

DOCUMENT RETRIEVAL TRIGGERED BY SPACECRAFT ANOMALY: USING THE KOLODNER CASE-BASED REASONING (CBR) PARADIGM TO DESIGN A FAULT-INDUCED RESPONSE SYSTEM

F. Kronberg, A. Weiner,¹ T. Morgan, B. Stroozas, E. Girouard, A. Hopkins, L. Wong, M. James,² J. Kneubuhl,³ & R. F. Malina⁴
Center for EUV Astrophysics, 2150 Kittredge Street,
University of California, Berkeley, CA 94720-5030, USA

ABSTRACT

We report on the initial design and development of a prototype computer-mediated response system, the Fault Induced Document Officer (FIDO), at the UC Berkeley Center for EUV Astrophysics (CEA) Extreme Ultraviolet Explorer project (EUVE). Typical 24x7 staffed spacecraft operations use highly skilled expert teams to monitor current ground systems and spacecraft state for responding to anomalous ground system and spacecraft conditions. Response to ground system error messages and spacecraft anomalies is based on knowledge of nominal component behavior and the evaluation of relevant telemetry by the team. This type of human-mediated operation is being replaced by an intelligent software system to reduce costs and to increase performance and reliability. FIDO is a prototype software application that will provide automated retrieval and display of documentation for operations staff. Initially, FIDO will be applied for ground systems. Later implementations of FIDO will target spacecraft systems. FIDO is intended to provide system state summary, links to relevant documentation, and suggestions for operator responses to error messages. FIDO will provide the operator with near realtime expert assistance and access to necessary information. This configuration should allow the resolution of many anomalies without the need for on-site intervention by a skilled controller or expert.

Keywords: case-based reasoning, knowledge-based systems, knowledge capture system.

¹ Research Engineer, Omitron, 6411 Ivy Lane, Suite 600 Greenbelt, MD 20770.

² AI Group, NASA/Cal Tech Jet Propulsion Lab, 4800 Oak Grove, Pasadena, CA 91109.

³ UC Berkeley undergraduate student.

⁴ Also Director, Laboratoire d'Astronomie Spatiale, Traverse du Syphon BP8, 13376 Marseille, France Cedex 12.

INTRODUCTION

In February of 1995, the Extreme Ultraviolet Explorer Science Operations Center (ESOC) moved to single-shift operations. Artificial Intelligence (AI) software monitors the telemetry in order to reduce staffing needs and improve monitoring of the health and safety of the satellite. Two of the human-monitored shifts were replaced by AI software that pages personnel when an anomaly is detected. The dynamics of building an anomaly response team under this new paradigm have changed since staff must be called in, and responses are inherently slower. In any anomalous situation, response time is always critical. Delays in responding to an anomaly may jeopardize science quality and observatory health.

The Fault Induced Document Retrieval Officer (FIDO) prototype software reduces the need for a large and costly response team. FIDO research initially focuses on the design and development of software that will provide a system state summary (a configurable engineering summary that displays the state of the ground systems and observatory) and documentation for diagnosis of a potentially failing component that might have caused the error message or anomaly. These features are complemented by a high-level documentation browsing interface indexed to the particular error message. The document collection consists of operations procedures, engineering problem reports, component documentation, and engineering drawings. The prototype design includes the capability of combining information on the state of the ground systems, with detailed, hierarchically organized, hypertext-enabled documentation.

FIDO will capture the knowledge of operators in two ways. Initially, when FIDO first retrieves documents for a particular error message or anomaly, the operator will be able to designate which documents he or she considers most relevant. FIDO will archive this scoring of the documents and present them accordingly when the error or anomaly occurs again. Because the documents will be scored by operators when errors occur, the most relevant documents would become the first to be displayed during subsequent events. If the operator creates a new document in response to a particular error or anomaly, FIDO will save the new document and present it the next time that the problem appears. A successful implementation of this prototype would decrease the response time and level of expertise needed to respond to errors, thereby increasing science productivity and reducing overall operating costs.

FIDO: PROBLEM STATEMENT

The typical ESOC response to an anomaly or a potential emergency involves a team of highly skilled employees and contractors. This team attempts to assess the state of the ground systems and spacecraft based on error messages and the latest telemetry update as

well as known behavior of the systems and components. This process usually involves a certain amount of research into ground systems, instrument, and flight software documentation. The process can be long, involved, and costly as experts on contract are expensive, especially when committed for rapid response times.

