StarPlan II: Evolution of an Expert System

Ronald W. Siemens, Marilyn Golden,
and Jay C. Ferguson

Ford Aerospace & Communications Corporation
Sunnyvale Operation, 1260 Crossman Avenue
Sunnyvale, California 94089-1198, (408)743-3206

ABSTRACT

An expert system for satellite anomaly resolution must perform monitoring, situation assessment, diagnosis, goal determination and planning functions in real time. StarPlan is such a system being developed at the Ford Aerospace Sunnyvale Operation. This paper details the evolution of the StarPlan architecture from a rule-based system in which multiple "experts" classified and resolved anomalies to a more generic architecture that utilizes an object model of the domain to perform fault diagnosis using causal reasoning. The StarPlan I architecture is described; the lessons learned in StarPlan I implementation are discussed; and the architecture of StarPlan II is presented.

1. INTRODUCTION

This is the second paper in a series on the evolution of the StarPlan architecture and knowledge representation [1]. StarPlan is an expert system that performs a fault diagnosis and resolution function for satellites [2]. The system monitors incoming telemetry from a satellite, alerts the satellite control operator to anomalous conditions and suggests corrective actions.

The architecture of StarPlan I, the first generation of the system, is described in this paper, and the lessons learned during implementation are detailed. Our experience with StarPlan I led to a significant architectural restructuring of the knowledge representation scheme that captures a model of the domain and uses that model to perform fault diagnosis by utilizing the relational links between the objects of the domain model and the declarative description of the object behaviors. Production rules are used only when the information being captured is not defined well enough to be modeled.

2. STARPLAN I ARCHITECTURE

The StarPlan I system architecture is based on Minsky's Society of Experts approach [3]. There are multiple knowledge sources [4] that are customized to specific problems; knowledge sources exist at different levels of abstraction; both goal driven and opportunistic control strategies are used [5]; and the communication between knowledge sources occurs through higher level meta-monitors.

The architecture of the system consists of five major components, shown in Figure 1.

- o Guardians classify incoming data, filter relevant data, and translate the data through methods from numeric to symbolic ranges to derive a set of hypotheses.
- o Monitors reason from the set of hypotheses established by the guardians to resolve specific classes of anomalies.
- o Meta-Monitors are responsible for the control, interaction and data fusion of the individual monitors.

