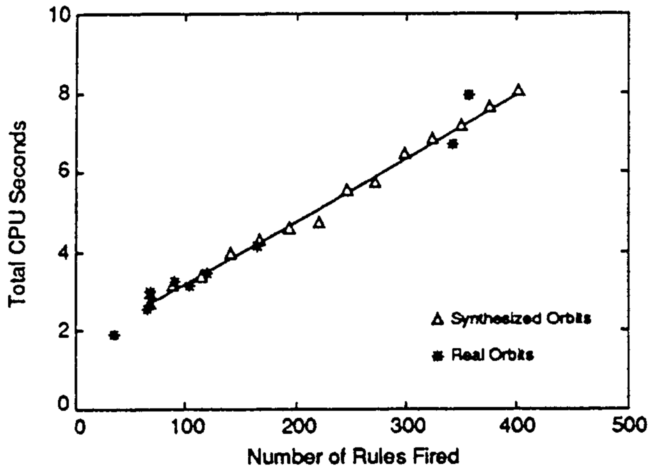

Figure 3: System Test Results

Figure 3 indicates the relationship between the number of rules in the orbit and the total CPU time to evaluate that orbit. Synthesized orbits (triangles) and actual orbits (asterisks) are shown. The line is a linear least squares fit to the synthesized orbits. It appears that the actual orbit set has a similar linear function.

The second category of system tests compared PVDaCS to the existing command verification system, PVCOMPARE. PVCOMPARE looks for differences between individual requested and reported commands, without attempting to explain those differences or to establish a confidence level for orbit. Human interpretation of the PVCOMPARE output file is then required. To verify the same baseline orbit, PVCOMPARE used 14.92 seconds of CPU time. Human interpretation of the output file required about 5 minutes.

Future Plans

Performance results indicate the PVDaCS prototype successfully certifies data with respect to command knowledge. Based on this success, the decision has been made to implement the remaining system modules. Future plans include the development of: the remaining rule bases, operational specifications such as session save-restore, and implementation of the learning mechanism. PVDaCS will employ an experienced-based learning technique. The addition of learning to PVDaCS will improve the certification process by enhancing both procedural and declarative knowledge.

Conclusions

One objective of this prototype was to determine if a system could be built to replace the human expert in the time consuming task of science data certification. The prototype demonstrates that the requested and reported command sets can be certified faster and with more detail than can by done by the human expert or the existing PVCOMPARE software.

Another goal of this research was to investigate whether or not an instrument independent knowledge based expert system could be built by employing domain independent concepts. Examples of domain independent concepts within PVDaCS include hierarchical sequences and running configuration elements. An instrument independent knowledge-based expert system should allow for substitution of knowledge bases across related instruments, provided that other architectural components remain unchanged. For example, the Galileo Ultraviolet Spectrometer, Voyager Photopolarimeter, and PV OUVS all use hierarchical sequences for commanding. If in fact a complete knowledge base of sequences can be built for each instrument, then knowledge bases can be interchanged to create an instrument specific knowledge-based expert system.

Acknowledgements

This work is supported by the PV OUVS project (NASA Contract NAG2-504) and by the Galileo Ultraviolet Spectrometer instrument (NASA Contract JPL 955235). The authors would like to thank Ian Stewart, Kim Chantala, Clayton Lewis, and Gary Bradshaw for their invaluable comments.

References

[BFK85] L. Brownston, R. Farrell, E. Kant, and N. Martin, *Programming Expert Systems in OPS5: An Introduction to Rule Based Programming*, Addison-Wesley Publishing Company, Inc., Reading MA., 1985.

[BMS86] D. G. Bobrow, S. Mittal, and M.J. Stefik, "Expert Systems: Perils and Promise", *Communications of the ACM* **29**, 9 (September 1986), 880-894.

[ChS73] W. G. Chase and H. A. Simon, "Perception in Chess", *Cognitive Psychology* 4 (1973), 55-81.

[Dav82] R. Davis, "Expert Systems: Where Are We? And Where Do We Go From Here?", *The AI Magazine*, Spring 1982, 3-22.

[For86a] C. L. Forgy, *The OPS83 User's Manual System Version 2.2*, Production Systems Technologies, Inc., Pittsburgh PA, 1986.

[For86b] C. L. Forgy, *The OPS83 Report System Version 2.2*, Production Systems Technologies, Inc., Pittsburgh PA, 1986.

[HWL83] F. Hayes-Roth, D. A. Waterman, and D.B. Lenat, editors, *Building Expert Systems*, Addison-Wesley Publishing Company Inc., Reading MA., 1983.

[NeS72] A. Newell and H. A. Simon, *Human Problem Solving*, Prentice-Hall, Inc., Englewood Cliffs NJ., 1972.

[NLK87] R. Neches, P. Langley, and D. Klahr, "Learning, Development, and Production Systems" in D. Klahr, P. Langley, and R. Neches, editors, *Production System Models of Learning and Development*, The MIT Press, Cambridge MA., 1987, 1-53.

[SiC87] K. E. Simmons and T. C. Clarke, *EOS Transactions*. American Geophysics Union (abstract), (1987) **68**, 16, 280.

[Ste80] A. I. F. Stewart, *IEEE Transactions on Geoscience and Remote Sensing*, (January 1980) **GE-18**, 1.