In the process of developing the “zero shift” capability at CEA, we discussed the need for computer-mediated or -assisted documentation selection and display for operations during the times without on site operators.

Experience teaches us that ground systems failures are much more common than failures of the spacecraft platform or payload components. Responses to ground systems failures occupy a majority of operators’ time and attention. Thus, ground systems become the most pertinent domain for applications of FIDO.

FIDO AS A KNOWLEDGE SYSTEM

The EUVE research regarding the delivery of complex sets of documentation to operators engaged in real-time activities involving ground and spacecraft systems relates to Feigenbaum’s concept of “knowledge publishing” (Feigenbaum 1993).¹ Knowledge publishing, Feigenbaum reports, is the conversion of normal, passive books into a system that delivers knowledge to the user in “active books,” specifically in the context of a user’s need or request. FIDO will “make active” the documentation used by operators during responses to errors and anomalies. FIDO is a knowledge system because it captures the knowledge of the operators.

FIDO AND THE KOLODNER MODEL OF CASE-BASED REASONING

Another aspect of our work is based on the research of Kolodner, which focuses on the use of previous experience as a learning resource for responding to subsequent anomalous events (Kolodner 1993). “Case-based reasoning” (CBR) is defined as a model that incorporates problem solving, problem understanding, and problem learning. CBR maintains a referential “case base” of earlier events and solutions.

¹ Edward A. Feigenbaum, in a video-taped lecture, “Tiger in a Cage: Applications of Knowledge-Based Systems” (Feigenbaum 1993) discusses various applications of knowledge-based systems. After a review of the history of knowledge systems—artificial intelligence using heuristic searching, problem-solving methods, and knowledge representation, Feigenbaum discusses four major areas of knowledge systems. Of the four systems, planning and scheduling, compliance or “rule helpers” configuration application and knowledge publishing, FIDO is most relevant to knowledge publishing but also has some relationship to configuration application, knowledge systems.

Kolodner bases her work on five premises. References to old cases are advantageous, and references to similar situations are necessary. Understanding or interpreting a situation is a necessary part of the reasoning cycle. Old cases need to be adapted to new situations. Learning occurs as a natural consequence of reasoning. Learning results in: learning of new procedures, refining of procedures, and learning when each is appropriately used.

FIDO follows Kolodner's model to the extent that it incorporates previous search results into its current search criteria, however, the current design of FIDO does not yet promote the "understanding" of problems as they occur. FIDO supports two of Kolodner's premises (case reference and case adaptation), but does not support the premises regarding "learning."

According to Kolodner, these premises suggest that the quality of an instance of case-based reasoning depends on five criteria: experiences it has had, ability to understand new situations in terms of old experiences, adeptness at adaptation, adeptness at evaluating and repair, and ability to appropriately integrate new experiences into existing memory.

Of Kolodner's five criteria that influence the "quality" of case-based reasoning, we believe that FIDO: will generate a body of experience as it is employed over time; may be able to understand new situations in terms of previous ones; at present, can adapt only manually to new situations; has some ability to evaluate and change its response; and can partially integrate new experiences into memory.

FIDO COMPARED TO FIXIT: CASE-BASED REASONING FOR SATELLITE GROUND CONTROL

Weiner et al. (1995) report on research using "case-based reasoning as a technique for storing fault management experience and making it available to operators confronting similar anomalous situations."

Weiner et al. suggest three trends motivate the use of CBR to aid anomaly resolution: increasing levels of automation, reduced budgets, and higher levels of personnel turnover. Our experience confirms this theory (Kronberg 1995).

The Fault Information Extraction and Investigation Tool (FIXIT) is a case-based architecture for computationally encoding fault management experience. This project represented one of the first applications of case-based reasoning to realtime decision making and system control. The FIXIT research explored the use of case-based reasoning as a technique for storing fault management experience and making it available to operators confronting similar anomalous situations. Specifically, the project investigated the use of case-based reasoning technology to construct a knowledge base of actual fault

management experience. This knowledge base was organized so as to enable the retrieval of fault management information in response to system inputs.

FIXIT has been implemented in a proof-of-concept form as a decision-aid to support controllers of NASA's SAMPEX spacecraft. The anomaly case base contained information from actual NASA spacecraft anomaly logs currently used to diagnose and remediate faults. To evaluate the effectiveness of the FIXIT architecture in supporting realtime fault management, Weiner et al. conducted an experiment using satellite ground control operators at NASA Goddard Space Flight Center.