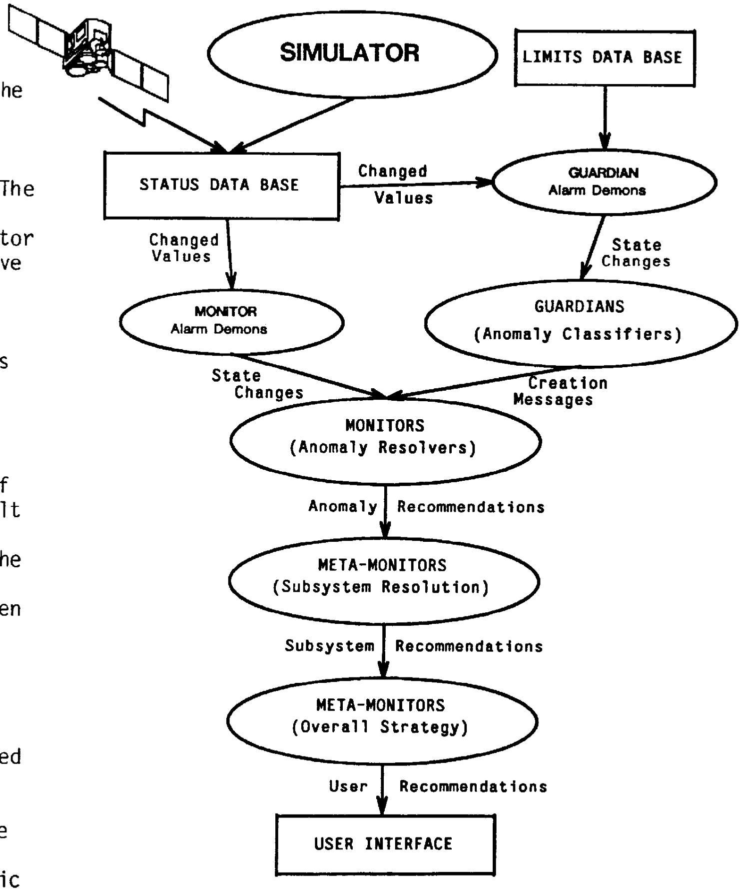

FIGURE 1 STAR-PLAN 1 ARCHITECTURE

- o Data Bases are used by the other components of the system to obtain relationships, facts, and other relevant information.

- o The Simulator models the satellite systems.

The system is implemented in the Knowledge Engineering Environment (KEE) from Intellicorp. KEE allows object oriented programming within class/subclass/member hierarchies, message passing, inheritance, and active values along with an integrated graphics interface [6].

2.1 Problem Detection

Each guardian, upon initialization, attaches alarms (active demons) to the necessary data values in the telemetry database. These alarms initialize their range sensitivity parameters from the data range limits database. Upon telemetry receipt the attached alarms check the sensitivity range. When a range has been exceeded the alarm sends the attached guardian a message containing the violated range parameter and the relevant satellite object involved (i.e., a battery). Based on the data range limits database the guardian incorporates new range parameters. The alarms act both as a filter to minimize processor utilization and an abstractor to reclassify specific data to a symbolic representation. This simplistic classification uses data ranges rather than specific data values.

2.2 Problem Determination

The guardians contain contextually partitioned subsets of rules [7] that watch for specific anomaly classes, which are usually grouped according to objects. A typical guardian may, for example, watch the batteries for associated anomalies (over or under temperature, current, or voltage problems). The attempt is to provide a cover set of rules to perform anomaly detection for a specific anomaly class that is small enough to be readily managed and verified by an expert. Once an anomaly is detected by the guardian, a monitor from a set of prototype monitors for that class of monitors (i.e., battery emergency overtemperature) is instantiated for the specific object (i.e., Battery 1) to resolve the anomalous condition. The current status of the alarm demon messages are maintained by the guardian as well as the monitor status (active/inactive). Further alarm messages regarding an anomaly with an active monitor are maintained but not acted on until the active monitor notifies the guardian it has completed resolution of the problem and removes itself from the system.

The guardians forward chain through the covering rule sets to try to match the incoming symbolic telemetry status patterns against known or expected anomaly patterns.

2.3 Problem Resolution

The monitors (anomaly resolvers) are goal-driven and contain rule-sets contextually partitioned to the specific anomaly class. Each hypothesis contains a rule set to guide the strategy of diagnostic procedure, problem resolution methods, command sequences and operational considerations.

Diagnostic strategy may require command recommendations to change satellite configuration (for safety or to eliminate specific hypotheses) which conform to an allowable set of command sequences designed to preserve spacecraft integrity. The monitor may require additional, or more detailed telemetry data on spacecraft status, so a monitor may set its own alarm demons to watch for rapid or unexpected changes, and then reschedule itself to rerun the rule set at a future time, allowing the satellite system time to respond. Alarms may tag data to allow explicit temporal reasoning [8,9] if necessary. Upon anomaly resolution the status database is updated to reflect the state change.

When the unexpected telemetry values that triggered the monitor have been controlled or corrected to the monitor's satisfaction and the system status has been updated, the monitor will delete its alarm demons and itself from the system, thereby allowing the guardian to once again set a monitor on that specific object and anomaly class if necessary. If the diagnostic procedure proves how an object failed, it will be marked as to the specific failure so that future diagnostic strategy and problem resolution reasoning can take the failed condition into account. For reasons of time constraints or satellite safety an object may be marked as having an unknown status, which may affect future actions in an entirely different manner.

2.4 Management of Multiple Experts

Since the guardians are looking for contextually specific patterns [10], even a single fault anomaly can activate multiple monitors that will be working on independent hypotheses. The conflicting, and sometimes contradictory, diagnostic procedures generated by the various monitors must be resolved by subsystem meta-monitors [11], which can give diagnostic control to the most urgent hypothesis within a group of monitors, usually centered around a satellite subsystem. The subsystem recommendations are coordinated in a top-level meta-monitor that decides overall strategy (i.e., if there are payload problems, status telemetry losses, and a power subsystem problem, then shut down non-essential payloads and allow the power subsystem monitor to make recommendations).

2.5 The Knowledge Base

The knowledge base consists of various databases that are used by the system to obtain required information. There are four databases to support the guardian and monitor functions.

behavioral information was still embedded in rules, sustaining complex rule sets.

3.4 Recommendations

The two major recommendations arising from an analysis of StarPlan I were: (1) Separate the functions of classification, diagnosis, goal determination, and planning and command to provide more modularity and less overlap of functions performed at various levels of processing, and (2) Define knowledge representation techniques in place of rules that provide semantic knowledge that can be addressed by generic problem-solving mechanisms.

4. STARPLAN II ARCHITECTURE

The StarPlan II architecture separates the monitoring, situation assessment, diagnostic, goal determination and planning that was inherent in the monitors and meta-monitors of StarPlan I. These correspond to monitoring, problem identification, diagnosis, goal identification and plan and modify in Clancy's Classification Hierarchy [14]. The five major components of the new system are: the Active DataBase, Situation Assessment, Causal Diagnosis, Goal Determination, and Planning & Command, as shown in Figure 2. These modules all operate on the same underlying knowledge representation, which is generalized and constrained by the knowledge acquisition tools so that the domain experts can represent their environment in a consistent manner.

The knowledge base contains a description of each object that can be reasoned about, given the telemetry available from the satellite. Each object defined has three basic parts: the attributes of the object, the relationship of the object to other objects in the satellite, and a behavioral description of the object. The behavioral description is captured declaratively in a process description language [15]. This language allows the expert to define the object's behavioral states, events and processes so that the information can be reasoned about.

4.1 Problem Detection

The general mechanism of the StarPlan I Alarm Demons was extended to create the Active DataBase [16]. The function of the Active DataBase is to monitor the incoming telemetry data to detect and notify the system when telemetry fails to meet expectations. All incoming data values are translated into symbolic point or interval values [17] that relate to the expectations that are in place at the time of data receipt. The expectations are symbolically expressed and when those expectations are violated a la Schank [18], the Active DataBase sends a notification message to the Situation Assessment module that an event of interest has occurred, for example "Battery 1 temperature is critically high." Knowledge acquisition tools [19] were developed that allow the knowledge engineer (KE) to identify each incoming telemetry datum from the satellite and specify both the symbolic translation and event notification mechanisms to be applied. The KE can graphically enter ranges, trends or specific conditions to be applied to the data each time it is received from the satellite (or simulator), the events of which the Situation Assessment module should be notified and their relative importance. Much of the factual information that was represented as production rules in StarPlan I can be entered into the computer using the knowledge acquisition tools in a manner that is natural and logical to the KE, is consistent to allow generic control mechanisms and is self-documenting.