Subjects detected anomalies, identified them as replicas or variants of previous anomalies, and determined the correct response more quickly with FIXIT than when using their conventional fault management tools. Results also showed that subjects' accuracy in diagnosis and response was significantly higher with FIXIT than with conventional tools.

The success of the proof of concept FIXIT application for satellite ground control suggests that a case-based fault management tool can be very effective in aiding operators performing fault management. FIXIT's success indicates that a FIXIT-like architecture describes a useful way to present fault management information to spacecraft operators. Furthermore, the FIXIT research indicates that providing operators with realtime access to information about previous system faults, coupled with tailoring the organization of this information to match the operator's fault management strategies, can lead to more effective fault management.

FIDO: DESIGN

We discussed the issue of what would be useful to an operator who is called in to respond to a possible anomaly or critical situation. We reviewed a number of knowledge systems. We compared the requirements with programs elsewhere. We determined the following preliminary requirements for FIDO:

1. User-friendly retrieval of knowledge: if an operator is not willing to interact with FIDO, the knowledge system will have little value.
2. Standard, intuitive user interface: design issues regarding human computer interaction must be addressed successfully.
3. Organized process of gathering documentation: the process itself must be logically organized and intuitively apparent to the operator, so as to provide an inherent understanding of why a specific document is being displayed.

4. A tree display (multilevel display of information with drill-down capability): too much information presented at one time is as bad as the presentation of incorrect information. The most relevant documents must be selected and displayed in such a way that an operator can choose as many or as few as needed.
5. Flat file database or Unix ASCII text: a simple file structure and data format will facilitate portability and easy transfer to other systems.
6. Learning ability to incorporate user experience into a case base: as the number of cases is built up over time, the system should increase its ability to present the most relevant information for each incident.
7. “Just-in-time” knowledge engineering: as little knowledge engineering as possible built into the document database. The relevance of each document to a specific error or anomaly must accumulate through the use of FIDO, and not be built in.

Based on the requirements determined from the discussions with users, we chose the following features and capabilities:

1. The user has: easy access to knowledge; capability to “forage” or drill-down through information; ability to capture knowledge and set preferences for future retrievals
2. The domain presents: system state information; feedback of effects of actions
3. Documentation types: operations procedures; engineering problem reports; reference material including design specs and memos; engineering drawings
4. Documentation structure: flat files; ASCII text; Postscript drawings
5. The process: takes input from error messages; maps error messages to keywords; indexes keywords and frequency of hits to specific documentation; retrieves and presents relevant information; allows the user to score appropriateness of documents retrieved; receives and integrate user response into a case base repository.

FIDO: FUNCTIONAL COMPONENTS

SYSTEM STATE AND SUMMARY WINDOW. The engineering state and summary window is a graphical representation of system state information. The window consists of a Java applet to be displayed in a Web browser, and receives input from the telemetry stream. This window displays immediate information about a fault and gives an operator the ability to view additional state information that is not a part of the fault display but may be connected up or down stream. This capability was demonstrated in FIDO v1.0.

DICTIONARY/ THESAURUS FILE. The dictionary/thesaurus file lists keywords that are mapped to ground system error messages from the host system. This file is formatted as an ASCII hash-table with each system error being linked to a primary keyword as well as a number of secondary keywords.

RULE BASE / INFERENCE ENGINE. The rule base/inference engine will not be implemented in FIDO version 1.5. The implementation decision awaits the results of the operational evaluation of FIDO v1.5, and is dependent upon the ability of the existing FIDO architecture to supply operators with relevant documentation. If FIDO requires a more complex knowledge structure, this will be implemented using the SHINE inference engine developed at Jet Propulsion Laboratory.

SEARCH ENGINE / INDEXER. FIDO utilizes the Glimpse search engine, developed at the University of Arizona, to index and retrieve documentation. Glimpse is a very powerful indexing and query system that allows whole files system to be searched extremely quickly. Glimpse is very similar to the 'grep' program familiar to most UNIX users, but it has several additional features that facilitate its use in FIDO. Glimpse is easily customizable, allowing for many different types of layered and combined queries, and also has an excellent Web interface.

DOCUMENT DATABASE. The FIDO document database consists of the existing collection of ESOC operational procedures, problem reports, problem reports, and reference documentation. Because most contemporary Web browsers have the capacity to view the majority of these document formats (ASCII, HTML, Postscript, GIF, JPEG), little or no document conversion will be required.