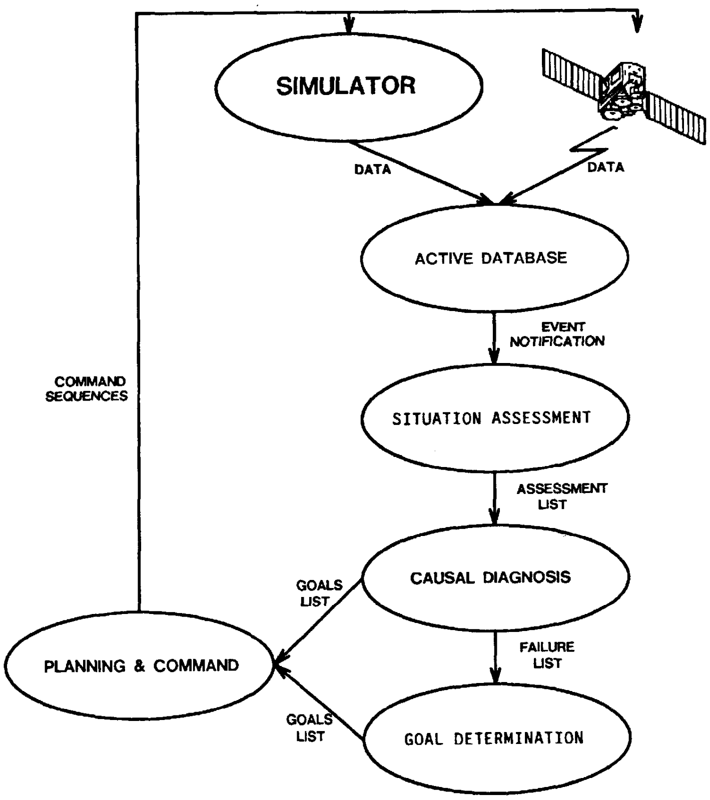

FIGURE 2 GENERIC PROBLEM-SOLVING ARCHITECTURE

During system operation, the telemetry data are acquired from the satellite in bursts called frames. A symbolic translation mechanism [20] assesses each datum of telemetry and sets its point and interval symbolic value (normal, high, low, increasing, not changing, unstable, etc.). Then the symbolic value is evaluated to see if the Situation Assessment mechanism is to be invoked (e.g., make the notification when the value is unstable and not increasing). After the Active DataBase has processed an incoming frame of data, it notifies the Situation Assessment module that a data cycle is complete.

4.2 Situation Assessment

As each data expection failure notification

The limits database contains the various ranges that a telemetry value can take (i.e., emergency over temperature) with the upper and lower values for each range specified. All alarms reference these limits. These range limits may be modified by historical analysis, configuration status changes, or short-term expectations derived from the diagnostic procedure. The status database contains information reflecting status data received, inferred from commands sent, and discovered during the diagnostic process. This database represents the dynamic status of the satellite and is used from pass to pass for continuity and planning. The telemetry database contains the latest telemetry received from the satellite. The alarms are attached to this database. The commands are grouped into their own database to facilitate capture of expertise concerning order and allowable combinations in command strings.

2.6 The Simulator

The simulator is used to generate telemetry data for testing purposes. Mathematical models of the various satellite systems have been developed to allow the simulation of the satellite in various states. The simulator allows real time testing for completeness of classification rules.

3. LESSONS LEARNED IN STARPLAN I

As StarPlan I was extended by the Knowledge Engineers to cover more complex portions of the domain, several structure, system test and knowledge engineering issues surfaced that pointed to weaknesses in the architecture.

3.1 Structure

Although one of the key features of the architecture is the contextual partitioning of the rules, the control of distributed rule sets can be complex and costly in terms of overhead. This is especially true when the data driven system must deal with multiple manifestations of a single fault or multiple faults [12]. In addition, the structure (or lack thereof) of rules themselves hampered the use of generic mechanisms for knowledge manipulation (one goal of our design was to be have a common core of generic mechanisms that would act on satellite-specific data so that expert systems for different satellites could be easily produced by swapping the satellite-specific data).

3.1.1 Multiple Manifestations of a Single Fault. When a fault occurs that causes telemetry associated with different classes of satellite objects to be out of limits, the guardians assigned to each class will instantiate monitors to handle the perceived problems. A meta-guardian is then required to detect that possibly a single fault, not multiple faults, is responsible for the problem, and an appropriate meta-monitor that could reason across both classes would be required to work on the problem solution in conjunction with the two class monitors. No meta-guardians were designed into StarPlan I which caused a problem when dealing with anomalies that were manifested in several classes of objects. A method for focusing on the most likely cause(s) of the problem is required; the combinatorial explosion involved with meta-guardians makes that an unlikely choice.

3.1.2 Multiple Faults. In the rare event that multiple faults occur simultaneously onboard the satellite, multiple monitors are instantiated. A meta-monitor regulates and controls the monitors' processing and command sequencing. The problem is that the meta-monitor may subsume the lower-level monitors' strategies which defeats the envisioned goals of partitioning. This seems inevitable as long as the diagnostic, problem resolution, and command sequence planning rules are interwoven.

3.1.3 Rule Structure. Rule-based systems offer enormous advantages over traditional software systems when it comes to separation of inference from control mechanisms. However, the lack of consistency and structure inherent in a rule-based system limit the use of generic processing mechanisms to pattern-matchers that operate only on patterns that exist in the data. If the data were structured in a manner that had semantic as well as syntactic significance, more generic problem-solving algorithms could be employed in the system.

3.2 System Test

Because of the possible side-effects introduced with any change to any rule in the system, testing of the system proved to be a difficult task [13]. The only really effective test approach for system validation is exhaustive testing, and re-testing after modification of the system.

3.3 Knowledge Engineering

One of the most difficult problems in building expert systems is obtaining the domain information from the expert and transferring it to an appropriate representation for use by the expert system. There were several problems encountered in collecting the knowledge for StarPlan I.

3.3.1 Mismatch Between Object Classes and Anomaly Classes. Although partitioned rule sets facilitated the knowledge acquisition process by constraining the expert to describe a small portion of the knowledge base at a time, the structure of the partitions did not always correspond well to the natural thinking processes of the domain experts.

3.3.2 Rules for Non-heuristics. When the original OPS-5 implementation of StarPlan was moved to the frame-based, semantic network provided by KEE, the factual data moved from rules to frames. However, the procedural and is received from the Active DataBase, the Situation Assessment module identifies the objects that appear to be involved and the event in which they are involved. After all notifications are generated from an incoming telemetry frame, the situation assessment module sweeps the active objects with a focus mechanism [21] and extracts the list of objects of maximum interest. This object list is passed to a ranking mechanism which ranks the list to form a situation assessment agenda to be passed to the Causal Diagnosis module. The entire situation assessment mechanism operates on the structure of the knowledge base and on any knowledge entered by the KE/expert. There are several ranking mechanisms which can be used either singly or in groups.

4.3 Causal Diagnosis

The function of the Causal Diagnosis module is to perform a causal analysis to explain expectation failures [22]. Using the Situation Assessment agenda as a guide, the causal diagnostic mechanism can directly reference an object in the knowledge base and use its local attributes, relationship data, and behavioral description to determine what has failed [23]. For example, if the battery 1 temperature is too high, the diagnostic module can get battery 1's internal variables for the temperature, and look at the behavior associated with temperature. This definition shows that the temperature is calculated from the internal variable current times the internal variable resistance plus the external variable heat from heater A. If the battery itself is not causing the excess temperature, the stored relationship data will provide indexes to the external objects that can contribute to the problem, and further analysis continues until the cause of the problem is detected. The output of the causal Diagnosis module is a list of what objects are broken and the state in which they failed (e.g., Heater A on Battery 1 is failed open).

During causal diagnosis, the diagnostic mechanism may initiate tests in the form of satellite configuration changes to prove/disprove a specific hypothesis under consideration. The diagnostic goal is passed to the Planning and Command module for planning and for mission operational constraint checking. The Planning and Command module may reformulate the plan on satellite safety or operational priority grounds.