GRAPHICAL USER INTERFACE. FIDO will make use of a Web browser for its graphical user interface. This provides a cross-platform standard for developing FIDO's GUI (i.e., HTML) and allows FIDO to be accessed remotely from any number of platforms. FIDO's GUI consists of three main parts: a document browser, a number of document editors, and a tree display. The engineering state and summary window will also be displayed using the Web browser interface.

DOCUMENT BROWSER. FIDO's document browser provides the operator with a graphical way to access all of the information in the documentation database. Operators can use "Up" and "Down" buttons to scroll between different Procedures and problem reports. Each Procedure and problem report will also be hyperlinked to similar and related documents with the documentation database; all problem reports and problem reports will be explicitly linked to Procedures. This linking will enable an operator to quickly move from choosing a contingency procedure to viewing examples of its previous usage. Figure 1 displays several conceptual sketches of the document browser.

EDITORS. The rule editor consists of a window which allows for generation of entries in the Dictionary/Thesaurus file and for existing entries to be modified and saved. The rule editor will take the form of an HTML page using forms. The procedure editor consists of a window which allows new procedures to be written and existing procedures to be

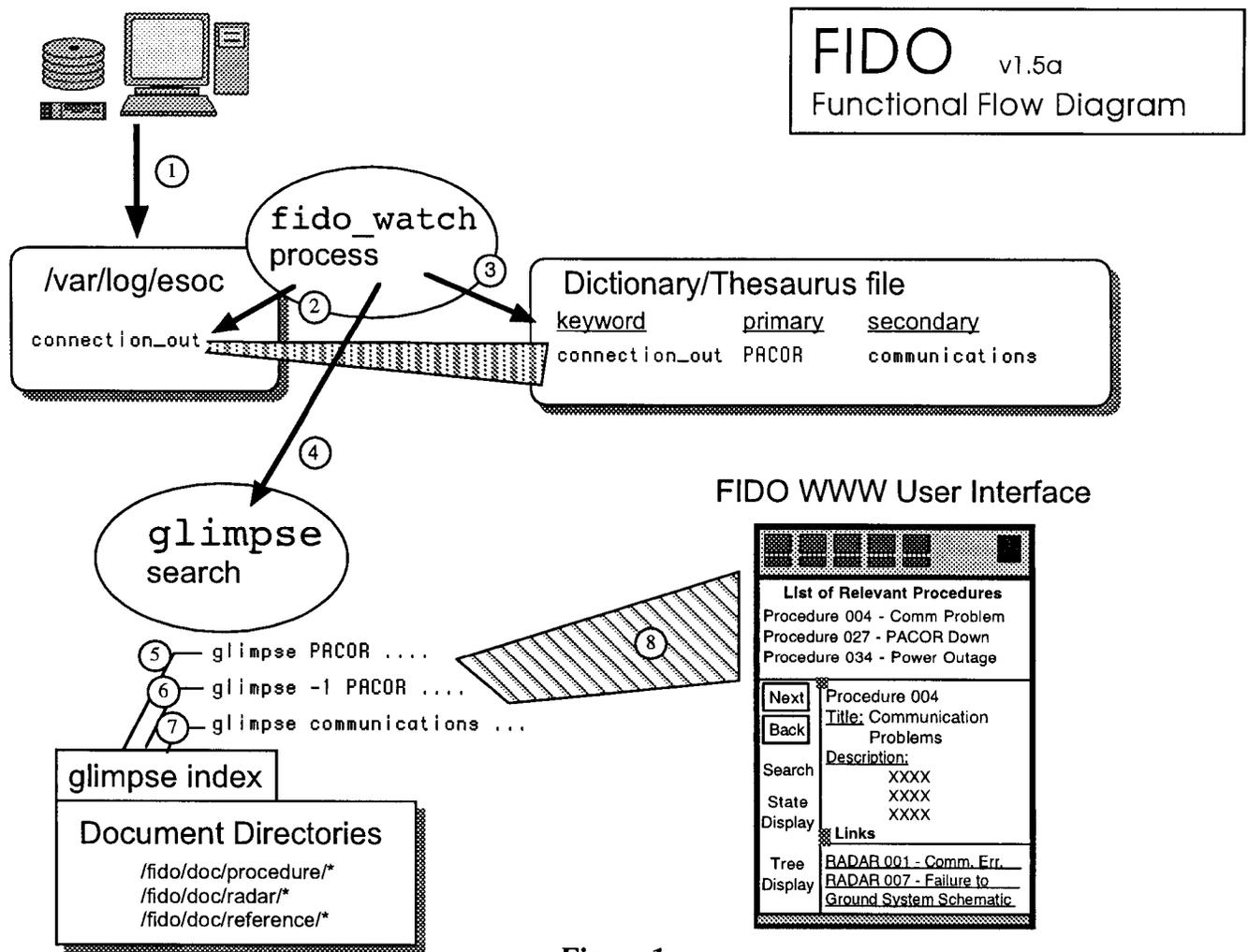


Figure 1.

modified and saved. The procedure editor will take the form of an HTML page using forms and a standard HTML text widget. Procedures will be stored as HTML; although the introduction of more powerful HTML editors (e.g., Netscape Gold) may allow for more comprehensive editing capabilities in future releases. The existing problem reporting system will be utilized for the initial release of FIDO.

TREE DISPLAY. The tree display is being developed in the form of a Java applet. The tree display will show a breakdown of the ground system by component and by function. At the leaf nodes of the tree will be Procedures, which will also be linked to individual problem reports. We have yet to determine how to store the structure of the tree, either as a string consisting of parent-child nodes or as representative of file directory structure.

FIDO PROCESS. In the first step of the FIDO diagnosis process, an element of the ground system generates an error. Whenever a ground system error occurs, an error message appends to the /var/log/esoc file. A shell script running as a cron job, fido_watch, checks the /var/log/esoc file for new entries. When a new error message is detected, fido_watch

retrieves the error message from the /var/log/esoc file and finds its associated keywords in the Dictionary/Thesaurus file. Fido_watch then initiates multiple separate Glimpse searches using these keywords.