4.4 Goal Determination

With the list of failed objects and their failure status, the Goal Determiner can then identify the configuration goals needed to resolve the anomaly [24]. In some cases the diagnostic procedure may have left the satellite in an unbroken condition (e.g., broken heater A is now isolated from power) and only the status of the object of concern must be marked with the failure so it will not be activated in the future. In other cases the satellite may be placed in a "safe" mode with all unnecessary functions shut down. The Goal Determination module must then provide a goal list for powering the systems back up, bypassing failed components. This goal list is passed on to Planning and Command for operation constraint checking.

4.5 Planning and Command

This module receives a set of goals and creates a plan for transitioning from the current state to the goal state, and then determines the command sequence necessary to accomplish the plan [25]. Next operational constraints are taken into consideration to ascertain the correctness of the plan. The behavioral descriptions of the objects in the domain can be consulted to look at the effects of the plan prior to execution.

4.6 Simulator

In StarPlan I the simulator used a mathematical modeling mechanism. The experts had to have an object simulation working before being able to test any of the rules for the object monitors. In StarPlan II the behavior of the object is declaratively defined [26] and is used as the basis of the simulator. Since the states, events to induce state changes, and processes that occur in each state are defined, it is possible to compile this declarative representation in LISP (or any other language) and execute it directly. Due to the objects being constrained to using only internal variables or variables that can be accessed through relational links, the system is defined in a controlled manner. The simulator can be used to simulate all or part of the satellite telemetry; the objects that are to be simulated plus their external inputs can be selected to be run in simulation mode.

4.7 Knowledge Representation

It is the underlying structure provided by the knowledge representation methodology, PARAGON, [27] that supports the functional modules (i.e., classification, diagnostics, etc.) that gives this system its powerful, generic application capability. This representation allows the expert to create concepts (a noun that describes a group of things or the actual instances of things) and the relationships that exist between concepts in the domain of interest, i.e., a model of the domain with which the system can reason.

The selection of the knowledge representation method for StarPlan II followed an analysis of the most common techniques for knowledge representation, a summary of which is detailed in Table 1. A hybrid knowledge representation scheme was designed that incorporated the strong points of each of the techniques and eliminated the weaknesses through an overriding requirement for consistent definition.

Table 1
Knowledge Representation Technique Tradeoffs

TECHNIQUE	DESCRIPTION	STRENGTHS	WEAKNESSES
RULES	CONTEXTUALLY DEPENDENT FACTS	•FLEXIBLE •STAND ALONE •REPRESENT POORLY STRUCTURED AND/OR POORLY UNDERSTOOD INFORMATION •AVAILABLE DEVELOPMENT TOOLS	•LACK OF STRUCTURE •NO METHODOLOGY OF DEVELOPMENT •SEMANTICS DIFFICULT •PROBLEM SOLVING TECHNIQUES LIMITED •DIFFICULT TO MANAGE •DIFFICULT TO MAINTAIN •HINDER GENERIC DEVELOPMENT •DIFFICULT TO REPRESENT CONTROL AND/OR TEMPORAL KNOWLEDGE
OBJECT ORIENTED	FRAMES OF RELATED FACTS AND BEHAVIOR USING MESSAGE PASSING FOR CONTROL	•FLEXIBLE •DATA AND BEHAVIOR PACKED TOGETHER •MAINTAINABLE •AVAILABLE DEVELOPMENT TOOLS	•NO UNDERLYING PRINCIPLES OR CONSTRAINTS •LACK OF DEVELOPMENT METHODOLOGY •NO ASSOCIATED PROBLEM-SOLVING TECHNIQUES
CLASSIFICATION SYSTEM	HIERARCHICALLY-ORDERED FRAMES OF RELATED FACTS AND BEHAVIORS	•STRUCTURE •INHERITANCE •MAINTAINABLE	•LACK OF CONSISTENT DEFINITION BETWEEN LEVELS •LACK OF DEVELOPMENT METHODOLOGY •ONE DIMENSIONAL (LIMITS NUMBER OF RELATIONSHIPS REPRESENTED)
SEMANTIC NETWORK	GRAPH OF NODES (REPRESENTING CONCEPTS) AND LINKS (REPRESENTING RELATIONSHIPS)	•WIDE VARIETY OF RELATIONSHIPS REPRESENTED •SOME METHODOLOGY •NATURAL REPRESENTATION	•AMBIGUOUS DEFINITION OF RELATIONSHIPS •LACK OF DEFINED PROBLEM SOLVING METHODS
BLACKBOARD	STRUCTURE OF DOMAIN AND HOW LEVELS OF DOMAIN COMMUNICATE OR INTERACT	•MULTIPLE LEVELS •ABILITY TO DEFINE INTERACTION BETWEEN LEVELS •INDEPENDENT KNOWLEDGE SOURCES CONTRIBUTE	•COMPLEXITY OF DEFINITION •LACK OF EXPLICIT CONTROL METHODOLOGY

4.8 Knowledge Acquisition Tools

The best way to achieve a consistent underlying knowledge representation structure throughout the PARAGON knowledge acquisition process [28] is to provide knowledge acquisition tools [29] which translate the experts' input into that structure. A large effort has gone into producing tools that assist the expert in defining telemetry data, the expectations associated with the data, the concepts that comprise the domain, their interrelationships and their behavior.

5. SUMMARY

The StarPlan II architecture, its underlying knowledge representation scheme, and the automated knowledge acquisition tools are a vast improvement over the StarPlan I system. The consistent definitions applied throughout the knowledge acquisition process have allowed the development of generic control and problem-solving mechanisms. Perhaps the greatest benefit derived from StarPlan II is that not only will it facilitate building anomaly resolution systems for a wide variety of satellites, it is generic enough to be the basis for any problem-solving system in which the domain is understood well enough to be declaratively modeled.

REFERENCES

[1] Ferguson, J.C., R.W. Siemens, R.E. Wagner, STAR-PLAN: A Satellite Anomaly Resolution andPlanning System, Proceedings of AAAI Workshop on Coupling Symbolic and Numerical Computing in Expert Systems, August 27-29, 1985.

[2] Golden, M., R.W. Siemens, An Expert System for Automated Satellite Anomaly Resolution, Proceedings AIAA/ACM/NASA/IEEE Computers in Aerospace V Conference, October 21-23, 1985

[3] Minsky, M., Matter, Mind & Models in Semantic Information Processing, edited by Marvin Minsky, MIT Press, Cambridge, Mass., 1968

[4] Hayes-Roth, B., The Blackboard Architecture: A General Framework for Problem Solving, Stanford University Heuristic Programming Project Report #HPP-83-30, May, 1983.

[5] Buchanon, B.G., E.H. Shortliffe, Rule-Based Expert Systems, Addison-Wesley, 1984

[6] Fikes, R., T. Kehler, The Role of Frame-Based Representation in Reasoning, Communications of the ACM, Vol. 28 Number 29, September 1985, pp 904-920.

[7] Cohen, P.R., E.A. Feigenbaum, Rule Space, The Handbook of Artificial Intelligence, Vol. III, Heuristech Press, W. Kaufmann Inc.

[8] Denbigh, K.G., Three Concepts of Time, Springer-Verlag, Berlin, 1981.

[9] Fagan, L.M., VM: Representing Time-Dependent Relations in a Medical Setting, Ph.D. Thesis, Stanford University, 1980.

[10] Watterman, D.A., F. Hayes-Roth, Pattern-Directed Inference Systems, Academic Press, 1978

[11] Genesereth, M.R., D.E. Smith, Meta-Level Architecture, Stanford University Heuristic Programming Project Memo HPP-81-6, Dec. 1982.[12] Genesereth, M.R., The Use of Design Descriptions in Automated Diagnosis, Qualitative Reasoning about Physical Systems, D.G. Bobrow, pp. 411-436, MIT Press, Cambridge, Mass.