The first Glimpse search is based on the primary keyword linked to the error message. The second Glimpse search again uses the primary keyword, but allows for a single misspelling when looking for relevant files. This is a useful technique for getting around the problem of over- under-specified keywords or procedure entries. Without this search, it is conceivable that a search for “tape_drive_3” would miss an entry generically related to tape drives that refers to “tape_drive_N”. All additional searches are performed using the secondary keywords from the Dictionary/Thesaurus file.

The output of the Glimpse searches is piped to a cgi script, which produces an HTML page with links to the referenced documents. In this way, the operator has access to all of the documentation associated with a particular keyword, with files listed according to their perceived relevance to the anomaly. Future versions of FIDO will make extensive use of the SHINE inference engine being developed at the Jet Propulsion Laboratory by James (1994).

At this point, the operator can access all of the available operational documentation in the ESOC, including procedures, problem reports, and additional documentation such as schematics and strip charts. FIDO also provides graphical tools for browsing, bookmarking, and editing this store of documentation. Bookmarks are used to supply FIDO with feedback about the relevancy of particular documents, thus even further reducing response time if the anomaly were ever to recur.

LESSONS LEARNED

Our work has resulted in several lessons that have guided significant application and design decisions. From FIDO 1.0 to FIDO 2.0 we decided that the spacecraft was too well behaved to warrant this type of knowledge system, accordingly we changed our focus to ground systems, which proves a much more relevant domain. From the work of Weiner et al., we realized the critical role of user confidence level and the necessity to facilitate the ease of use, so we implemented relevant features, e.g.: do not make user cut and paste between windows and allow the user to add to and change procedures in realtime. This work is supported by NASA contract NAS5-29298 and NASA/Ames grant NCC2-902.

REFERENCES

Feigenbaum, E.A. Tiger in a Cage, University Video Communications, POB 5129, Stanford, CA, 1993.

James, Mark L. SHINE 2.0 Reference Manual, Jet Propulsion Laboratory, NASA, 1994.

Kolodner, J. L. Case Based Reasoning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

Kronberg, R, Ringrose, P., Losik, L., Biroscak, D., and Malina, R. F. "Re-engineering the EUVE Payload Operations Information Flow Process to Support Autonomous Monitoring of Payload Telemetry," Reengineering Telemetry, XXXI, 1995, 286-294.

Weiner, A. J. et al. "FIXIT: A Case-Based Architecture For Computationally Encoding Fault Management Experience," Proceedings of the 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, Cambridge, MA, 1995.