[13] Hickam, D.H., E.H. Shortliffe, M.B. Bischoff, A.C. Scott, C.D. Jacobs, A Computer-Based Treatment Consultant for Clinical Oncology, The Quality of Computer-Generated Advice, Stanford University Heuristic Programming Project Memo HPP-84-9, May 1984.

[14] Clancy, W.J., Heuristic Classification, AI Journal, Vol. 27, Number 3, Dec., 1985, pp. 289-350, North-Holland, Amsterdam.

[15] Ferguson, J.C., Declarative Representation for Procedural Knowledge: A Structured Semantic Network Approach, Internal Proprietary Report, Ford Aerospace & Communications Corporation, Sunnyvale Operation, 1260 Crossman Ave., Sunnyvale, CA 94089-1198

[16] Heher, D.M., Genesis: An Intelligent Active Data Base, Internal Proprietary Report, Ford Aerospace & Communications Corporation Sunnyvale Operation, 1260 Crossman Avenue, Sunnyvale, California 94089-1198.

[17] Kahn, M.G., J.C. Ferguson, E.H. Shortliffe, & L.M. Fagan, An Approach for Structuring Temporal Information in the Oncocin System, Proceedings of the Symposium on Computer Applications in Medical Care, 1985

[18] Schank, R.C., Dynamic Memory, Cambridge University Press, 1982

[19] Contreras, V., J.C. Ferguson, Knowledge Management, Internal Proprietary Report, Ford Aerospace & Communications Corporation, Sunnyvale Operation, 1260 Crossman Avenue, Sunnyvale, California 94089-1198

[20] Forbus, K.D., Qualitative Process Theory, Qualative Reasoning about Physical Systems, D.G. Bobrow, pp. 85-168, MIT Press, Cambridge, Mass.

[21] Ferguson, J.C., Situation Assessment: An Efficient Focus Mechanism, Internal Proprietary Report, Ford Aerospace & Communications Corporation, Sunnyvale Operation, 1260 Crossman Avenue, Sunnyvale, California 94089-1198

[22] Schank, R.C., Questions and Thought, Yale University Report YALEU/CSD/RR#385, August 1985

[23] Ferguson, J.C., Causal Analysis: Explanation based on Accountability, Internal Proprietary Report, Ford Aerospace & Communications Corporation, Sunnyvale Operation, 1260 Crossman Avenue, Sunnyvale, California 94089-1198

[24] Ferguson, J.C., Using Behavioral Domain Knowledge to Determine Goals, Internal Proprietary Report, Ford Aerospace & Communications Corporation, Sunnyvale Operation, 1260 Crossman Avenue, Sunnyvale, California 94089-1198

[25] Ferguson, J.C., Planning through Behavioral Knowledge Based on Constraints & Skeletal Plans, Internal Proprietary Report, Ford Aerospace & Communications Corporation, Sunnyvale Operation, 1260 Crossman Avenue, Sunnyvale, California 94089-1198

[26] Ferguson, J.C., Simulation through Declarative Models, Internal Proprietary Report, Ford Aerospace & Communications Corporation, Sunnyvale Operation, 1260 Crossman Avenue, Sunnyvale, California 94089-1198

[27] Ferguson, J.C., PARAGON: A Knowledge Representation Methodology for Defining Declarative Models, Internal Proprietary Report,Ford Aerospace & Communications Corporation, Sunnyvale Operation, 1260 Crossman Avenue, Sunnyvale, California 94089-1198

[28] Musen, M.A., L.M. Fagan, D.M. Combs, E.H. Shortliffe, Facilitating Knowledge Entry for an Oncology Therapy Advisor Using a Model of the Application Area, Stanford University Technical Memo KSL-86-1

[29] Ferguson, J.C., Graphical Acquisition & Translation of Knowledge into PARAGON, Internal Proprietary Report, Ford Aerospace & Communications Corporation, Sunnyvale Operation, 1260 Crossman Avenue, Sunnyvale, California 94089-1198